



**FAIR:
Forecasting en Netwerk Analytics voor
het Beheer van het Inningsrisico**

**Proefschrift voorgedragen tot
het behalen van de graad van
Doctor in de Toegepaste
Economische Wetenschappen**

door

Véronique VAN VLASSELAER

Daar de proefschriften in de reeks van de Faculteit Economie en Bedrijfswetenschappen het persoonlijk werk zijn van hun auteurs, zijn alleen deze laatsten daarvoor verantwoordelijk.

Since the theses in the series published by the Faculty of Economics and Business are the personal work of their authors, only the latter bear full responsibility.

Committee

<i>Chairperson</i>	Prof. dr. Katrien Antonio	KU Leuven
<i>Promotor</i>	Prof. dr. Bart Baesens	KU Leuven
<i>Co-Promotor</i>	Prof. dr. Monique Snoeck	KU Leuven
	Prof. dr. Tina Eliassi-Rad	Rutgers University
	Prof. dr. Leman Akoglu	Stony Brook University
	Ir. Dries Van Dromme	Smals Research
	Dr. Jan Meskens	Erasmus Hogeschool

Acknowledgments

This Ph.D would have never been completed without the support, feedback and encouragement of many people. Although listing people would be a difficult and dangerous exercise, I wish to acknowledge some of them.

First and foremost, I would like to express my sincere gratitude to my promotor, prof. dr. Bart Baesens for offering me this position, for believing in me and my skills. His enthusiasm has always driven me forward; his confidence in me – even when I lost it myself – is remarkable. Bart, I have very much enjoyed working with you. You are an extremely talented man: a great researcher but also an excellent advisor and mentor who always knows how to motivate and support people. It is a pleasure to have been a part of your team. I will always remember the great laughs we had (which password should we take?), our joint presentations, unforgettable EasyJet flights (or was it SleasyJet?) and the pleasant trips to conferences in Barcelona and Paris together with your charming wife Katrien and your kids.

I am indebted to all other committee members of this Ph.D for their valuable insights and suggestions. To prof. dr. Monique Snoeck who amazes me with her mathematical skills and knowledge about conceptual modeling. Monique, you were a great help to perfect my work down to the details. Furthermore, as being the former vice-dean of education, you easily guided me through all the complex administrative steps of a Ph.D. To prof. dr. Tina Eliassi-Rad, who I first met on the ICDM conference in Brussels in 2012. This was the start of a

fruitful and enjoyable collaboration. Tina, thank you for inviting me twice to New York. I loved being there, and I definitely lost a piece of my heart in New York. You hold impressive knowledge on all papers related to our field (I will always remember you saying things like “Why don’t you have a look at paper X from author Y presented on conference Z in year T? There is a section A in that paper that handles that specific problem.”). If Google fails one day, at least we have an alternative in our field. Also, you know so many other researchers (I remember you saying “I know him/her. I will send him/her an email for further info.”) that the reason why state-of-the-art social network analysis techniques struggle with scalability issues is probably because of you and your personal social network. Furthermore, I loved the exciting pool (re-)matches with you and your husband Branden. To prof. dr. Leman Akoglu for her impressive technical insights and utmost creative ideas. To ir. Dries Van Dromme and dr. Jan Meskens for guiding me through the complex data of the social security institution. Dries and Jan, because of you this dissertation is not merely a scientific what-if story. You both carefully screened and evaluated all the ideas formulated in this dissertation (and many more) on its usability and practical implication, making this work perfectly intertwining theory and practice. Also, I enjoyed working once a week at Smals in Brussels, together with you and all your other colleagues.

I cannot think of one day during these past three years that I did not like to go to work. This is especially due to all my former and current colleagues that were and are still working at our department: Aimée, Bing, Estefanía, Eugen, Filip, Gayane, Helen, Jan, Jasmien, Jochen, Johannes, Klaas, María, Michael, Philippe, Seppe, Tine, Tom, Wilfried and Wouter. I wish to offer a special thank you to Aimée, who started her Ph.D together with me and who is more than a colleague. Our talks and after-work body workouts were inspiring for body and mind. To Helen, my former colleague and friend with whom I shared an office and who taught me many things about the Ethiopian culture. To Michael, with whom I currently share an office, for his enthusiastic morning greetings and interesting cultivating skills. To Johannes, a great thinker with whom I studied the bachelor and master program Information Systems Engineering.

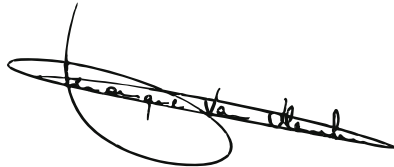
To Wouter, an intelligent, down-to-earth researcher and co-author of our book “Fraud Analytics” which we wrote together with Bart.

The perfect way to relax and forget about the long computing time, all the out-of-memory errors and L^AT_EX layout issues, are undoubtedly the amusing, and often foolish conversations with my lovely friends Daisy, Emmy, Erica, Gert, Jasper, Jasper, Jens, Joachim, Jochen, Joline, Joris, Kimberley, Maarten, Roselín, Sarah, Stef, Tijs, Tine, Tom, Vreni and Willem who have been part of more than half of my life. Especially our Thursday evening ‘Spaghettidagen’ have been a great amusement to start off the last working day of the week with a fresh mind and new crazy ideas.

Furthermore, I am grateful to my parents-in-law, Peter and Yvette, and siblings-in-law Jietse and Inne, Siebe and Lisa for the warm and enjoyable Friday evenings which we spend with the whole family. They have always been there for me, listening to my (probably for them) boring stories about (probably for them) geeky statistics, data issues and coding.

I sincerely appreciate the everlasting support of my parents Eddy and Els in everything I do, giving me the opportunity to study, to enjoy my student life and making my life easy without any worries. I could have always count on them. They have helped me more than they probably realize.

Finally, I would like to deeply thank the man who has been part of this wonderful journey from the start, who has always been there for me, the man who makes me smile every single day, the man who makes me who I am... my husband and soul mate Niels. Niels, thank you for all the encouragement, love and support. Your positivity is inspiring. I have enjoyed every moment with you and I look forward to many more. Thank you for being you!



Véronique Van Vlasselaer
September, 2015

*To my husband and soul mate Niels for his never-ending support.
To my parents, parents-in-law, and siblings-in-law.*

Preface

Fraud! Many stories about successful and unsuccessful perpetration of fraudulent actions are going around. And if we are truly honest, we enjoy hearing them. Fraud is something sensational that attracts the attention of many people, and is in some way sexy and exciting. Oxford Dictionary defines fraud as “*the wrongful or criminal deception to result in financial or personal gain*”. According to this definition, fraud is a broad concept, even including small-scale, low-impact activities where most people come up against at least once. Think about withholding the fact that you accept too much change at a shop, or buying a cheap, designer sunglasses knowing that it is a fake, or when you “borrowed” glasses from bars as a student, etc. While such stories are the so-called “white collar” crimes which are rather innocent in nature and are considered as ‘acceptable’ or ‘borderline acceptable’ (Smith et al., 2010), other stories are more notorious as they involve huge amounts of money or the deception of large groups of people or even whole countries. A couple of stories keep echoing through history:

- The first well-known example of financial fraud is the sale of the Roman Empire to Didius Julianus (193 AD) for 25000 sestertii per soldier (around 1 billion euros) by the Praetorian guards who murdered the previous emperor Pertinax. Back then, Praetorian guards were seen as the loyal army supporting the current emperor. Unfortunately, the Praetorian guards sold something that did not belong to them. Three months later, Septimius Severus claimed the Empire back, decapitating Didius Julianus (Dio, c. 170).

Afterwards, many famous landmarks were unlawfully sold through history: the Eiffel Tower by Victor Lustig (sold twice), the Brooklyn Bridge by George C. Parker (sold twice a week for 30 years), the non-existing island Poyais off the coast of Honduras by Gregor MacGregor, the Taj Mahal and the Parliament House of India together with its members by Natwarlal, etc.

- In 1717, John Law was sent to Louisiana in order to help in the development of the French colony. He founded the Mississippi Company. When he returned back to France about the conditions, the French people were overwhelmed about stories of massive amounts of gold and jewelry in the colony. Shares of Mississippi Company skyrocketed. Unfortunately for the investors, it was all propaganda. When the investors realized it was a con, shares plummeted and many people lost all their money. Nowadays, such fraud is known as the Mississippi bubble.
- Another notorious example of deception, is a story from 1785 where Cardinal Prince de Rohan bought a diamond necklace of more than two million livres for Queen Marie Antoinette with whom he was having an affair. Unfortunately for de Rohan, he was having an affair with a prostitute that resembled to the Queen.
- The Baker Estate Swindle lasted for more than 70 years deceiving more than 3000 people for more than 3 million dollars. It all started with Colonel Jacob Baker who died in 1839 in Philadelphia. At the time, Colonel Jacob Baker supposedly owned more than a quarter of the whole city. Many years later, William Cameron Morrow Smith founded an association to help the presumable heirs with tracing their family back to Jacob Baker and claiming their inheritance which, according to the association, would comprise more than 2000 acres of land. Many people with last name Baker deposited money to support the association. Unfortunately, there was no Baker estate (Nash, 2004).
- ...

The list is endless and still growing. Although fraudsters operate in the spirit of their time, resulting in new types of fraud that continuously pop up, old-fashioned ways of perpetrating fraud are constantly coming back and are still exploited. On the flip side, fraud fighters in all areas have developed a profound set of skills and techniques to prevent known criminal activities and to limit unseen fraud.

This dissertation aims to fight fraud from a *data science perspective*. That is, the development of automated detection techniques that are capable of processing massive amounts of data in a limited time span which generate a highly accurate, meaningful and precise output. Data science encompasses every theory, strategy and action undertaken that use data, ranging from data definition and collection to the interpretation, implementation and evaluation of knowledge derived from data. To date, many businesses and industries start to explore the world of data science, including data analysis and predictive modelling. Many successful use cases prove the power and possibilities of data science, and its ability to support business decisions in an accurate and efficient way. Examples include, among others, credit risk modelling (Baesens et al., 2003a), customer churn prediction (Verbeke et al., 2011; Backiel et al., 2014), process mining (De Smedt et al., 2015; vanden Broucke et al., 2014), data quality (Moges et al., under review), recommender systems (Seret et al., 2012) and fraud detection.

In order to develop high-performing, efficient detection models, a set of challenges need to be addressed. In Chapter 3, a more thorough and detailed characterization of the multifaceted phenomenon of fraud is given which states that fraud is an *uncommon, well-considered, imperceptibly concealed, time-evolving* and often *carefully organized* crime which appears in many types and forms. Each characteristic is associated with particular challenges related to develop a fraud detection model. Throughout this dissertation we will show how these potential threats can be translated into opportunities. The main focus of this work lies especially on the last two properties: how to leverage social interactions among people, whilst taking into account that the intensity of such interactions varies over time. The proposed tech-

niques are tested on data of two application domains: social security fraud (Chapter 3 - 5) and credit card fraud (Chapter 6).

Although this work tends towards a generic approach to tackle fraud, there is no silver bullet solution. There are two reasons for this. First, the development of fraud detection mechanisms and models requires the insights and knowledge of subject matter experts. It is a two-way process: experts guide models, whilst models guide experts in turn. Experts are able to provide (a) a concise set of fraudulent observations or transactions and (b) more details about the modi operandi of fraudsters. While (a) is necessary to learn supervised detection models, (b) is essential in both supervised and unsupervised learning. In supervised learning, one learns from labeled (e.g., fraud or non-fraud) observations and tries to predict the behavior of new, unseen instances. Unsupervised learning does not rely on labeled data, but tries to find a hidden structure in the data. This work mainly focuses on supervised fraud detection. Whereas traditional detection techniques mainly use structured data, this dissertation shows the value of a new data source: the social network, containing the social relationships among people. A set of detection techniques based on social network analysis is proposed. These techniques help the data scientist to seize networked data by formulating the right network definition, extracting useful and relevant features (i.e., the *featurization* process), and enriching traditional fraud detection models which ultimately tends to result in an increased performance and novel, valuable insights.

A second reason why no silver bullet can exist is that newly developed detection models force fraudsters to change their tactics. Fraud is dynamic. Fraudsters learn from their mistakes and try to find new loopholes. Models need to be continuously updated. As such, the fight against fraud will remain a cat-and-mouse game between the fraud fighter and perpetrator.

This dissertation does not claim to be a guide on how to stop fraud in the end, but rather to sensitize about the power of data science in fraud detection and to serve as a source of inspiration for every data scientist that is entailed to uncover past, present and future methods of fraudsters in any application domain.

Contents

Committee	iii
Acknowledgments	v
Preface	xi
1 Introduction	1
1.1 Data science	1
1.2 The fraud triangle	4
1.3 Types of fraud	5
1.4 Outline and contributions	10
1.4.1 Chapter 2	10
1.4.2 Chapter 3	11
1.4.3 Chapter 4	12
1.4.4 Chapter 5	13
1.4.5 Chapter 6	14
2 Fraud! A Social Network Approach	17
2.1 It's the network, you stupid!	19
2.1.1 Social networks	21
2.1.2 Network components	24
2.1.3 Network representation	28
2.2 Is fraud a social phenomenon?	32
2.3 Overview of the featurization process	36
2.3.1 PageRank	39
2.3.2 Gibbs Sampling	40
2.3.3 Iterative Classification Algorithm	40

2.3.4	Relaxation labeling	41
2.3.5	Loopy belief propagation	41
2.4	Conclusion	41
3	GOTCHA! A network-driven approach for fraud detection	43
3.1	Introduction	43
3.2	Social Security Fraud Detection	48
3.2.1	Background	48
3.2.2	Challenges	52
3.2.3	Related Work	55
3.2.4	Proposed Fraud Detection Process	57
3.2.5	GOTCHA!'s Fraud Detection Framework	58
3.3	Network Analytics for Fraud Detection	60
3.3.1	General Concepts and Notations	60
3.3.2	Time-weighted Bipartite Networks	61
3.3.3	GOTCHA!'s Fraud Propagation Algorithm: Defining high-risk nodes in the network	63
3.3.4	Network Feature Extraction	72
3.4	Modeling Approach	75
3.4.1	Rebalancing the data set	76
3.4.2	Learning algorithm	77
3.5	Results	77
3.5.1	Out-of-time Validation	85
3.5.2	Curtailing newly originated spider constructions	88
3.6	Conclusions	89
4	GOTCHA'!! Fraudulent clique detection	93
4.1	Introduction	93
4.2	Social Security Fraud	96
4.3	Related Work	97
4.4	Proposed Method	98
4.4.1	Task description	98
4.4.2	Individual Exposure Score	99
4.4.3	Clique Detection and Scoring	100
4.4.4	Feature extraction	103
4.4.5	Detection model	104
4.5	Empirical Evaluation	105

4.5.1	Data Set	105
4.5.2	Performance over time	105
4.5.3	Precision	109
4.5.4	Variable importance	110
4.6	Conclusions	111
5	AFRAID: Active Fraud Investigation and Detection	113
5.1	Introduction	114
5.2	Background	116
5.3	Network definition	117
5.4	Active Inference	118
5.4.1	Collective Inference Technique	120
5.4.2	Probing strategies	122
5.4.3	Temporal weighing of label acquisition	125
5.5	Results	127
5.6	Related Work	132
5.7	Conclusion	133
6	APATE: Anomaly Prevention using Advanced Transaction Exploration	135
6.1	Introduction	136
6.2	Credit Card Transaction Fraud	139
6.2.1	Background	139
6.2.2	Credit Card Fraud Detection Process	142
6.2.3	Related Work	143
6.3	Proposed Methodology	145
6.3.1	Intrinsic Feature Extraction	146
6.3.2	Network Feature Extraction	150
6.4	Results	158
6.4.1	Prediction Results	159
6.4.2	Variable Importance and Network Variable Impact	161
6.5	Conclusions	166
7	Conclusions	167
7.1	Conclusions	167
7.2	Future research	173
7.2.1	Network dynamics	173

7.2.2	Multi-view learning	175
7.2.3	Rationales	176
7.2.4	Internet of Things (IoT)	177
	Publication list	201
	DOCTORAL DISSERTATIONS LIST	205

Chapter 1

Introduction

“The government are very keen on amassing statistics. They collect them, add them, raise them to the n^{th} power, take the cube root and prepare wonderful diagrams. But you must never forget that every one of these figures comes in the first instance from the village watchman, who just puts down what he damn pleases.”

— Josiah Stamp, 1929

1.1 Data science

Data science is a booming industry. Recently, IDC reported that investments in big data and data science would increase from \$16.55 billion dollar in 2014 to about \$41.52 billion in 2018. More and more companies realize the potential asset that might come along with data science. Rather than relying business decisions on pure intuition, decisions are now based on statistically confirmed evidence extracted from... data. Competition in terms of just doing data science no longer suffices, it is now all about doing good data science. Harvard Business Review even stated that a company’s effort on big data is often lost. They say that the actual success of those investments strongly depend on the ability of *good people* to use *good data*.¹

¹<https://hbr.org/2013/12/you-may-not-need-big-data-after-all>, retrieved on July 2015.

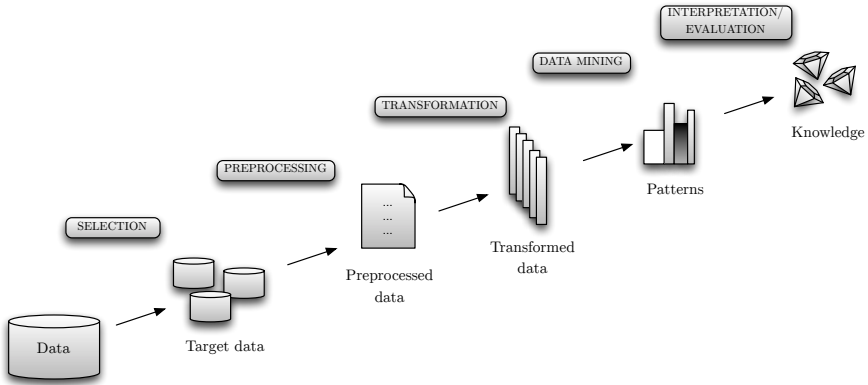


Figure 1.1: Knowledge Discovery in Databases (KDD) process.

Only this will eventually result in good results and a worthwhile *return on analytics* that effectively support the corporate decisions. According to Baesens et al. (2015), the capabilities of a data scientist are fivefold: a data scientist should (a) have solid quantitative skills, (b) be a good programmer, (c) excel in communication and visualization skills, (d) have a profound business understanding and (e) be creative. Data science is seen as a multidisciplinary field. Being a data scientist requires a thorough understanding of other areas such as machine learning, data mining, statistics and often profound business knowledge and insights, but then again it is the sexiest job of the 21st century.² Secondly, data should be reliable, accurate and sound. It should reflect reality in the best way possible. One of the oldest principles in data analysis is “garbage in, garbage out”. A well thought-out data collection, storage and management system is one of the indispensable requirements of good data analysis.

But what is data science exactly? Data science encompasses every theory, strategy and action undertaken that use data, ranging from data definition, collection and storage to the interpretation, implementation and evaluation of knowledge derived from data. One of the important pillars in data science is the KDD (Knowledge

²<https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century/>, retrieved on July 2015.

Discovery in Databases) process, which describes how one has to convert raw data into useful information and knowledge which serves to support operational, tactical and strategic decision making. In general, the KDD consists of five steps: data selection, pre-processing, transformation, data mining and interpretation of the results. This is depicted in Figure 1.1. Each knowledge generation problem has to go through the several stages of the KDD process. This dissertation is mainly situated in the transformation and data mining steps of the KDD process by (a) transforming unstructured network data to useful and meaningful features that can be used to support further analysis, and (b) mining the data to find new evidence of fraud. Remark that the data collection, pre-processing and evaluation steps of the KDD process are also implicitly part of this dissertation, but due to the sensitivity of the data at hand, it is not further considered.

The focus of this work is fraud detection, one of the applications in the multifaceted research domain of data science. The aim is to develop automated detection techniques which help fraud experts in the fight against fraud. Fraud detection approaches data from another perspective. Rather than searching for a pattern repeatedly popping up in a data set, research in fraud tries to find abnormal behavior or anomalies. Two remarks need to be made. First, under the assumption that average behavior is normal, behavior of each individual should be compared to the others. The question that arises here is, “Is one’s behavior in line with overall behavior?” This is compliance on *data set level*. Second, a sudden change in customer’s personal behavior might also indicate fraud. This leads to the following question: “Does one’s behavior comply with normal behavior for that person?” A shift in a person’s spending patterns or callee list may hint towards a stolen credit card or a telcom account that is taken over. This is compliance on *data item level*. In this dissertation, detection models that operate on both data set and data item level are presented.

This dissertation is an interactive play between theory and practice. From a theoretical point of view, new approaches with regard to fraud detection are proposed. State-of-the-art network analysis is

used to enrich traditional detection models. Each chapters aims to be generic. All approaches are evaluated on a real-life fraud data set. Chapter 3-5 use data provided by the Belgian Social Security Institution (social security fraud), Chapter 6 is tested on data from Worldline (credit card fraud).

1.2 The fraud triangle

Without people having the tendency to commit fraud, this dissertation would have never existed. This leads to an insurmountable, first question, “Why do people commit fraud?”, or in other words, what drives people to seize opportunities to commit fraud. In order to understand the underlying motives or drivers of fraudsters, Cressey (1953) developed the “fraud triangle” (see Figure 1.2). The triangle analyzes the motives behind fraud from three angles:

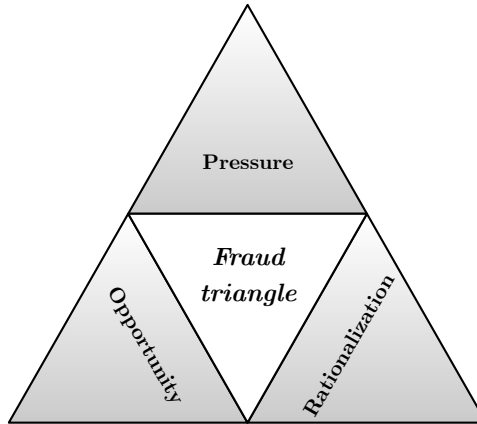


Figure 1.2: The fraud triangle.

- **Pressure:** Pressure (or incentive, or motivation) is triggered by something in a fraudster’s personal life that creates a stressful need to proceed to illegal activities (Singleton and Singleton, 2010). The motives of fraud are rather personal and divergent. Stamler et al. (2014) categorizes fraud motives into four main

areas: economic, egocentric, idealistic and psychotic. The most prevalent reason is economical in nature. It is often sourced from the fact that people do not have the money they need (Stamler et al., 2014). This does not necessarily mean that fraudsters tend to obtain money in cash. In credit card fraud, a credit card is unlawfully used to make purchases. Call behavior fraud comprises the crime where fraudsters use the account of someone else to make their calls. In insurance or healthcare fraud, people abuse the system such that they are granted a allowance or are exempted from payment. From an egocentric point of view, people commit fraud to achieve prestige, a good reputation or even fame at any expense. Scientific fraud, often carried out by plagiarism, is an example where one illicitly steals someone else's ideas and results; or where one makes up the surrounding setting and/or the results. The idealistic motive can be traced back as a way of people to counterbalance discrimination. They often believe that they act legally and blame their victims for their actions. Psychotic motives include a distorted sense of reality, delusions of grandeur or persecution (Singleton and Singleton, 2010).

- ***Opportunity:*** This angle comprise the ability of fraudster to commit fraud. Loopholes in governmental laws or corporate policies facilitate the perpetration of fraud. Opportunities range from crimes committed in situations that coincidentally occur – which are referred to by Smith et al. (2010) as “white collar” crimes – to deliberately searching for ways to cheat.
- ***Rationalization:*** is a psychological mechanism that explains why fraudsters do not refrain from committing fraud and think of their conduct as acceptable (Baesens et al., 2015).

1.3 Types of fraud

Many types of fraud exist. Moreover, fraud is deeply influenced by the time being. In ancient Rome, one of the most popular ways to swindle was to quarry low-quality marble with many holes, filling these holes

with wax and selling the marble for a high price.³ Nowadays, the share of construction fraud has to make way for other, currently more common types of fraud.

Based on Baesens et al. (2015), the following non-exhaustive list of fraud categories is composed that are currently prevalent:

Credit card fraud In credit card fraud there is an unauthorized taking of another's credit. Some common credit card fraud subtypes are counterfeiting credit cards (for the definition of counterfeit, see below), using lost or stolen cards, or fraudulently acquiring credit through mail (definition adopted from definitions.uslegal.com). Two subtypes can be identified (Bolton and Hand, 2002): (1) Application fraud, involving individuals obtaining new credit cards from issuing companies by using false personal information, and then spending as much as possible in a short time span; (2) Behavioral fraud, where details of legitimate cards are obtained fraudulently and sales are made on a 'Cardholder Not Present' (CNP) basis. Remark that this does not necessarily require stealing the physical card, only stealing the card credentials. Behavioral fraud concerns most of credit card fraud. Also debit card fraud occurs, although less frequent. Credit card fraud involves in fact a form of identity theft, as will be defined below. Chapter 6 discusses a network-based approach on how to deal with credit card fraud.

Insurance fraud Broad category spanning fraud related to any type of insurance, both from the side of the buyer or seller of an insurance contract. Insurance fraud from the issuer (seller) includes selling policies from non-existent companies, failing to submit premiums and churning policies to create more commissions. Buyer

³<http://www.fraud-magazine.com/article.aspx?id=4294972770>, retrieved on July 2015.

fraud includes exaggerated claims (property insurance: obtaining payment that is worth more than the value of the property destroyed), falsified medical history (healthcare insurance: fake injuries), post-dated policies, faked death, kidnapping or murder (life insurance fraud), faked damage (automobile insurance: staged collision), etc. (definition adopted from www.investopedia.com).

Corruption Corruption is the misuse of entrusted power (by heritage, education, marriage, election, appointment or whatever else) for private gain. This definition is similar to the definition of fraud provided by the Oxford Dictionary discussed before in that the objective is personal gain. It is different in that it focuses on misuse of entrusted power. The definition covers as such a broad range of different sub-types of corruption, so does not only cover corruption by a politician or a public servant, but also e.g. by the CEO or CFO of a company, the notary public, the team leader at a workplace, the administrator or admissions-officer to a private school or hospital, the coach of a soccer team, etc. (definition adopted from www.corruptie.org).

Counterfeit An imitation intended to be passed off fraudulently or deceptively as genuine. Counterfeit typically concerns valuable objects, credit cards, identity cards, popular products, money, etc. (definition adopted from www.dictionary.com).

Product warranty fraud A product warranty is a type of guarantee that a manufacturer or similar party makes regarding the condition of its product, and also refers to the terms and situations in which repairs or exchanges will be made in the event that the product does not function as originally described or intended. (definition adopted from www.investopedia.com). When a product fails to offer the described functionalities or displays deviating

characteristics or behavior that are a consequence of the production process and not a consequence of misuse by the customer, compensation or remuneration by the manufacturer or provider can be claimed. When the conditions of the product have been altered due to the customer's use of the product, then the warranty does not apply. Intentionally wrongly claiming compensation or remuneration based upon a product warranty is called product warranty fraud.

Healthcare fraud

Healthcare fraud involves the filing of dishonest healthcare claims in order to make profit. Practitioner schemes include: individuals obtaining subsidized or fully-covered prescription pills that are actually unneeded and then selling them on the black market for a profit; billing by practitioners for care that they never rendered; filing duplicate claims for the same service rendered; billing for a non-covered service as a covered service; modifying medical records, etc. Members can commit healthcare fraud by providing false information when applying for programs or services, forging or selling prescription drugs, loaning or using another's insurance card, etc. (definition adopted from www.law.cornell.edu).

Opinion fraud

Online reviews often have a major impact on the popularity of a certain product or service. As a result, review systems are often targeted by opinion spammers who seek to distort the perceived quality of a product by creating fraudulent reviews. Opinion fraud involves reviewers (often paid) writing bogus reviews (Akoglu et al., 2013).

Telecommunications fraud

Telecommunication fraud or call behavior fraud is the theft of telecommunication services (telephones, cell phones, computers, etc.) or the use of telecommunication services to commit other forms of fraud (definition adopted from itlaw.wikia.com). An important

example concerns cloning fraud, i.e. the cloning of a phone number and the related call credit by a fraudster, which is an instance of superimposition fraud in which fraudulent usage is superimposed upon (added to) the legitimate usage of an account (Fawcett and Provost, 1997).

Money laundering

The process of taking the proceeds of criminal activity and making them appear legal. Laundering allows criminals to transform illegally obtained gain into seemingly legitimate funds. It is a worldwide problem, with an estimated \$300 billion going through the process annually in the United States (definition adopted from legal-dictionary.thefreedictionary.com).

Click fraud

Click fraud is an illegal practice that occurs when individuals click on a website's click-through advertisements (either banner ads or paid text links) to increase the payable number of clicks to the advertiser. The illegal clicks could either be performed by having a person manually click the advertising hyperlinks or by using automated software or online bots that are programmed to click these banner ads and pay per click text ad links (definition adopted from www.webopedia.com).

Identity theft

The crime of obtaining personal or financial information of another person for the purpose of taking over that person's name or identity in order to make transactions or purchases. Some identity thieves sift through trash bins looking for bank account and credit card statements; other more high-tech methods involve accessing corporate databases to steal lists of customer information (definition adopted from www.investopedia.com).

Tax evasion

Tax evasion is the illegal act or practice of failing to pay taxes which are owed. In businesses, tax evasion can occur in connection with income taxes, employment

taxes, sales and excise taxes, and other federal, state, and local taxes. Examples of practices which are considered tax evasion: knowingly not reporting income or under-reporting income, i.e. claiming less income than you actually received from a specific source (definition adopted from biztaxlaw.about.com). Chapter 3 - 5 tackle social security fraud which is a type of tax evasion (see *infra*).

Plagiarism Plagiarizing is defined by Meriam Webster's online dictionary as to steal and pass off (the ideas or words of another) as one's own, to use (another's production) without crediting the source, to commit literary theft, to present as new and original an idea or product derived from an existing source. It involves both stealing someone else's work and lying about it afterward (definition adopted from www.plagiarism.org).

The approaches developed in this dissertation are applied on (a) social security fraud, which is a form of corporate tax evasion, and (b) credit card fraud. Although we lack data from other application domains, these approaches are promising for similar problem settings.

1.4 Outline and contributions

In this section, we highlight the outline and main contributions of each chapter in this dissertation. Chapter 2 is an introductory chapter, familiarizing the reader with network analytics in a fraud context. Chapter 3 - 5 develop network-based detection techniques to address social security fraud. The last chapter 6 deals with how to tackle credit card fraud from a network-based perspective.

1.4.1 Chapter 2

In this chapter, the reader is introduced to the main concepts of network analysis and why network analysis might provide useful information to enrich traditional fraud detection techniques. In general, the following topics are discussed:

- This chapter provides a detailed overview of networks and their components, accompanied by examples of fraud networks that can easily be created using in-house business data. In addition, it is argued that the network can be represented (a) graphically for visualization purposes which are often part of the pre- and post-processing phase, and (b) mathematically for the automated computation of useful statistics and extraction of meaningful features.
- The concept of homophily is reviewed, and how this can serve as a primary indication whether the network might contain meaningful and relevant information for fraud detection. Two main approaches to measure homophily are discussed.
- The featurization process is introduced which addresses how unstructured network information can be translated into a set of structured features for each observation.

Chapter 2 has been published in:

Bart Baesens, Véronique Van Vlasselaer, and Wouter Verbeke. *Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection*. John Wiley & Sons, 2015.

1.4.2 Chapter 3

This chapter provides a framework on how to incorporate insights from network analysis into fraud detection models. The main focus of this chapter is to identify individual fraudsters. The proposed approach is named GOTCHA!

- From a data science perspective, a definition of fraud is presented identifying the main challenges that concur with fraud. The definition serves as a guiding principle throughout this and the next chapters, systematically addressing each challenge in order to develop a generic approach to tackle fraud.

- An extensive literature review is conducted including all research related to automated fraud detection using machine learning and network analysis.
- A novel, generic, scalable and integrated approach on how (social) network analytics can improve the performance of traditional fraud detection models in a social security fraud context is developed. The proposed approach is called GOTCHA! which exploits bipartite graphs in a timely manner.
- Given a limited set of fraudulent nodes in the network, GOTCHA! propagates fraud through the network, threatening fraud as a virus contaminating nearby neighbors. A time-dependent exposure score for *each* node in the network is derived.
- GOTCHA! is both forgiving and proactive in nature by anticipating future fraud whilst simultaneously decaying the importance of past fraud.

Chapter 3 has been submitted for publication in:

Véronique Van Vlasselaer, Tina Eliassi-Rad, Leman Akoglu, Monique Snoeck, and Bart Baesens. Gotcha! network-based fraud detection for social security fraud. *Management Science*, under review.

In addition, Chapter 3 has been published in:

Véronique Van Vlasselaer, Jan Meskens, Dries Van Dromme, and Bart Baesens. Using social network knowledge for detecting spider constructions in social security fraud. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 813–820. IEEE, 2013.

1.4.3 Chapter 4

Rather than focusing on individual fraudulent behavior, Chapter 4 elaborates on finding cliques of fraudsters, the so-called guilt-by-constellation. Using the properties of confirmed fraudulent cliques, GOTCHA!!! is able to detect uncovered groups in order to disband the complete fraudulent structure.

- As the structure of bipartite graphs differs from one-mode graphs, a new definition of cliques is proposed.
- Using a time-dependent individual exposure score of every node, cliques in the network are assigned a suspiciousness score which expresses the extent to which a clique acts suspiciously.
- Rather than guiding learning algorithms by confirmed fraud, GOTCHA'll! aims to learn from the structure of cliques, and the local properties of the clique members. It is shown that bankruptcy is an important indicator and often comes along with fraud.

Chapter 4 has been published in:

Véronique Van Vlasselaer, Leman Akoglu, Tina Eliassi-Rad, Monique Snoeck, and Bart Baesens. Guilt-by-constellation: fraud detection by suspicious clique memberships. In *Proceedings of 48th Annual Hawaii International Conference on System Sciences (HICSS)*, 2015.

1.4.4 Chapter 5

This chapter investigates how active inference fosters the fraud detection process for social security fraud. The goal is to select a limited set of nodes to be probed – i.e., inspected to confirm the true label – such that the misclassification cost of the collective inference algorithm is minimized. The proposed approach is called AFRAID.

- AFRAID is a new approach for active inference in a timely manner by (1) using time-evolving graphs, and (2) weighing inspectors' decisions according to recency. (1) This contribution is more elaborately discussed in Chapter 3. (2) Given that an inspector labels a specific node at time t , this chapter proposes how to temporarily integrate an inspector's decision in the network model, decreasing the value of the decision over time.
- A combination of fast and simple probing strategies is proposed to identify nodes that might possibly distort the results of a collective inference approach. Probing decisions made by (1)

a committee of local classifiers, and (2) by insights provided by inspectors are evaluated. (1) A committee of local classifiers collectively votes for the most uncertain nodes without relying on domain expertise. (2) Inspectors use their intuition to formalize which nodes might distort the collective inference techniques

- In fraud, inspectors often have a limited budget at their disposal to investigate suspicious instances. The benefits of investing a part of the total budget in learning a better model are investigated.

Chapter 5 has been published in:

Véronique Van Vlasselaer, Tina Eliassi-Rad, Leman Akoglu, Monique Snoeck, and Bart Baesens. Afraid: fraud detection via active inference in time-evolving social networks. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Network Analysis and Mining (ASONAM)*, IEEE, 2015.

1.4.5 Chapter 6

Chapter 6 presents a new approach, named APATE, on how to fight credit card transaction fraud.

- The requirements imposed by the credit card transaction domain differ from those of the social security domain. The network consists of credit card holders and merchants connected by means of transactions. Instead of starting from a limited set of fraudulent nodes (in social security fraud: companies), APATE uses a set of confirmed fraudulent edges (i.e., the transactions) to propagate fraud through the network. This chapter shows how this can be done by opening up the bipartite graph to a tripartite structure.
- A new incoming transaction is evaluated whether it is in line with normal customer behavior. Intrinsic or local-only features are derived using the fundamentals of RFM (Recency – Frequency – Monetary Value), enriched with features from the network.

- APATE complies with the six-second rule of decision, i.e. within six seconds the APATE algorithm decides whether a transaction can or cannot be pursued.

Chapter 6 has been published in:

Véronique Van Vlasselaer, Cristián Bravo, Olivier Caelen, Tina Eliassi-Rad, Leman Akoglu, Monique Snoeck, and Bart Baesens. Apate: A novel approach for automated credit card transaction fraud detection using network-based extensions. *Decision Support Systems*, 75:38–48, 2015.

Chapter 2

Fraud! A Social Network Approach

In the last decade, the use of social media web sites in everybody's daily life is booming. People can continue their conversations on on-line social network sites like Facebook, Twitter, LinkedIn, Google+, Instagram, etc. and share their experiences with their acquaintances, friends, family, etc. It only takes one click to update your whereabouts to the rest of the world. Plenty of options exist to broadcast your current activities: by picture, video, geo-location, links, or just plain text. You are on the top of the world... and everybody's watching. And this is where it becomes interesting.

Users of on-line social network sites explicitly reveal their relationships with other people. As a consequence, social network sites are an (almost) perfect mapping of the relationships that exist in the real world. We know who you are, what your hobbies and interests are, to whom you are married, how many children you have, your buddies with whom you run every week, your friends of the wine club, etc. This whole interconnected network of people knowing each other somehow, is an extremely interesting source of information and knowledge. Marketing managers no longer have to guess who might influence whom to create the appropriate campaign. It is all there... And that is exactly the problem. Social network sites acknowledge the richness of the data sources they have, and are not willing to share them as such and free of cost. Moreover, those data are often

privatized and regulated, and well-hidden from commercial use. On the other hand, social network sites offer many good built-in facilities to managers and other interested parties to launch and manage their marketing campaigns by exploiting the social network, without publishing the exact network representation.

However, companies often forget that they can reconstruct (a part of) the social network using in-house data. Telecommunication providers, for example, have a massive transactional data base where they record call behavior of their customers. Under the assumption that good friends call each other more often, telcom providers are able to recreate the network and indicate the tie strength between people based on the frequency and/or duration of calls (Backiel et al., 2014). Internet infrastructure providers might map the relationships between people using their customers' IP-addresses. IP-addresses that frequently communicate are represented by a stronger relationship. In the end, the IP-network will envisage the relational structure between people from another point of view, but to a certain extent as observed in reality. Many more examples can be found in e.g. the banking, retail and online gaming industry.

Also, the fraud detection domain might benefit from the analysis of social networks. In this dissertation, we underline the social character of fraud. This means that we assume that the probability of someone committing fraud depends on the people (s)he is connected to. These are the so-called *guilt-by-associations* (Koutra et al., 2011). If we know that five friends of Bob are fraudsters, what would we say about Bob? Is he also likely to be a fraudster? If these friends are Bob's only friends, is it more likely that Bob will be influenced to commit fraud? What if Bob has 200 other friends, will the influence of these five fraudsters be the same?

This chapter will briefly introduce the reader to networks and their applications in a fraud detection setting. One of the main questions answered throughout this chapter is how to represent a network for (a) visualization purposes, and (b) in a mathematically interesting manner. Next, the concept of *homophily* is reviewed. Homophily measures the extent to which fraudsters are connected to

other fraudsters, and is a way of deciding upfront whether detection models might benefit from network analysis. Next, we enter upon how unstructured network information can be translated into useful and meaningful characteristics of a subject. This is referred to as the featurization process. The main approach taken in this dissertation is to enrich traditional data analysis techniques with network-based features.

The remainder of this chapter is organized as follows: in Section 2.1 networks, their components and representation are discussed. Section 2.2 focuses on the concept homophily. Section 2.3 continues with a brief overview of neighborhood and centrality metrics that can be derived from the network. Additionally, collective inference algorithms are discussed. Section 2.4 concludes this chapter.

2.1 It's the network, you stupid!

Networks are everywhere. Making a telephone call requires setting up a communication over a (wired) network of all possible respondents by sending voice packages between the caller and the callee. The supply of water, gas and electricity for home usage is a complex distribution network that consists of many source, intermediary and destination points where sources need to produce enough output such that they meet the demand of the destination points. Delivery services need to find the optimal route to make sure that all the packages are delivered at their final destination as efficiently as possible. Even a simple trip to the store involves the processing of many networks. What is the best route to drive from home to the store given the current traffic? Given a shopping list, how can I efficiently visit the store such that I have every product on my list?

One of humans' talents is exactly the processing of these networks. Subliminally, people have a very good sense in finding an efficient way through a network. Consider your home-work connection, depending on the time and the day, you might change your route to go from home to work without explicitly drawing the network and running some optimization algorithm. Reaching other people, even without the telecommunication media of nowadays like telephone and

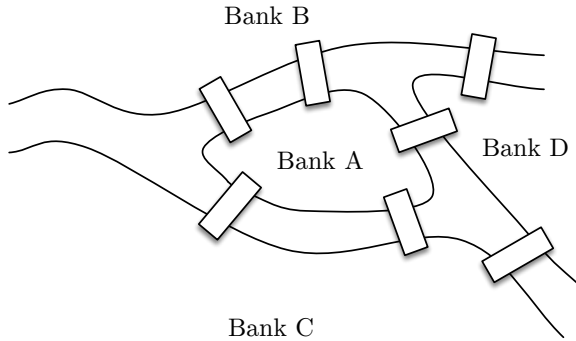


Figure 2.1: Königsberg bridges.

internet, is often an easy task for people. There is always a friend of a friend who knows the guy you are looking for.

The mathematical study of optimizing network-related problems has been introduced many years ago by Euler (1736). He formulated the problem of the *Königsberg bridges*. Königsberg (now Kaliningrad) was a city in Lithuania which was divided into four parts by the river Pregel. Seven bridges connected the four banks of the city (see Figure 2.1). The problem is as follows, ‘Does there exist a walking route that crosses all seven bridges exactly once?’ A path that can traverse all edges (here: bridges) of a network exactly once, is an *Eulerian path*. Euler proved that such path cannot exist for the Königsberg bridge problem. More specifically, an Eulerian path only exists when all nodes (here: banks) are reached by an even number of edges, except for the source and sink node of the path which should have an odd number of bridges pointing to it. Analogously, a *Hamiltonian path* in the network is a path that visits each node exactly once. For example, the *Travelling Salesman Problem (TSP)* tries to find a Hamiltonian path in the network. Given a set of cities, the idea is that a salesman has to visit each city (i.e., node) exactly once to deliver the packages. As this is an NP-hard problem, research mainly focuses on finding good heuristics to solve the TSP.

2.1.1 Social networks

Although in the previous example networks are built and developed by humans, they are not social. A key question here is, ‘What makes a network social?’ In general, we might say that a network is social whenever the actors are people or groups of people. A connection between actors is based on any form of social interaction between them, such as a friendship. As in the real world, social networks are also able to reflect the intensity of a relationship between people. How well do you know your contacts? The relationship between two best friends completely differs from the relationship between two distant acquaintances. Those relationships and their intensity are an important source of *information exchange*.

The psychologist Stanley Milgram measured in 1967 how social the world is. He conducted a *Small World experiment* whereby he distributed 100 letters to random people all over the world. The task at hand was to return the letter to a specified destination, which was one of Milgram’s friends. Rather than sending the letter back by mail, people could only pass the letter to someone they knew. This person, on their turn, had to forward the letter to one of his/her contacts, and so on... until the letter reached its final destination. Milgram showed that, on average, each letter reached its destination within six hops. That is, less than six people are needed to connect two random people in the network. This is the *average path length* of the network. The result of the experiment is widely known as the *Six Degrees of Separation* theorem. Milgram also found that many letters reached their target destination within three steps. This is due to the so-called “funneling effect”. Some people are known and know many other people, often from highly diverse contact groups (e.g., work, friends, hobby, etc.). Those people are sociometric superstars, connecting different parts of the network to each other. Many paths in the network pass through these people, giving them a high betweenness score (see Section 2.3).

While the Six Degrees of Separation theorem is based on results in real-life, many studies already proved that an average path length of six is an overestimation in online social networks. Those studies reported an average path length of approximately four hops between any two random people in an online social network (Kwak et al., 2010;

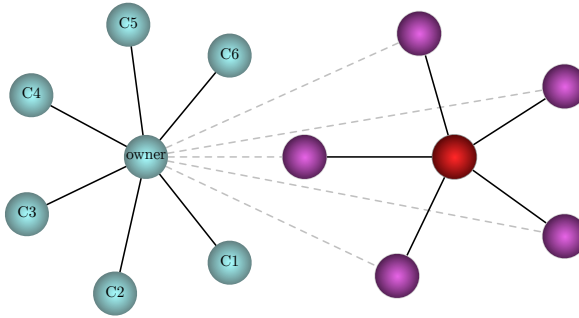


Figure 2.2: Identity theft. The frequent contact list (C1-C6) of a person is suddenly extended with other contacts (dark nodes). This might indicate that a fraudster (dark node in the center) took over that customer’s account and “shares” his/her contacts.

Van Vlasselaer et al., 2012). Online social networks are thus denser than real-life networks. However, the intensity between relationships might strongly differ.

Social networks are an important element in the analysis of fraud. Fraud is often committed through illegal set-ups with many accomplices. When traditional analytical techniques fail to detect fraud due to a lack of evidence, social network analysis might give new insights by investigating how people influence each other. These are the so-called *guilt-by-associations*, where we assume that fraudulent influences run through the network. For example, insurance companies often have to deal with groups of fraudsters, trying to swindle by resubmitting the same claim using different people. Suspicious claims often involve the same claimers, claimees, vehicles, witnesses, etc. By creating and analyzing an appropriate network, inspectors might gain new insights in the suspiciousness of the claim and can prevent the pursue of the claim.

In social security fraud, employers try to avoid paying their tax contributions to the government by intentionally going bankrupt. Bankrupt employers are not capable to redeem their tax debts to the government, and are discharged from their obligations. However, social network analysis reveals that the employer is afterwards re-

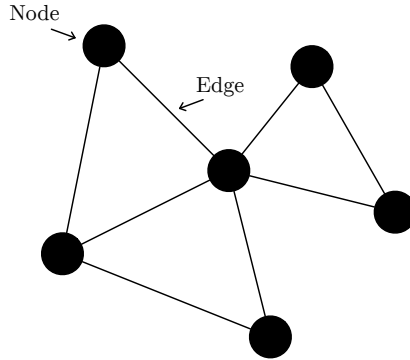


Figure 2.3: Network representation.

founded using almost the same structure (see Chapter 3). As such, using social network analysis, fraud experts are able to declare the foundation of the new employer unlawful and still recover the outstanding debts. Opinion fraud occurs when people untruthfully praise or criticize a product in a review. Especially online reviews lack control to establish the genuineness of the review. Matching people to their reviews and comparing the reviews with others using a network representation, enables review web sites to detect the illicit reviews.

Identity theft is a special form of social fraud, as introduced in Chapter 1, where an illicit person adopts another person's profile. An example of identity theft is the takeover of one's phone number. This is depicted in Figure 2.2. The light-colored node in the center is the true owner of the phone number, surrounded by his/her frequent contact list (contact C1-C6). The figure indicates that in an illicit takeover, the current contact list of a person is expanded with new contacts (dark nodes connected with a dashed line to owner), associated with a fraudster's previous account (dark node in the center). This comprises the fact that fraudsters often cannot withstand to call their family, friends, acquaintances, etc. The frequent contact list of the fraudster is a strong indicator for fraud.

While networks are a powerful visualization tool, they mainly serve to support the findings by automated detection techniques. We will focus on how to extend the detection process by extracting useful and

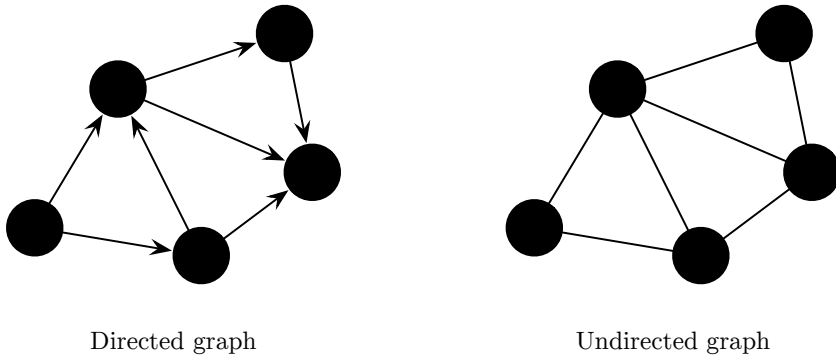


Figure 2.4: Example of a (un)directed graph.

meaningful features from the network. The network representation can be used afterwards to verify the obtained results.

2.1.2 Network components

This section will introduce the reader to graph theory, the mathematical foundation for the analysis and representation of networks.

Complex network analysis (CNA) studies the structure, characteristics and dynamics of networks that are irregular, complex and dynamically evolving in time (Boccaletti et al., 2006). Those networks often consist of millions of closely interconnected units. Most real-life networks are complex. CNA uses graph theory to extract useful statistics from the network. Boccaletti et al. (2006) define graph theory as the natural framework for the exact mathematical treatment of complex networks, and, they state that formally, a complex network is represented as a graph.

A graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ consists of a set \mathcal{V} of vertices or nodes (the points) and a set \mathcal{E} of edges or links (the lines connecting the points). This is illustrated in Figure 2.3. A node $v \in \mathcal{V}$ represents real-world objects such as people, computer, activities, etc. An edge $e \in \mathcal{E}$ connects two nodes in the network, and

$$e(v_1, v_2) | e \in \mathcal{E} \text{ and } v_i \in \mathcal{V}. \quad (2.1)$$

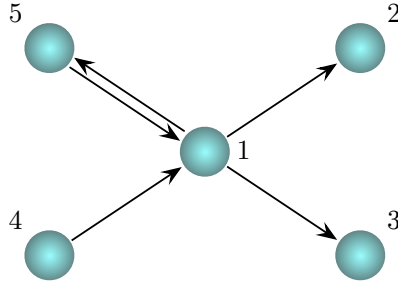


Figure 2.5: Follower-followee relationships in a Twitter network.

An edge represents a *relationship* between the nodes it connects, such as a friendship (between people), a physical connection (between computers), attendance (of a person to an event), etc. A graph where the edges impose an order or direction between the nodes in the network, is a directed graph. If there is no order in the network, we say that the graph is undirected. This is shown in Figure 2.4. The social network web site Twitter can be represented as a directed graph. Users follow other users, without necessarily being re-followed. This is expressed by the follower-followee relationships, and is illustrated in Figure 2.5. User 1 follows User 2, 3 and 5 (follower relationships), and is followed by User 4 and 5 (followee relationship). There is a mutual relationship between User 1 and 5.

In general, edges connect two nodes to each other. However, some special variants are sometimes required to accurately map the reality (see Figure 2.6):

- **Self-edge:** a self-edge is a connection between the node and itself. E.g., a person who transfers money from his/her account to another account s/he owns.
- **Multi-edge:** a multi-edge exists when two nodes are connected by more than one edge. E.g., in credit card transaction fraud, a credit card holder is linked to a merchant by a multi-edge if multiple credit card transactions occurred between them. Multi-edges will be discussed in Chapter 6.
- **Hyper-edge:** a hyper-edge is an edge that connects more than

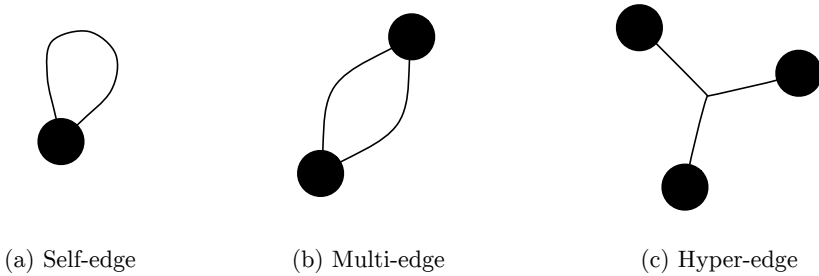


Figure 2.6: Edge representation.

one node in the network. E.g., three people who went to the same event.

A graph where the edges express the intensity of the relationships, is a weighted graph $\mathcal{G}^w(\mathcal{V}, \mathcal{E})$.

- **Binary weight:** This is the standard network representation. Here, the edge weight is either 0 or 1, and reflects whether or not a link exists between two nodes. An extension of the binary weighted graphs are the signed graphs where the edge weight is negative (-1), neutral (0) or positive (1). Negative weights are used to represent animosity, and positive weights are used to represent friendships. Neutral weights represent an “I don’t know you”-relationship.
- **Numeric weight:** A numeric edge weight expresses the affinity of a person to other persons s/he is connected to. High values indicate a closer affiliation. As people do not assign a weight to each of their contacts by themselves, many approaches are proposed to define an edge weight between nodes. A popular way is the *Common Neighbor* approach. That is, the edge weight equals the total number of common activities or events both people attended. An activity/event should be interpreted in a broad sense: the total number of messages sent between them, common friends, likes on Facebook, etc.
- **Normalized weight:** The normalized weight is a variant of the numeric weight where all the outgoing edges of a node sum up

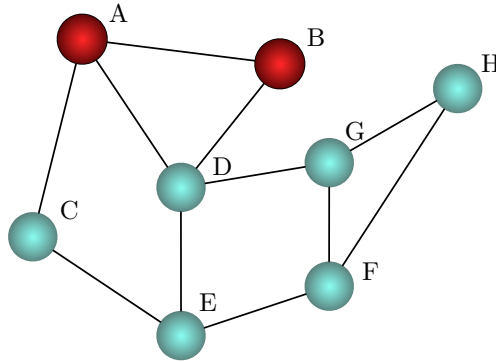


Figure 2.7: Example of a fraudulent network.

to 1. The normalized weight is often used in influence propagation over a network.

- **Jaccard weight:** The edge weight depends on how “social” both nodes are (Gupte and Eliassi-Rad, 2012), and

$$w_{(v_1, v_2)} = \frac{|\Gamma(v_1) \cap \Gamma(v_2)|}{|\Gamma(v_1) \cup \Gamma(v_2)|} \quad (2.2)$$

with $\Gamma(v_i)$ the number of events node v_i attended. For example, assume that person A attended 10 events and person B attended 5 events. They both went to 3 common events. Then, according to the Jaccard Index, their edge weight equals $1/4$.

Edge weights represent the connectivity within a network, and are in some way a measure of the sociality between the nodes in the network. Nodes, on the other hand, use labels to express the local characteristics. Those characteristics are mostly proper to the node itself and may include e.g. demographics, preferences, interests, beliefs, etc. When analyzing fraud networks, we integrate the fraud label of the nodes into the network. A node can be fraudulent or legitimate, depending on the condition of the object it represents. For example, Figure 2.7 shows a fraud network where legitimate and fraudulent people are represented by light- and dark-colored nodes respectively. Given this graph, we know that node A and B

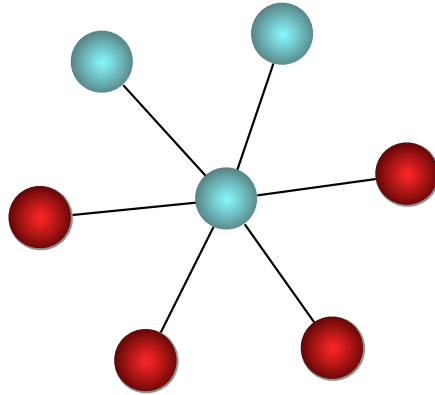


Figure 2.8: An egonet. The ego is surrounded by 6 alters of whom 2 are legitimate (light-colored) and 4 are fraudulent (dark-colored).

committed fraud beforehand. Node C is a friend of node A and is influenced by the actions of node A. On the other hand node D is influenced by both node A and B. A simple conclusion would be that node D has the highest propensity of perpetrating fraud, followed by node C.

While real-life networks often contain billions of nodes and millions of links, sometimes the direct neighborhood of nodes provides enough information to base decisions on. An *ego-centered network* or *egonet* represents the one-hop neighborhood of the node of interest. In other words, an egonet consists of a particular node and its immediate neighbors. The center of the egonet is the *ego*, and the surrounding nodes are the *alters*. An example of an egonet is illustrated in Figure 2.8. Such networks are also called the first-order neighborhood of a node. Analogously, the n -order neighborhood of a node encompasses all the nodes that can be reached within n hops from the node of interest.

2.1.3 Network representation

Transactional data sources often contain information about how entities relate to each other, e.g., call record data, bank transfer data, etc. An example transactional data source of credit card fraud is

Credit Card	Merchant	Merchant Category	Country	Amount	Time stamp	ACCEPT	FRAUD
82020922171246263	207005	056	USA	112.99	2014-11-06 00:28:38	TRUE	<i>No</i>
1887940000202544	105930	234	IRL	3.58	2014-11-06 00:28:40	TRUE	<i>No</i>
2070050002009251	79768	612	BEL	149.50	2014-11-06 00:28:47	TRUE	<i>No</i>
1809340000672044	11525	056	BEL	118.59	2014-11-06 00:28:49	FALSE	<i>No</i>
4520563752703209	323158	056	USA	22.27	2014-11-06 00:28:50	TRUE	<i>Yes</i>
5542610001561826	68080	735	FRA	50.00	2014-11-06 00:28:51	TRUE	<i>No</i>
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Figure 2.9: Example of credit card transaction data.

given in Figure 2.9. Each line in the transactional data source represents a money transfer between two actors: a credit card holder and a merchant. Despite the structured representation of the data, the relationships between credit card holders and merchants are hard to capture. Real-life data sources contain billions of transactions, making it impossible to extract correlations and useful insights. Network visualization tools offer a powerful solution to make information hidden in networks easy to interpret and understand. Inspecting the visual representation of a network can be part of the preprocessing phase as it familiarizes the user with the data and can often quickly result in some first findings and insights. In the post-processing phase, the network is a useful representation to verify the obtained results and understand the rationale. In general, a network can be represented in two ways:

- **Graphically**
- **Mathematically**

The graphical representation of a network, or *sociogram*, is the most intuitive and straightforward visualization of a network. A toy example of a credit card fraud network is shown in Figure 2.10. Credit card holders are modeled by rectangles, the merchants by circles. Solid (dashed) edges represent legitimate (fraudulent) transactions. Based on the figure, we expect that the credit card of user Y is stolen and that merchant 1 acts suspiciously. The sociogram can be used to present results at different levels in an organization: the operational, tactical and strategic management all benefit from

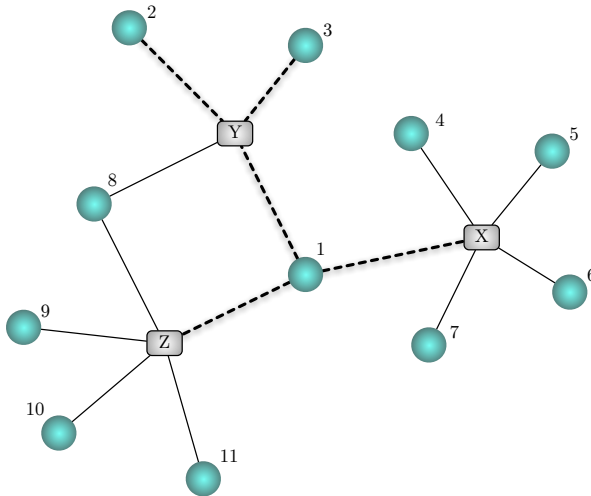


Figure 2.10: Toy example of credit card fraud.

interpreting the network representation by evaluating how to detect and monitor suspicious business processes (operational), how to act upon it (tactical) and how to deal with fraud in the future and take prevention measures (strategic).

While graphical network representations are mainly appropriate for visualization purposes, it is an unstructured form of data and cannot be used to compute useful statistics and extract meaningful characteristics. As a consequence, there is an urge to represent the network in a mathematically interesting way. The adjacency matrix and the adjacency list are two network representations that fulfill these requirements. The *adjacency* or *connectivity matrix* $\mathbf{A}_{(n \times n)}$ is a matrix of size $n \times n$ with n the number of nodes in the network; and $a_{(i,j)} = 1$ if a link exists between node i and j , and $a_{(i,j)} = 0$ otherwise. Figure 2.11a shows an example of a small undirected network. The corresponding adjacency matrix is depicted in Figure 2.11b. Remark that the adjacency matrix is a *sparse* matrix, containing many zero values. This is often the case in real-life situations. Social networks have millions of members, but people are only connected to a small number of friends - e.g., in 2012, Twitter had 500M users and each

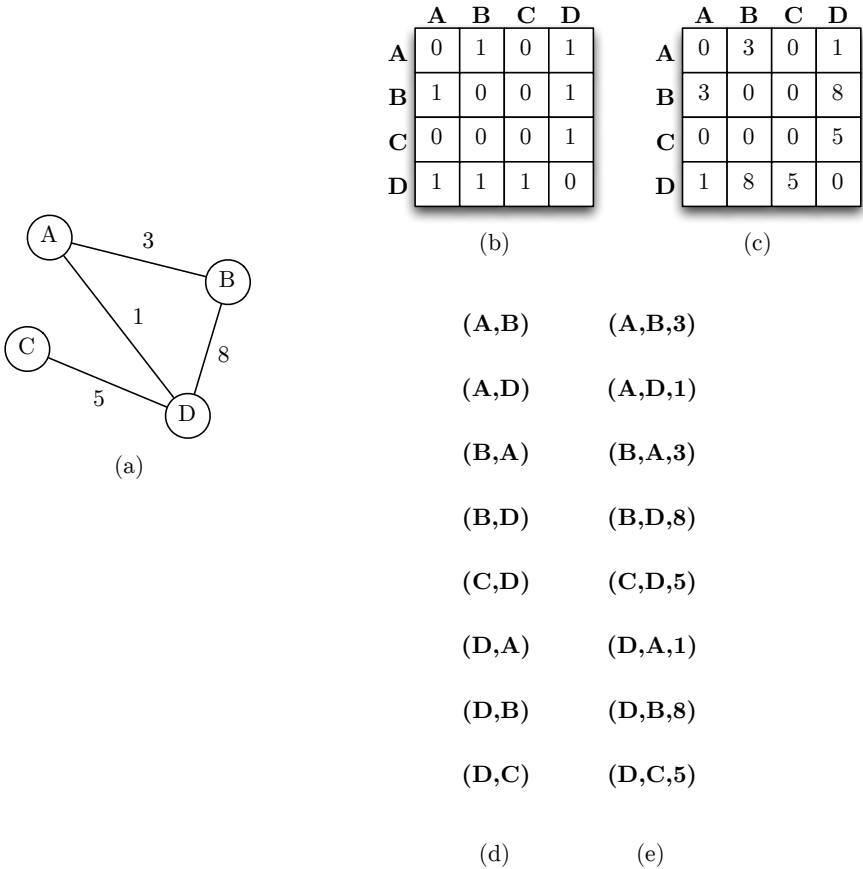


Figure 2.11: Mathematical representation of (a) a sample network : (b) the adjacency or connectivity matrix; (c) the weight matrix; (d) the adjacency list and (e) the weight list.

user follows approximately 200 other users.¹ The adjacency matrix of a undirected network, as presented here, is symmetric, whilst the adjacency matrix of a directed network is asymmetric. The adjacency matrix of a network records which nodes are connected to each other, irrespective of the edge weight. The *weight matrix* $\mathbf{W}_{(n \times n)}$ expresses the edge weight between the nodes of a network, and $w_{(i,j)} \in \mathbb{R}$ if a link exists between node i and j , and $w_{(i,j)} = 0$ otherwise. The weight matrix of the sample network is given in Figure 2.11c. The *adjacency list* is an abstract representation of the adjacency matrix, and provides a list of all the connections present in the network. A relationship between node v_i and node v_j is denoted as (v_i, v_j) . This is illustrated in Figure 2.11d. The *weight list* extends the adjacency list by specifying the weights of the relationships, and has the following format $(v_i, v_j, w_{(i,j)})$ with $w_{(i,j)}$ the weight between node v_i and node v_j (see Figure 2.11e).

2.2 Is fraud a social phenomenon? An introduction to homophily

One of the essential questions before analyzing the network regarding fraud, is deciding whether the detection models might benefit from CNA (Complex Network Analysis). In other words, do the relationships between people play an important role in fraud, and is fraud a contagious effect in the network? Are fraudsters randomly spread over the network, or are there observable effects indicating that fraud is a social phenomenon, i.e. fraud tends to cluster together. We look for evidence that fraudsters are possibly exchanging knowledge about how to commit fraud using the social structure. Fraudsters can be linked together as they seem to attend the same events/activities, are involved in the same crimes, use the same set of resources, or even are sometimes one and the same person (see also identity theft).

Homophily is a concept borrowed from sociology and boils down to the expression: “Birds of a feather flock together”. People have a strong tendency to associate with others whom they perceive as being

¹<http://news.yahoo.com/twitter-statistics-by-the-numbers-153151584.html>, retrieved on July 2015.

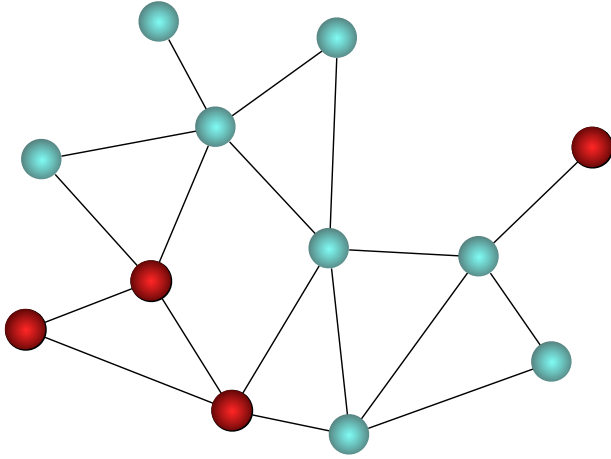


Figure 2.12: A homophilic network.

similar to themselves *in some way* (Newman, 2010). Friendships are mostly built because of similar interests, same origin, high school, neighborhood, hobbies, etc. or even the tendency to commit fraud. Relationships determine which people are influenced by whom and the extent to which information is exchanged.

A network is homophilic if nodes with label x (e.g., fraud) are to a larger extent connected to other nodes with label x . In marketing, the concept of homophily is frequently exploited to assess how individuals influence each other, and to determine which people are likely responders and should be targeted with a marketing incentive. For example, if all John's friends are connected to telecom provider *Beta*, John is likely to sign the same contract with provider *Beta*. A network that is not homophilic, is heterophilic.

The same reasoning holds in fraud. We define a homophilic network as a network where fraudsters are more likely to be connected to other fraudsters, and legitimate people are more likely to be connected to other legitimate people.

Advanced network techniques take into account the time dimension. Few fraudulent nodes that are popping up together in the network might indicate a newly originating web of fraud, while subgraphs characterized with many fraudulent nodes are far-evolved structures.

Preventing the growth of new webs and the expansion of existing webs are important challenges that both need to be addressed in the fraud detection models.

We already showed that a graphical representation might give a first indication of the homophilic character of the network, and thus whether network analysis might make sense in the fraud detection task at hand. Mathematically, a network is homophilic if fraudulent nodes are significantly more connected to other fraudulent nodes, and as a consequence, legitimate nodes connect significantly more to other legitimate nodes. More concretely, let l be the fraction of legitimate nodes in the network and f the fraction of fraudulent nodes in the network, then $2lf$ is the expected probability that an edge connects two dissimilar labeled nodes. These edges are called *cross-labeled edges*. A network is homophilic if the observed fraction of cross-labeled edges \hat{r} is significantly less than the expected probability $2lf$, i.e. if the null hypothesis

$$H_0 : \hat{r} \geq 2lf \tag{2.3}$$

can be rejected. Consider Figure 2.12. The dark-colored (light-colored) nodes are the fraudsters (legitimate people). The network consists in total of 12 nodes: 8 legitimate nodes and 4 fraudulent nodes. The fraction l and f equal $\frac{8}{12}$ and $\frac{4}{12}$ respectively. In a random network, we would expect that $2lf = 2 \cdot \frac{8}{12} \cdot \frac{4}{12} = \frac{8}{18}$ edges are cross-labeled. The network in Figure 2.12 has 5 cross-labeled edges, and 3 fraud and 10 legit same-labeled edges. The observed fraction of cross-labeled edges is thus $\hat{r} = \frac{5}{18}$. We expect to see 8 edges in the network that are cross-labeled, instead of the 5 edges we observe. The null hypothesis H_0 is rejected with a significance level of $\alpha = 0.05$ (p -value of 0.02967) using a one-tailed proportion test with a normal approximation. The network is homophilic.

Other measures to assess whether there are significant patterns of homophily present in the network include *dyadicity* and *heterophilicity* (Park and Barabási, 2007). In many systems the number of links between nodes sharing a common property is larger than if the characteristics were distributed randomly in the network. This is the dyadic effect. For a network where the labels can only take

two values, **1** (Fraud) and **0** (Legitimate), let $n_1(n_0)$ be the number of fraudulent (legitimate) nodes and $N = n_0 + n_1$. Now, we can define three types of dyads: (**1** - **1**), (**1** - **0**), and (**0** - **0**), indicating the label (**Fraud** - **Fraud**), (**Fraud** - **Legitimate**) and (**Legitimate** - **Legitimate**) of two end points connected by a link. The total number of dyads of each kind are represented as m_{11} , m_{10} and m_{00} respectively, and $M = m_{11} + m_{10} + m_{00}$. If nodes are randomly connected to other nodes regardless of their labels, then the expected values of m_{11} and m_{10} equal:

$$\bar{m}_{11} = \binom{n_1}{2} p = \frac{n_1(n_1 - 1)p}{2} \quad (2.4)$$

$$\bar{m}_{10} = \binom{n_1}{1} \binom{n_0}{1} p = n_1(N - n_1)p \quad (2.5)$$

with $p = \frac{2M}{(N(N-1))}$ the connectance, representing the probability that two nodes are connected. If $p = 1$, all nodes in the network are connected to each other. Dyadicity and heterophilicity can then be defined as:

$$D = \frac{m_{11}}{\bar{m}_{11}} \quad (2.6)$$

$$H = \frac{m_{10}}{\bar{m}_{10}} \quad (2.7)$$

A network is dyadic if $D > 1$, indicating that fraudulent nodes tend to connect more densely among themselves than expected for a random configuration. A network is heterophobic (opposite of heterophilic) if $H < 1$, meaning that fraudulent nodes have fewer connections to legitimate nodes than expected at random (Park and Barabási, 2007). The network represented in Figure 2.12 is dyadic and heterophobic. If a network is dyadic and heterophobic, it exhibits homophily; a network that is anti-dyadic and heterophilic, is inverse-homophilic.

A network that exhibits evidence of homophily, is worthwhile to investigate more thoroughly. For each instance of interest, we extract features that characterize the instance based on its relational structure.

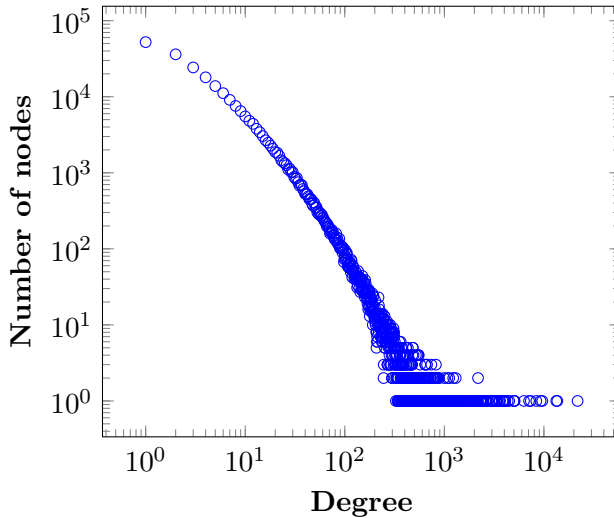


Figure 2.13: Illustration of the degree distribution for a real-life network of social security fraud. The degree distribution follows a power law (log-log axes).

2.3 Overview of the featurization process

In this section, we will discuss the main metrics to measure the impact of the social environment on the nodes of interest. In general, we distinguish between three types of analysis techniques:

- **Neighborhood metrics**
- **Centrality metrics**
- **Collective Inference algorithms**

Neighborhood metrics characterize the *target of interest* based on its direct associates. The n -order neighborhood around a node consists of the nodes that are n hops apart from that node. Due to scalability issues, many detection models integrate features derived from the egonet or first-order neighborhood (see Section 2.1.2). That is, the node and its immediate contacts. Neighborhood metrics that are discussed throughout this dissertation are degree, triangles, den-

Metric	Description
<i>Degree</i>	Number of connections of a node (in- versus out-degree if the connections are directed).
<i>Triangles</i>	Number of fully connected subgraphs consisting of three nodes.
<i>Density</i>	The extent to which nodes in a network or subgraph are connected to each other, and $d = \frac{2M}{N(N-1)} \tag{2.8}$ with d the density, M the number of edges and N the number of nodes in the (sub)graph.
<i>Relational Neighbor</i>	Relative number of neighbors that belong to class c (e.g., to class fraud). $P(c n) = \frac{1}{Z} \sum_{\{n_j \in \text{Neighborhood}_n \text{class}(n_j)=c\}} w(n, n_j) \tag{2.9}$ whereby Neighborhood_n represents the neighborhood of node n , $w_{(i,j)}$ the weight of the connection between n and n_j , and Z is a normalization factor to make sure all probabilities sum up to 1.
<i>Probabilistic Relational Neighbor</i>	Probability to belong to class c given the posterior class probabilities of the neighbors. $P(c n) = \frac{1}{Z} \sum_{\{n_j \in \text{Neighborhood}_n\}} w(n, n_j) P(c n_j) \tag{2.10}$ whereby Neighborhood_n represents the neighborhood of node n , $w_{(i,j)}$ the weight of the connection between n and n_j , and Z is a normalization factor to make sure all probabilities sum up to 1.

Table 2.1: Overview of neighborhood metrics.

sity, relational neighbor and probabilistic relational neighbor. The neighborhood metrics are summarized in Table 2.1.

The degree distribution of a network describes the probability distribution of the degree in the network. The degree distribution in real-life networks follows in general a *power law*. That is, many nodes are only connected with few other nodes while only few nodes in the network link to many other nodes. Figure 2.13 gives an example of the degree distribution (log-log scale) of a real-life fraud network of a social security institution (Van Vlasselaer et al., under review).

Centrality metrics quantify the importance of an individual in a social network (Boccaletti et al., 2006). Centrality metrics are

Metric	Description
Geodesic path	Shortest path between two nodes in the network.
Closeness	The average distance of a node to all other nodes in the network (reciprocal of farness). Given a network with n nodes, the mean geodesic distance or farness $g(v_i)$ from a node i to the other nodes is computed as follows $\left[g(v_i) = \frac{\sum_{j=1(j \neq i)} d(v_i, v_j)}{n-1} \right]^{-1}. \quad (2.11)$
Betweenness	Counts the number of times a node or connection lies on the shortest path between any two nodes in the network. Let g_{jk} be the number of shortest paths between node j and node k , and $g_{jk}(v_i)$ the number of shortest paths between node j and node k that pass through node v_i , then the betweenness becomes $\sum_{j < k} \frac{g_{jk}(v_i)}{g_{jk}}. \quad (2.12)$
Graph theoretic center	The node with the smallest maximum distance to all other nodes in the network .

Table 2.2: Overview of centrality metrics.

typically extracted based on the whole network structure, or a subgraph. Table 2.2 comprises geodesic paths, betweenness, closeness and the graph theoretic center. As these metrics lack scalability, they are not further used in this dissertation.

Given a network with known fraudulent nodes, how can we use this knowledge to infer a primary fraud probability for all the unlabeled nodes (i.e., the currently legitimate nodes)? As opposed to neighborhood and centrality metrics, *collective inference* (CI) algorithms compute the probability that a node is exposed to fraud and thus the probability that fraud influences a certain node. In CI, the label of a node is said to depend on the label of the neighboring nodes. A change in one node's label, might cause the labels of the neighboring nodes to change which might impact the label of their neighbors, and so on. As a result, long-distance propagation is possible (Hill et al., 2007). We consider PageRank, and briefly explain Gibbs sampling, iterative

classification, relaxation labeling, and loopy belief propagation.

2.3.1 PageRank

The PageRank algorithm was introduced by Page and Brin in 1999 and is the basis of Google's famous search engine algorithm for ranking web pages (Page et al., 1998). The PageRank algorithm tries to simulate surfing behavior. Specifically, the main idea is that important web pages (i.e., web pages that appear at the top of the search results) have many incoming links from other (important) web pages. The ranking of a web page depends on (a) the ranking of web pages pointing towards that web page, and (b) the out-degree of the linking web pages. However, visiting web pages by following a random link on the current web page is not a realistic assumption. Surfers' behavior is more random: instead of following one of the links on a web page, they might randomly visit another web page. Therefore, the PageRank algorithm includes the *random surfer* model which assumes that surfers often get bored, and randomly jump to another web page. With a probability of α the surfer will follow a link on the web page s/he is currently visiting. However, with a probability $1 - \alpha$, the surfer visits a random other web page. The PageRank algorithm is expressed as follows:

$$PR(A) = \alpha \sum_{i \in N_A} \frac{PR(i)}{d_{out,i}} + (1 - \alpha) \cdot e_A \quad (2.13)$$

with $PR(i)$ the ranking of web page i , $D_{out,i}$ the out-degree of web page i , $(1 - \alpha)$ the restart probability, and e_A the restart value for web page A which is often uniformly distributed among all web pages. This equation requires the ranking of the neighboring web pages. One option is to start with a random page rank value for every web page and iteratively update the page rank scores until a predefined number of iterations is reached or a stopping criterion is met (e.g., when the change in the ranking is marginal).

In Chapter 3, we start from Equation 2.13 to develop a propagation algorithm that captures the extent to which fraud influences through the network. For each node of the network, we derive an

exposure score, which tells how much the node is exposed to fraud.

2.3.2 Gibbs Sampling

Gibbs sampling (Geman and Geman, 1984) is a collective inference procedure that uses a local classifier to infer a posterior class probability in order to initialize the node labels in the network. More concretely, the original semi-labeled graph is transformed in a (fully) labeled graph by sampling the posterior probabilities of the local classifier. The predictive features of a local classifier consist of non network-based variables. An iterative procedure continually updates the expected class labels of the unknown nodes. The first $iter_b$ steps of the procedure approach a stationary distribution. This is the so-called burn-in period, where no statistics are recorded. During the last $iter_c$ steps, the algorithm keeps track of which class labels are assigned to each node. The final class probability estimate is computed as the normalized count of the number of times each class is assigned to a particular node.

2.3.3 Iterative Classification Algorithm

Like Gibbs sampling, the iterative classification algorithm (ICA) initializes the semi-labeled graph by using a local classifier (Lu and Getoor, 2003). Based on the local model's output, the most probable class label is assigned to each unknown node. This is the bootstrap phase. During the iteration phase, a relational learner updates the class labels of each unknown node based on the outcome of a relational logistic regression model. The input features are computed as link statistics of the current label assignments. Link statistics include e.g., *mode* (most occurring label of the neighboring nodes), *count* (number of neighboring fraud nodes), *binary* (at least one of the neighboring nodes are fraudulent). Nodes that are not yet classified are ignored. A new class label is assigned to each unknown node based on the largest posterior probability. This step is repeated until a stopping criterion is met. The final class label corresponds to the class label estimate generated during the last iteration.

2.3.4 Relaxation labeling

Relaxation labeling starts from a local classifier to initialize a node's class label. Previous approaches assigned a hard label (i.e., either legitimate or fraud) to each node. Relaxation labeling starts with assigning to each node a probability that indicates the likelihood of a node to belong to a certain class. This is soft labeling. Next, the probability class labels are used to iteratively update the class probability using a relational model. The class estimates of the last iteration are the final class label estimates.

2.3.5 Loopy belief propagation

Loopy belief propagation is a collective inference procedure based on iterative message passing (Pearl, 1986; Yedidia et al., 2003). The main idea is that the belief of each node to be in state x (let's say fraud) depends on the messages it receives from its neighbors. The belief of a node to be in state x is the normalized product of the received messages. The message as well as the belief is continuously updated during the algorithm.

2.4 Conclusion

This chapter is an introductory chapter to familiarize the reader with network analysis, and the opportunities it might open up. We discussed the main components of a network, as well as the different representation possibilities. We elaborate on how a network can be represented for (a) visualization purposes, especially in the pre- and post-processing phase of model development, and (b) in a mathematically interesting manner in order to derive useful statistics and meaningful features from the network in a scalable way. In addition, we contrast the various options to decide upon the weight of edges which are able to quantify the intensity of relationships. The concept homophily is introduced, being a measure to express the extent to which nearby social neighbors are alike. In this chapter, homophily is mainly approached from a fraud perspective, so to serve as a primary indication whether a fraud detection model might benefit from network analysis. This chapter is concluded by entering into the featurization

process. The featurization process defines how unstructured network information can be mapped into a set of structured features. We discussed neighborhood and centrality metrics, and briefly introduced collective inference algorithms.

Chapter 3

GOTCHA! A network-driven approach for fraud detection

In this chapter, we study the impact of network information for social security fraud detection. In a social security system, companies have to pay taxes to the government. This study aims to identify those companies that intentionally go bankrupt in order to avoid contributing their taxes. We link companies to each other through their shared resources, as some resources are the instigators of fraud. We introduce GOTCHA!, a new approach on how to define and extract features from a time-weighted network, and how to exploit and integrate network-based and intrinsic features in fraud detection. The GOTCHA! propagation algorithm diffuses fraud through the network, labeling the unknown and anticipating future fraud whilst simultaneously decaying the importance of past fraud. We find that domain-driven network variables have a significant impact on detecting past and future frauds, and improve the baseline by detecting up to 55% additional fraudsters over time.

3.1 Introduction

Fraud detection is a research domain with a wide variety of different applications and different requirements, including credit card fraud

(Chan and Stolfo, 1998; Quah and Sriganesh, 2008; Sánchez et al., 2009), call record fraud (Fawcett and Provost, 1997), money laundering (Gao and Ye, 2007; Jensen, 1997), insurance fraud (Dionne et al., 2009; Furlan and Bajec, 2008; Phua et al., 2004) and telecommunications fraud (Hilas and Sahalos, 2005; Estévez et al., 2006). The aforementioned problems generally exhibit the same characteristics, but the solution to each problem is rather domain-specific (Chandola et al., 2009). Data mining techniques – i.e., finding patterns and anomalies in large amounts of data – have already proven useful in risk evaluation (Baesens et al., 2003a,b), but fraud is an atypical example and requires built-in domain knowledge.

We introduce GOTCHA!, a new, generic, scalable and integrated approach on how (social) network analytics can improve the performance of traditional fraud detection tools in a social security context. We identify five challenges that concur with fraud. That is, fraud is an *uncommon, well-considered, time-evolving, carefully organized and imperceptibly concealed* crime that appears in many different types and forms. Whereas current research fails to integrate all these dimensions into one encompassing approach, GOTCHA! is the first to address each of these challenges together in one high-performance, time-dependent detection technique.

In short, GOTCHA! contributes to the fraud detection domain by proposing a novel approach on how to spread fraud through a (i) time-weighted network and features extracted from a (ii) bipartite graph (cfr. *infra*). We exploit dynamic network-based features derived from the direct neighborhood, and develop a new propagation algorithm that infers an initial exposure score for each node using the whole network. The exposure score measures the extent to which a node is influenced by fraudulent nodes. We integrate both intrinsic and network-based features into one scalable algorithm. We argue that fraud is a time-dependent phenomenon, and as a consequence GOTCHA! is designed such that a subject’s characteristics and fraud probability can change over time.

We test the validity of our approach on a real data set obtained from the Belgian social security institution, which registers and moni-

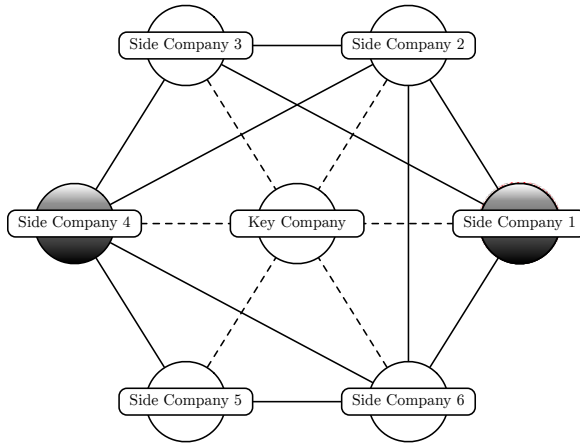


Figure 3.1: Example of a spider construction. Company 1 and 4 are fraudulent. Resources are transferred towards other companies (solid line). The key company organizes the fraudulent setup, but its links to other companies are hidden (dashed line).

tors every active company in Belgium and keeps track of all resources, and their associations with companies.¹ In a social security system, companies have to pay employer and employee contributions to the government. Fraud occurs when companies *intentionally* go bankrupt in order to avoid paying these taxes. A new/existing company with (partly) the same structure is founded afterwards and continues the activities of the former company. We can compare the structures of companies through their resources.

A spider construction is a fraudulent setup with an active exchange of resources between the companies, i.e., fraudulent companies do not transfer all of their resources to only one other company as this might attract too much attention (see Figure ??). They rather distribute their resources among many companies. Active companies that inherit resources from fraudulent companies, exhibit a high risk of perpetrating fraud themselves. In particular, we distinguish between the key and side companies. The *side*

¹Due to confidentiality issues, we will not elaborate further upon the exact type of resources, but the reader can understand shared resources in terms of the same address, equipment, buyers, suppliers, employees, etc.

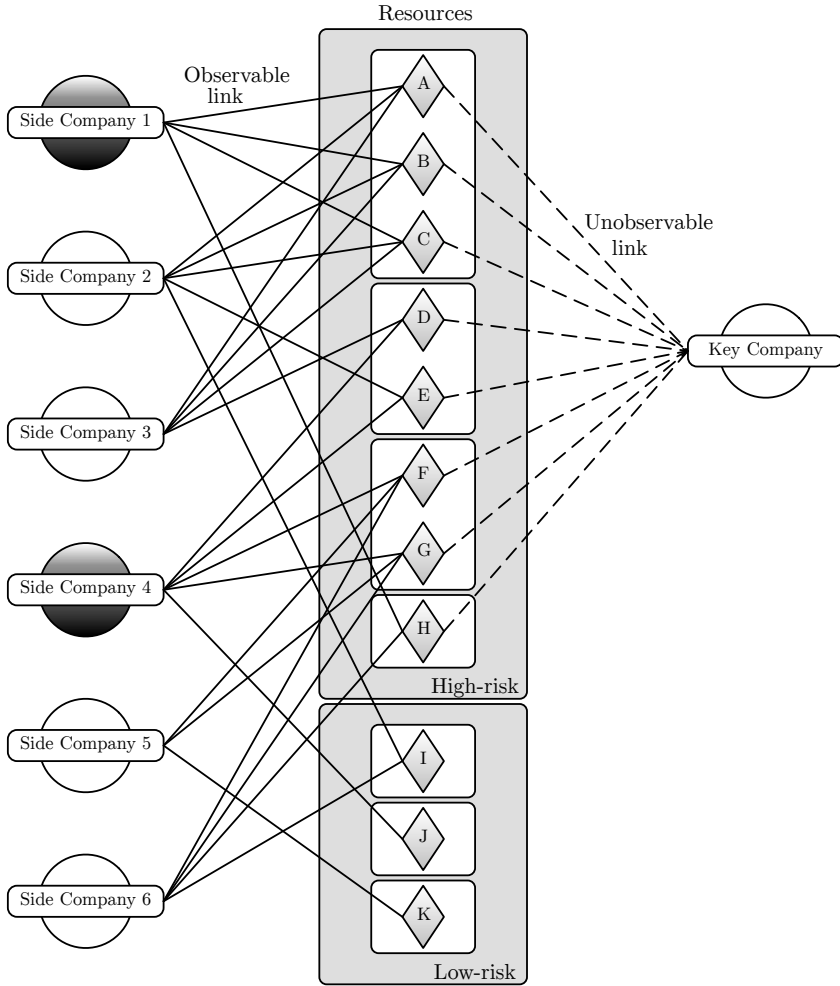


Figure 3.2: Bipartite graph of a spider construction. Companies are indirectly connected to each other through the resources.

companies are the perpetrators of the fraud and have an observable link to each other through shared resources. The core of a spider construction is the *key company*, which is responsible for organizing the fraud, setting up many side companies and pruning away their profits, so that they go bankrupt. However, the key company has unobservable links, and therefore we can only detect the side companies. The main goal of *GOTCHA!* is to exploit the associations between companies and their resources to infer which companies have a high risk to commit fraud in the future. We believe that network-based knowledge might strongly improve the standard approaches, which only use intrinsic variables in the detection models.

In order to assess the added value of our approach, we compare *GOTCHA!* to three baselines: (1) an intrinsic model, only including intrinsic features; (2) a unipartite model, linking companies directly together by means of the resources they shared or transferred among each other; (3) a bipartite model, which starts from the same network representation as our proposed approach, integrating both companies and resources (see Figure 3.2). Yet, the model is not time-weighted. Our results show that an optimal mix between intrinsic and time-weighted network-based attributes contribute to a higher accuracy and more precise output than the baselines. Moreover, it appears that many regular (i.e., non-intentional) bankruptcy companies are also outputted and classified as high risk. This is a strong indication that the developed approach is also able to find those companies that committed fraud, but were not caught in the past. As a result, we argue that our approach is suitable for both *future* and *retrospective* fraud detection.

This chapter is organized as follows: Section 3.2 motivates *GOTCHA!*'s fraud detection process and framework, as well as *GOTCHA!*'s contributions to existing research. Section 3.3 focuses on how network analysis is implemented for fraud detection. This section also discusses *GOTCHA!*'s propagation algorithm and how domain-driven networked features are defined and extracted from the network. Section 3.4 summarizes the modeling approach. Section 3.5 contains the results of *GOTCHA!* on social security fraud data. Section 3.6

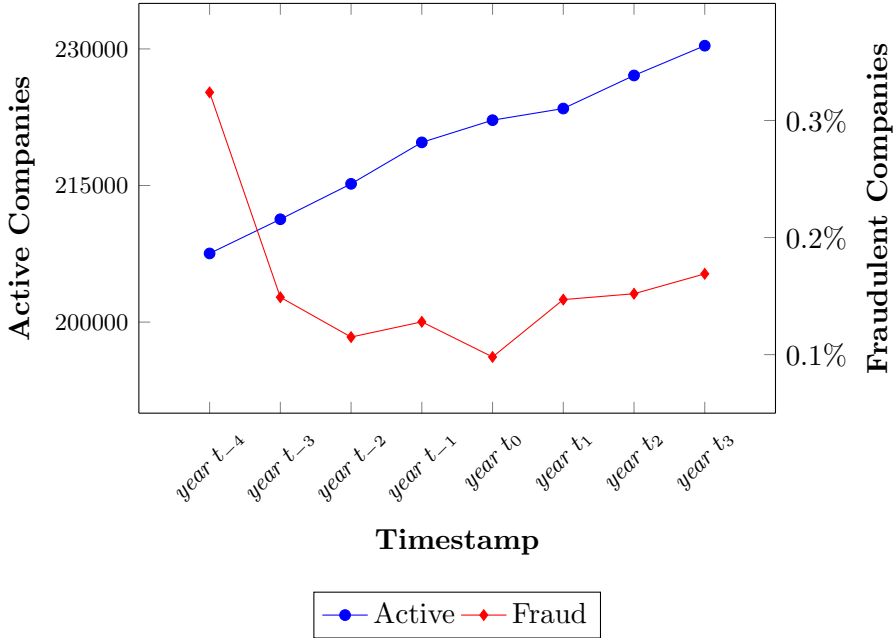


Figure 3.3: Overview of the total number of active companies (blue curve) and fraudulent companies (red curve). The number of active companies is consistently growing. A similar trend can be noticed in the number of fraudulent companies.

concludes this chapter.

3.2 Social Security Fraud Detection

3.2.1 Background

The Belgian Social Security Institution is a federal agency that monitors the tax contributions of every active company in Belgium. These contributions are used to fund the various branches in social security, such as family allowance funds, unemployment funds, health insurance, holiday funds, etc. Figure 3.3 gives an overview of the total number of active companies across the different years of analysis.²

Companies – or in general terms, the employers – need to pay

²Due to a non-disclosure agreement, we do not specify the exact time stamp.

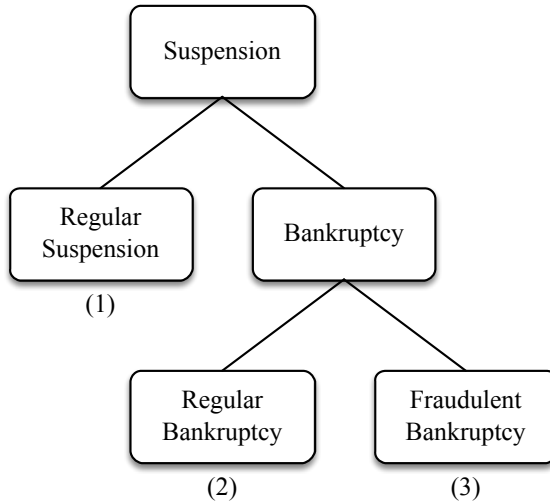


Figure 3.4: Overview of the different stages a company can go through when ending its economic lifecycle. Even though it is hard to detect fraud *ex ante*, it is also a challenging task to define fraud *ex post*. Bankruptcy corresponds to the disability of paying back debts to the social security institution. It is not straightforward which companies are evidence of regular economic failure and which companies went bankrupt due to some fraudulent structure.

employer and employee contributions to the government. This means that all payments are deducted by the company and passed on to the government. It is the employer's responsibility rather than that of the employee to fulfill the social contributions. Some companies, nevertheless, fail to redeem their obligations and file for bankruptcy. Recently, experts found evidence of fraudulent setups through bankruptcy. We say that if a company *intentionally* goes bankrupt so as not to pay its tax contributions, the company is fraudulent.

Although it is easy to formulate a definition for fraud, there are two main issues the social security institution is facing: *ex post* and *ex ante* fraud detection. First, it is not an obvious task to identify suspicious companies *ex ante*, or before the actual crime has been committed. Experts can closely follow up companies with a high risk of not paying off their taxes. As most of the controlling services oper-

ate manually, it is a challenging task to process the massive amounts of data and identify the anomalies. Using their field experience, experts almost never classify cases wrong. On the other hand, each year only a small part of all companies can be investigated by the experts, leaving a lot of future fraudulent companies unnoticed. Secondly, all fraudulent companies detected by subject matter experts are currently identified *ex post*. This means that the companies are already bankrupt with unrecoverable debts to the government. In general, there are three scenarios when a company decides to stop its activities: (1) regular suspension, (2) regular bankruptcy and (3) fraudulent bankruptcy. This is depicted in Figure 3.4. More specifically, a company is regularly suspended when it stops its economical activities and all remaining transactions are finalized. There are no outstanding debts. A company that is categorized as suspended by regular bankruptcy, however, did not succeed to pay back to all its creditors. When we say that a company ended its lifecycle by fraudulent bankruptcy, the company intentionally did not redeem its debts. It is this last category that is the subject of our fraud detection approach. Remark that it is especially hard to discriminate between regular and fraudulent bankruptcies, even for subject matter experts. Subject matter experts investigate suspicious bankruptcies and label them fraudulent as soon as they detect some abnormal activities. While those experts can accurately classify a bankruptcy as fraudulent or not, many fraudulent bankruptcies are not detected as the number of bankruptcies to investigate is too large. Experts require thus a detection tool that guides them towards potential high-risk companies.

Spider Constructions

We defined fraud as the intentional failure of a company to redeem its tax contributions. In real data, we observe small and dense “webs of fraud”, the so-called *spider constructions*. In addition to transparent forms of fraud – like systematically neglecting the legal registration of employees, a spider construction is a more complex type of tax evasion, involving many companies and people, and it is not obvious for human analysts to detect. Stating that fraud is rather a social than an individual phenomenon, communicated and encouraged

by the presence of other individuals who also wish to commit fraud (Neville et al., 2005), such theoretical constructions confirm the effect of social interactions in fraud. More concretely, a spider construction consists of (fraudulent) companies that are closely connected to each other through shared or transferred resources. Resources include address, equipment, buyers, suppliers, employees, etc.³ For example, two companies are associated with each other because they operate at the same location. The data reveals which resource is associated with which company for which specific time period. We observe that the profits of companies that belong to a fraudulent setup are often pruned away by a hidden key company (see Figure 3.1). Consequently, the company becomes insolvent and files for bankruptcy, leaving the government with unrecoverable debt claims. We see, however, that their operational resources move towards other currently legitimate or newly founded companies, e.g., 80% of the resources of the fraudulent company are re-used by a new or currently legitimate company. Those companies will continue the activities of the fraudulent company. The transfer (or sharing) of such resources induces the observable structure of spider constructions. Companies that inherit (many) resources of fraudulent companies, exhibit a high risk of perpetrating fraud in the future as well. Figure 3.2 shows how (groups of) resources are exchanged between various companies, transferring fraudulent knowledge on how to commit fraud (Levin and Cross, 2004) towards legitimate companies. We must note that *resource sharing* is nevertheless a normal activity in the corporate environment, complicating the detection process. Although the exact procedure of resource sharing is confidential, the reader can think in terms of e.g., the transfer or sharing of employees, equipment, buyers/suppliers, and addresses taken over by other employers, etc.

Governmental authorities do not have the necessary information to link the key company to its side companies, or even to identify the key company in each spider construction. We focus on the detection of the side companies. Nevertheless, the role of side companies is crucial in levying social security taxes. They are often found for a short term, first operating in a regular way, but by the time they have

³Due to a non-disclosure agreement, we cannot provide detailed information about the resources.

to pay taxes, they intentionally go bankrupt, leaving a large debt to the social security institution.

The requirements of fraud experts are threefold: (1) curtailing the growth of existing spider constructions; (2) preventing the development of new spider constructions; and (3) detecting uncaught spider constructions, i.e., dense subgraphs in the network with many bankruptcies. In this chapter, we focus on requirement (1) and (2). Recall that we do not have information to associate key companies to their side companies. Therefore, we aim to find suspicious side companies.

3.2.2 Challenges

A first contribution of this research is the investigation and identification of the underlying reasons why fraud detection cannot be resolved by applying standard data analytics. We identify five challenges present in most fraud detection problems, and discuss how each challenge can be addressed. In general, the main challenges that characterize fraud are as follows:

Definition 3.1. Fraud is an *uncommon, well-considered, time-evolving, carefully organized and imperceptibly concealed* crime which appears in many different types and forms.

I. Uncommon Fraud detection techniques must deal with extremely skewed class distributions. Subject matter experts are often only able to identify a limited number of confirmed fraud cases. Rather than using unsupervised techniques, how can we use and learn from (sparsely) labeled data? Resampling techniques (Provost, 2000; Chawla et al., 2011) are able to emphasize fraud and rebalance the data set.

Figure 3.3 depicts the number of active companies over 8 years (blue curve) and the percentage of fraudulent companies over the same time period (red curve) for the social security institution in our study.⁴ Each year, approximately 230K companies are active with a fraud ratio between 0.09% and

⁴Due to confidentiality issues, the exact date of each timestamp is omitted.

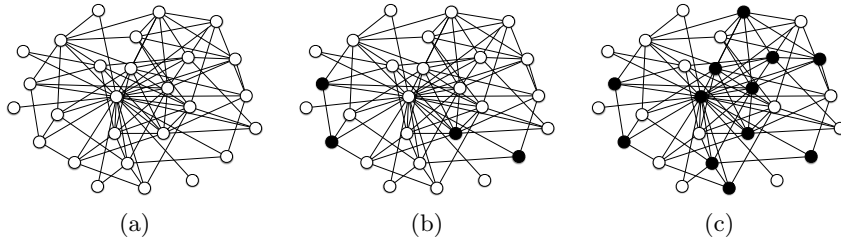


Figure 3.5: Real-life example of fraud propagating through a sub-network over time. Legitimate companies are unfilled, fraudulent companies are filled. The initial situation is represented in (a). When time passes, more nodes are influenced by fraudulent behavior of their neighbors (b), ultimately infecting almost the whole subgraph (c). This confirms the contagious effect of fraud.

0.18%, except for year t_{-4} .⁵ For reasons of stability, GOTCHA! is applied to year $t_0 - t_3$.

II. Well-considered Complex fraud structures are carefully planned and well thought through. Fraud is present in all attributes. Labeling instances based on a single action (e.g., outlier detection) is often inaccurate and insufficient. We believe that integrating intrinsic and domain-driven network attributes helps to improve model performance.

III. Time-evolving Fraud evolves over time. Fraudsters learn from the mistakes of their predecessors and are highly adaptive (Jensen, 1997). Models should be built for a varying temporal granularity, weighing information based on its recency (Rossi and Neville, 2012). We estimate models for different timestamps, resulting in a time-dependent fraud probability.

IV. Carefully organized Fraudsters often do not operate by themselves, but are influenced by close allies and influence

⁵During *year* t_{-4} a fraud detection team was assigned and experts effectively started to report fraud. The peak in fraud detection is mainly due to catching up the piling backlog of old fraud cases and entering them in the system.

others in turn. They transfer knowledge on how to commit fraud without being detected. This is *homophily*. Homophily states that instances that are closely related to each other are likely to behave in the same way (Aral et al., 2009; Bapna and Umyarov, 2012). A feasibility study (Park and Barabási, 2007; Easley and Kleinberg, 2010) on the social security data set indicates that fraudulent companies are indeed significantly more connected to other fraudulent companies (p -value ≤ 0.02 for $t_0 - t_3$ using a one-tailed proportion test, see Section 2.2).

V. Imperceptibly concealed Maes et al. (2002) formulated this as the presence of overlapping data. Fraudulent companies often have the same characteristics as legitimate companies. In the fraud detection domain, there is a need for extracting additional, meaningful features that uncover hidden behavior. We focus on influence. Influence is subtle and often subliminal. This challenge encompasses how to capture unobservable, subtle fraudulent influences from the external environment. We address this challenge by means of collective inference procedures, like network propagation techniques, to diffuse a small amount of fraudulent behavior through the network and infer a fraud *exposure score* for every node in the network.

Figure 3.5 illustrates how fraud spreads through a network over time, much like a virus. The closer the nodes are located to the region of a fraudulent source, the higher the probability of copying the fraudulent behavior. This phenomenon is known as the *propagation* effect (Prakash et al., 2010).

Sections 3.3, 3.4 and 3.5 of this chapter explain in more detail how we address each of these challenges. In particular, Section 3.3.3 describes how we infer an initial exposure score for every company, and consequently label the unknown resources based on fraudulent influences from the whole network (*Challenge V*). In Section 3.3.4, each company is then featurized based on its direct resources (*Challenge IV*). Section 3.4 discusses how we integrate intrinsic and network-based features (*Challenge II*) and resample the data set using SMOTE (Chawla et al., 2011) (*Challenge I*). The proposed fraud detection

#	Reference	Fraud type	Challenges				
			I	II	III	IV	V
1	(Goldberg and Senator, 1995)	<i>money laundering</i>				X	
2	(Jensen, 1997)	<i>money laundering</i>				X	
3	(Cortes et al., 2001)	<i>telecom fraud</i>			X	X	
4	(Chen et al., 2004b)	<i>insurance fraud</i>				X	
5	(Galloway and Simoff, 2006)	<i>law enforcement fraud</i>				X	
6	(Neville et al., 2005)	<i>security fraud</i>	X		X	X	
7	(Fast et al., 2007)	<i>security fraud</i>	X		X	X	
8	(Wang and Chiu, 2008)	<i>online auction fraud</i>			X	X	
9	(Akoglu et al., 2010)	<i>various</i>				X	
10	(Yanchun et al., 2011)	<i>online auction fraud</i>				X	
11	(Chiu et al., 2011)	<i>online auction fraud</i>				X	
12	(Chau et al., 2006)	<i>online auction fraud</i>		X		X	X
13	(Pandit et al., 2007)	<i>online auction fraud</i>			X	X	X
14	(Gallagher et al., 2008)	<i>various</i>				X	X
15	(McGlohon et al., 2009)	<i>accounting fraud</i>				X	X
16	(Šubelj et al., 2011)	<i>insurance fraud</i>		X		X	X
17	(Akoglu et al., 2013)	<i>opinion fraud</i>				X	X
18	GOTCHA!	<i>social security fraud</i>	X	X	X	X	X

Table 3.1: Overview of all published papers related to fraud detection using network analytics.

technique estimates time-weighted features and a time-dependent fraud probability for every company (*Challenge III*), which is explained in Section 3.5.

3.2.3 Related Work

Although fraud detection algorithms are frequently discussed in the literature, only few research studies acknowledge the importance of network analytics in fraud detection. To the best of our knowledge, Table 3.1 gives an overview of all published papers related to fraud detection using network analytics. The table evaluates each paper according to the identified challenges in Section 3.2.2. All papers comply with *Challenge IV*, i.e., including network analysis in the detection process.

Methods 1-5 focus on one type of network feature to measure or visualize fraud and rely to a larger extent on human interaction for effectively guiding the fraud detection process. GOTCHA! is designed

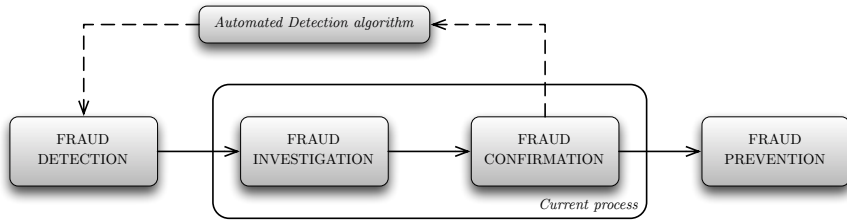


Figure 3.6: Fraud detection process for the social security institution.

such that it derives multiple network-based features in order to judge the fraudulence of other instances. Methods 6-10 are more advanced; they analyze and combine multiple aspects of the direct neighborhood to decide whether a node in the network is fraudulent or not. Collective inference procedures for fraud detection are discussed in methods 11-17. Rather than only taking into account the direct neighborhood, GOTCHA! implicitly uses the indirect neighborhood to infer a label for the unknown nodes, both anticipating future fraud and forgiving past associations.

Except for Šubelj et al. (2011) and Chau et al. (2006), all fraud detection papers exclusively use network variables to detect fraud, neglecting instance-specific information. Although we believe that the network effects play an important role in accurately identifying fraud, individual instance behavioral information often also contains subtle signs of new types of fraud and should therefore not be disregarded and considered as a valuable indicator in the fraud detection process. Our study differs from the work of Šubelj et al. (2011) and Chau et al. (2006) as they use intrinsic features only to bootstrap the network learning algorithms. In order to develop a comprehensible and usable technique for experts, we extend the intrinsic features with domain-driven network features. As such, we offer experts the opportunity to gain insights about the importance of each of the variables in the fraud detection process. Given that current research does not offer an encompassing approach, we developed GOTCHA!.

3.2.4 Proposed Fraud Detection Process

In order to make the GOTCHA! approach useable, it needs to be embedded in the global context of the fraud detection process. The goal of social security fraud detection is to define which companies are likely to commit fraud within a certain period of time. Currently, social security experts have mainly focused on manually inspecting random companies and determining whether they are involved in fraud or not. This section discusses how we propose to extend the current process. The fraud detection process is illustrated in Figure 3.6.

Fraud detection is the automated process of identifying high-risk instances. For reasons of generality, we use the term *Automated Detection Algorithms* to refer to any technique that is able to estimate a fraud detection model, such as tree models, linear or logistic functions, SVMs, ANNs, Bayesian learning, ensemble models, etc. (Hastie et al., 2001; Carrizosa et al., 2014). During *fraud investigation*, experts decide to agree or disagree with the high-risk companies identified by the model using their practical insights and knowledge. Note that, currently, experts are not guided as to which companies are potentially high-risk. This makes the fraud investigation process inefficient and time-consuming. The high-risk companies are passed on to the field auditors who finally confirm if their expectations are correct (*fraud confirmation*).

Observe the interactive nature of such a system: while experts feed the fraud detection algorithms with confirmed fraud, our algorithm guides the experts in turn where to look for fraud. In the end, the ultimate goal is to evolve towards *fraud prevention*, i.e., the ability of detecting fraud before it is even committed (Bolton and Hand, 2002).

This chapter studies the fraud detection phase by proposing GOTCHA!. The next section will discuss the fraud detection process in more detail. We expect that our process is more efficient and systematic than experts merely following their own intuition. Our estimated models give a good indicator which companies are likely to commit fraud (see Section 3.5).

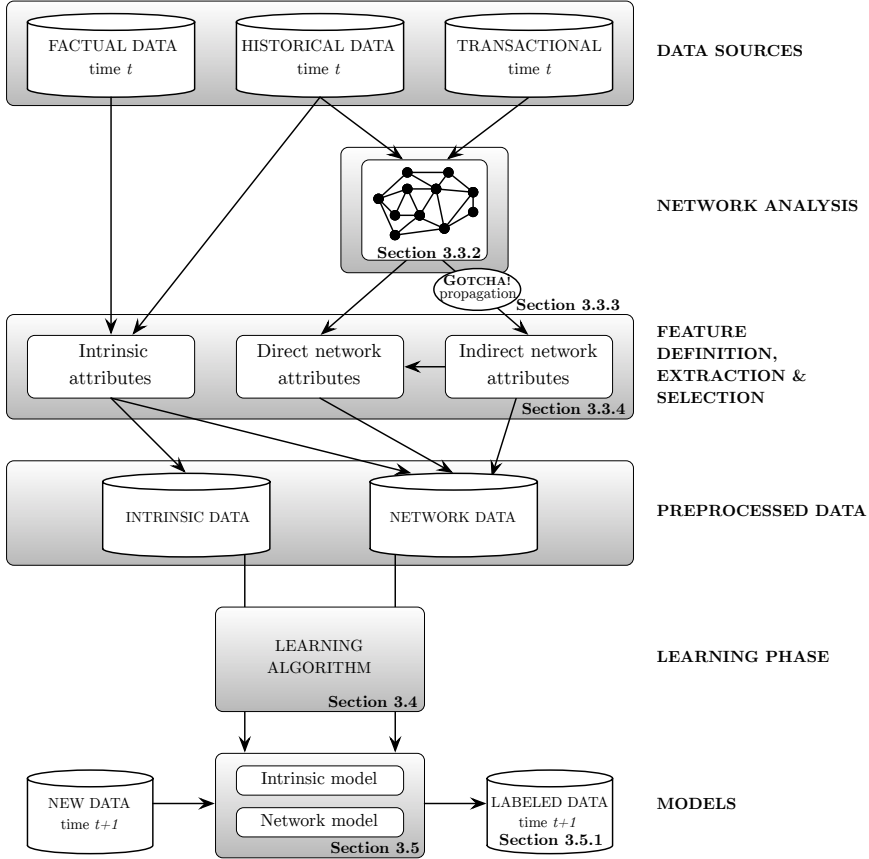


Figure 3.7: Proposed GOTCHA! framework for social security fraud detection.

3.2.5 GOTCHA!'s Fraud Detection Framework

Figure 3.7 illustrates in greater detail our proposed framework for the fraud detection phase (see Figure 3.6) in a social security context. We start from three data sources. A factual data source contains company-specific information such as regional, sectorial and legal characteristics of each company. Historical data log changes in company information, e.g., when a company changes its legal seat. Transactional data record which resources are associated to which companies, including the time period. Those data sources

COMPANY	Intrinsic Variables								Network Variables				Fraud?	
	Regional		Sectorial		Historical		Legal		Direct		Indirect			
	province	...	sector	...	age	...	form	...	degree	quadrangles	...	exposure		...
6357	P4	...	catering	...	3	...	Corp.	...	16	0	...	0.12	...	<i>No</i>
3904	P7	...	transport	...	1.5	...	PLLC	...	8	0	...	0.01	...	<i>No</i>
3041	P5	...	cleaning	...	0.7	...	LLC	...	56	8	...	0.65	...	<i>Yes</i>
7932	P2	...	catering	...	8	...	Corp.	...	93	7	...	0.03	...	<i>No</i>
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Figure 3.8: Example of a preprocessed data set.

are transformed into relevant company-specific and network-centric attributes. Transactional data form the basis to create the global network structure representing the relationships between companies and resources as a bipartite graph (Section 3.3.2). As historical relationships between companies and resources contain important information, we use the historical data sources to reconstruct historical links and add them to the network, weighing the links based on their recency. While the past and the present is explicitly implemented in such a network, future behavior can be estimated by exploiting both direct effects as well as collectively inferring fraud through the whole network (Section 3.3.3). Approximately 350K active and non-active companies and 5 million resources are considered in the network.

According to Verbeke (2012), variables can be classified into two categories:

Definition 3.2. A **local** or **intrinsic variable** represents intrinsic information of a company as if it was treated in isolation. Those variables include regional, sectorial, historical and legal characteristics.

Definition 3.3. A **network variable** aggregates information that is contained by the neighborhood of each company. We assume that behavior of a company’s neighbors has an influence on the company itself. Those variables include the degree, triangles and propagated exposure score (see Section 3.3.4 for details), and can be classified as direct and indirect network variables depending on whether they are

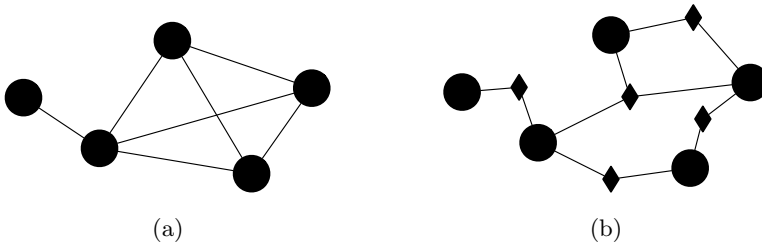


Figure 3.9: Overview of a unipartite (a) and a bipartite (b) graph.

derived from the direct neighborhood or take into account the full network structure.

Figure 3.8 gives an example of the preprocessed data, and features of each category. We derive regional, sectorial and legal variables from the factual data source; the historical features are extracted from the historical data. The transactional data source is the basis for the creation of the network variables and specifies which resources are assigned to which companies for which time period (see Section 3.3).

In the remainder of this chapter, we will use the terms intrinsic and network variables to indicate whether the variables are generated by instance-specific or network-centric information. The data preprocessing phase derives intrinsic, direct and indirect network attributes. Rather than using plain relational classifiers as proposed by (Macskassy and Provost, 2007) to predict fraud, the network data set imposes a mix of intrinsic and domain-driven network attributes. A learning algorithm will then estimate the corresponding models (Section 3.4). Those models are used to evaluate fraudulent behavior of companies (Section 3.5).

3.3 Network Analytics for Fraud Detection

3.3.1 General Concepts and Notations

Our proposed approach is based on fundamentals from graph theory, incorporating *Challenge IV* of Section 3.2.2. Boccaletti et al. (2006)

define graph theory as the natural framework for the exact mathematical treatment of complex networks. Formally, a complex network can be represented as a graph. A graph consists of a set of *vertices* $v \in \mathcal{V}$ and *edges* $e \in \mathcal{E}$. Vertices – also referred to as nodes or points – are connected by edges – also known as links or lines. A standard graph can thus mathematically be represented as $\mathcal{G}(\mathcal{V}, \mathcal{E})$, and is shown in Figure 3.9a. A graph can be either *directed* or *undirected*, depending on the direction imposed on the edges. When edges define the capacity or the intensity of a connection (Boccaletti et al., 2006), the graph is said to be *weighted*. Mathematically, a graph is represented as a matrix. The adjacency matrix $\mathbf{A}_{n \times n} = (a_{i,j})$ is the corresponding matrix representation of size $n \times n$ of a graph, with n being the total number of vertices and $a_{i,j} = 1$ if a link between node i and j exists, and $a_{i,j} = 0$ otherwise. The weight matrix $\mathbf{W}_{n \times n} = (w_{i,j})$ captures the link weight of the relationships between the nodes.

Most networks contain only one node type. Certain applications, however, require implementing a second entity. Such networks are bipartite graphs, as shown in Figure 3.9b. In contrast to unipartite graphs, a bipartite graph $\mathcal{G}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E})$ consists of two types of vertices $v_1 \in \mathcal{V}_1$ and $v_2 \in \mathcal{V}_2$. An edge $e \in \mathcal{E}$ exclusively connects objects from different classes to each other. For each edge in a bipartite graph, the following property holds:

$$e(v_1, v_2) \in \mathcal{E} | v_1 \in \mathcal{V}_1 \text{ and } v_2 \in \mathcal{V}_2 \quad (3.1)$$

This property enforces that two instances of the same class are never directly connected, but always connect through an object of the other class. The adjacency matrix of an undirected bipartite graph is formally written as $\mathbf{A}_{n \times m} = (a_{i,j})$, with $a_{i,j} = 1$ if a link between node $i \in \mathcal{V}_1$ and node $j \in \mathcal{V}_2$ exists, and $a_{i,j} = 0$ otherwise. The corresponding adjacency matrix has a size of $n \times m$, with n and m the number of objects in set \mathcal{V}_1 and \mathcal{V}_2 respectively. The weight matrix is $\mathbf{W}_{n \times m} = (w_{i,j})$.

3.3.2 Time-weighted Bipartite Networks

Reality is often difficult to capture in mathematical formulations or even a graphical representation. Network analysts, in consideration

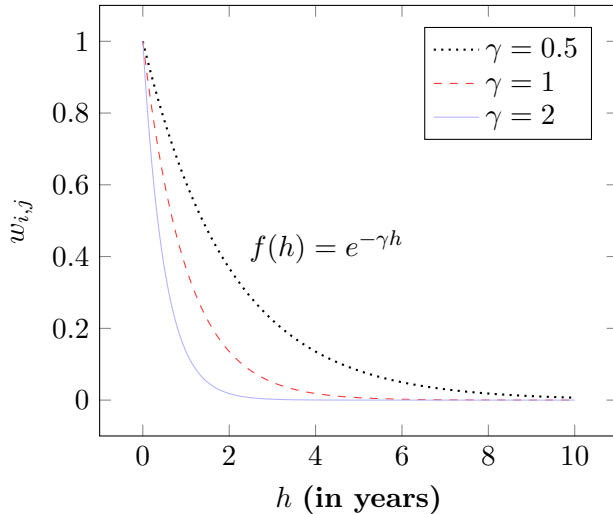


Figure 3.10: Exponentially weighting the recency of the relationships between companies and resources to determine the tie strength using different values of γ .

with field experts, should carefully choose and agree upon the right design of the network, reflecting the reality in the best possible way. It is particularly important to bridge the richness of experts' knowledge to the technical limitations of network analytics by selecting the most relevant data features for the analysis.

We argued in Section 3.1 that in a social security fraud detection problem companies are related to their resources. The goal of fraud detection is to find high-risk companies, but the resources are an important indicator as they help in executing the company's (fraudulent) activities. Resources are transferred from company to company. If a currently legitimate company inherits resources from a fraudulent company, this substantially increases the fraud risk of that company. Hence, we create a bipartite graph (or bigraph) connecting companies to their past and present resources. We work with undirected networks as fraud can pass from a company to a resource, and vice versa.

For computational reasons, the graphical representation is mapped into a weight matrix \mathbf{W} with size $c \times r$, where c and r specify the num-

ber of companies and resources respectively. The strength of the relationship between a company and resource is exponentially weighted in time:

$$\begin{cases} w_{i,j} = e^{-\gamma h} & \text{if a relationship exists between company } i \text{ and resource } j \\ w_{i,j} = 0 & \text{otherwise} \end{cases}$$

with γ the decay constant⁶, and h the time passed since the resource was linked to the company, with $h = 0$ representing a current relationship. The value of the decay constant γ indicates the rate at which past information declines, and is chosen (by mutual agreement with the experts) such that only limited past information is taken into account. Particularly, if experts say that the associations can be considered as irrelevant after x days, then we choose γ such that the decay function goes to zero for time values greater than x , i.e., $f(t > x) \approx 0$. For example, if one decides that information of only 5 years back should be taken into account, then $\gamma \approx 1$. This is depicted in Figure 3.10.

The matrix \mathbf{W} is time-dependent. To incorporate the time-evolving characteristics of fraud (cfr. *Challenge III* in Section 3.2.2), we create a matrix \mathbf{W}_t for each timestamp $t \in \{t_0, t_1, t_2, t_3\}$, representing the interrelated structure at time t . The social security bi-graph contains approximately 350K active and non-active companies and 5 million active and non-active resources in every timestamp of analysis. In each timestamp, the network density is around 4.5×10^{-6} .

3.3.3 GOTCHA!’s Fraud Propagation Algorithm: Defining high-risk nodes in the network

This section handles *Challenge V* (see Section 3.2.2). In particular, we answer the following questions: (1) Which *resources* are often involved in fraud and exhibit a high risk to entice other companies to perpetrate fraud as well? (2) Which *companies* are sensitive to fraud? More specifically, we need a score that indicates which *resources* are coincidentally associated with fraudulent companies (low-risk) and which resources systematically pop up when fraud is detected (high-risk). For example, assume an address that was previously used by a

⁶Due to confidentiality issues, we will not elaborate on the exact value of γ .

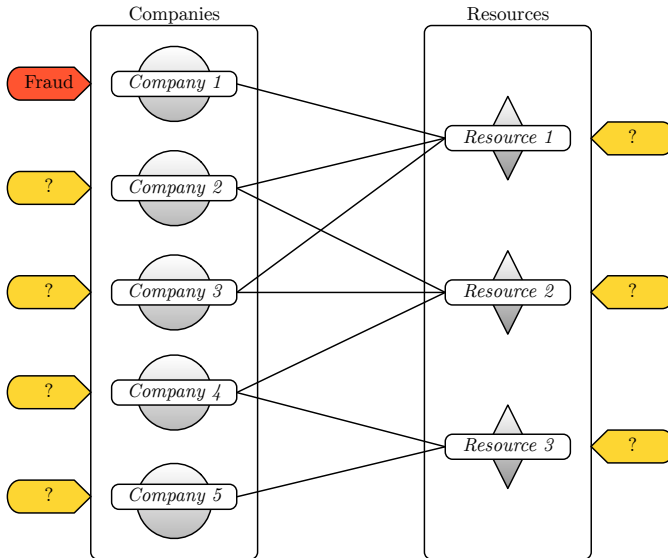


Figure 3.11: Overview of propagation task. Only a limited number of companies is labeled. Using a propagation algorithm we infer an exposure score for each company and resource in the network, representing the extent to which a company/resource is exposed to fraud.

fraudulent company is taken over by another company. What would you say about the riskiness of that resource? Would the resource riskiness change if you knew that the address was already used by many fraudulent companies previously, or if the address was the location of only one fraudulent company many years ago? Similarly, we derive a score that gives a primary indication of how the *company* is affected by the fraudulent influences from its neighborhood. Figure 3.11 gives an overview of the task at hand.

Given a time-weighted bipartite graph of companies and resources, we want to diffuse or *propagate* the effect of a limited number of known fraudulent companies through the network (see Figure 3.12) and infer an *exposure score* for every node (i.e., resource and company) in the network. The exposure score expresses the extent to which the node is affected by fraud. As only companies are directly attributed to fraud, we start from the label of the few confirmed

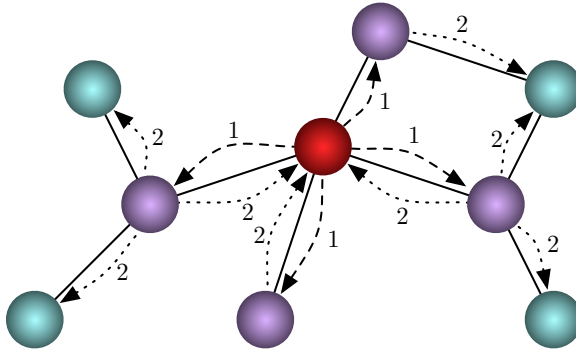


Figure 3.12: Illustration of GOTCHA's propagation algorithm. The dark node in the center propagates its fraudulent influence to its neighbors (step 1) The neighbors absorb the influence and propagate on their turn their fraudulent influence to their neighbors (step 1 + 2). The iterations are repeated several times until convergence.

fraudulent companies. The bipartite graph allow to spread fraudulent influence through the network and define an exposure score for each company and resource. As such, each company can be analyzed based on its own exposure score and the links to high- and low-risk resources.

We start from the Personalized PageRank algorithm (Page et al., 1998), one of the popular applications of the *Random Walk with Restarts* (RWR) method (Gleich, 2014; Gyöngyi et al., 2004), and extend it so that the following domain-specific requirements are fulfilled:

1. *Bipartite graphs*: fraud contaminates both companies and resources.
2. *Focus on fraud*: only fraud – and no legitimate effects – propagates through the network.
3. *Dynamics*: fraud is evaluated upon its recency.
4. *Degree-independent propagation*: high-degree companies spread proportionally more fraud than low-degree companies.

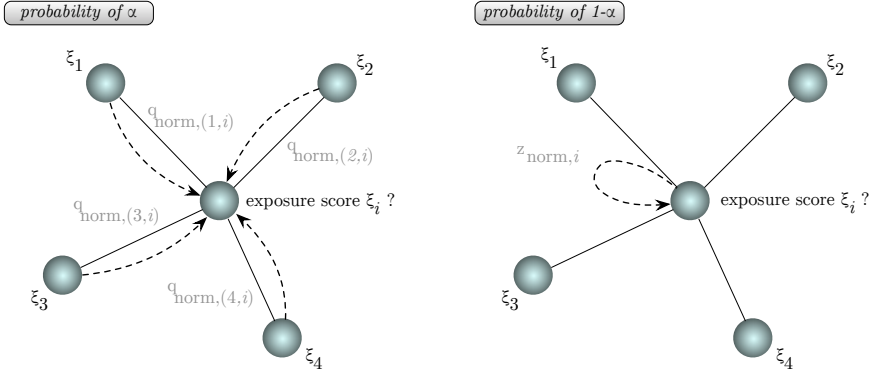


Figure 3.13: The exposure score for each node depends on (a) the exposure scores of the node’s neighborhood (left figure) and (b) a random jump towards another node in the network (right figure).

In general, the Personalized PageRank algorithm (see Figure 3.12 for an overview) computes an exposure score for each node which depends on (a) the exposure scores of the node’s neighborhood and (b) a random jump towards another node in the network. This is depicted in Figure 3.13. Mathematically, this can be written as,

$$(\vec{\xi}) = \alpha \cdot \mathbf{A}(\vec{\xi}) + (1 - \alpha) \cdot \vec{v} \quad (3.2)$$

with $(\vec{\xi})$ a vector containing the exposure scores of the nodes, \mathbf{A} the corresponding column-normalized adjacency matrix, $(1 - \alpha)$ the restart probability and \vec{v} the restart vector. The restart vector \vec{v} is uniformly distributed over all nodes, and normalized afterwards.

Solving Equation 3.2 requires a matrix inversion. This is often not feasible to compute in practice. The most widely used way to compute the relevance score is by the power iteration method, which iterates until convergence (Tong et al., 2006). Convergence is reached until the change is marginal, or after a maximum number of iteration steps. Next, we discuss how we integrate the fraud-specific domain requirements into the algorithm.

Requirement 1 Equation 3.2 is developed for unipartite graphs. We want to assess the extent to which fraud affects both companies and resources. Starting from the weighted adjacency matrix $\mathbf{W}_{c \times r}$

of the bipartite graph with c companies and r resources (see Section 3.3.2), the matrix is transformed to a unipartite representation, according to (Tong et al., 2008),

$$\mathbf{Q} = \begin{pmatrix} 0_{c \times c} & \mathbf{W} \\ \mathbf{W}' & 0_{r \times r} \end{pmatrix} \quad (3.3)$$

Matrix \mathbf{Q} is a symmetric matrix with $c + r$ rows and columns. Introducing zeros enforces that resources exclusively connect to companies and vice versa. The column-normalized matrix is \mathbf{Q}_{norm} , a matrix where all columns sum to 1. The iterative propagation procedure for bipartite graphs can then be written as,

$$(\vec{\xi}) = \alpha \cdot \mathbf{Q}_{norm}(\vec{\xi}) + (1 - \alpha) \cdot \vec{v} \quad (3.4)$$

Note that \mathbf{Q}_{norm} is a dynamic matrix, representing both present and past relationships. All active and non-active companies are included. This allows us to integrate and exploit all connections (ever established) among companies and resources. The vectors $\vec{\xi}$ and \vec{v} are of size $c + r$, containing the exposure scores and restart probabilities of the companies and the resources.

Requirement 2 The goal is to focus on fraud and exclusively propagate fraudulent influence through the network. A similar approach is taken in Provost et al. (2009) to compute brand affinity, measuring the proximity of a node to the seed nodes. Seed nodes are nodes that already are enticed about the product or, in our case, into fraud. Given information provided by seed nodes, how will this affect the other currently legitimate companies and resources in the network? We use the restart vector to personalize the ranking towards fraud and stress the fraudulent influences of the seed nodes. The restart vector specifies which nodes (here: companies) committed fraud, where $v_j = 1$ if entry j is a fraudulent company and $v_j = 0$ if entry j is a resource or a legitimate company. Although there is a lack of evidence of confirmed fraud nodes, the algorithm is able to cope with only few labeled nodes by emphasizing fraud in the restart vector.

Requirement 3 Fraud is dynamic. Recently caught companies are a more important source of spreading fraud than companies de-

tected many years ago. The restart vector reflects the fraudulent influence a certain company can disperse, and should depend on the recency of the fraud. The more time passed since fraud was detected, the lower a particular fraudulent company's influence. Inspired by the half-time decay of nuclear particles, we exponentially decay the relevance of fraudulent activities over time,

$$\begin{cases} v_j = e^{-\beta h} & \text{if entry } j \text{ is a fraudulent company} \\ v_j = 0 & \text{otherwise} \end{cases}$$

with β the decay constant (see Section 3.3.2 for details), and h the time passed since the company was detected fraudulent where $h = 0$ represents a current fraud company.

Requirement 4 Fraudulent companies infect their surrounding resources directly. However, low-degree companies have fewer links through which fraud can propagate and affect the resources more strongly. High-degree companies have many links, resulting in a marginal impact on the neighboring nodes. In realistic situations, this assumption does not hold. The influence of high-degree companies should be equally treated as low-degree companies, as high-degree companies have a wider range to influence other companies. Hence, fraud propagation has to be proportional to a node's degree, and

$$\vec{z} = \vec{v} \odot \vec{d} \quad (3.5)$$

with \vec{z} the degree-adapted restart vector, which is the element-wise product of the restart vector \vec{v} and the degree vector \vec{d} denoting the degree of each entry. The normalized vector is \vec{z}_{norm} .

After $k + 1$ iterations, the exposure score for each company and resource equals

$$\vec{\xi}_{k+1} = \alpha \cdot \mathbf{Q}_{norm} \cdot \vec{\xi}_k + (1 - \alpha) \cdot \vec{z}_{norm} \quad (3.6)$$

with $(1 - \alpha)$ the restart probability⁷, \mathbf{Q}_{norm} the column-normalized adjacency matrix, \vec{z}_{norm} the normalized degree-adapted restart vector, $\vec{\xi}_k$ a vector containing the exposure scores of all nodes

⁷based on Page et al. (1998), we choose $\alpha = 0.85$.

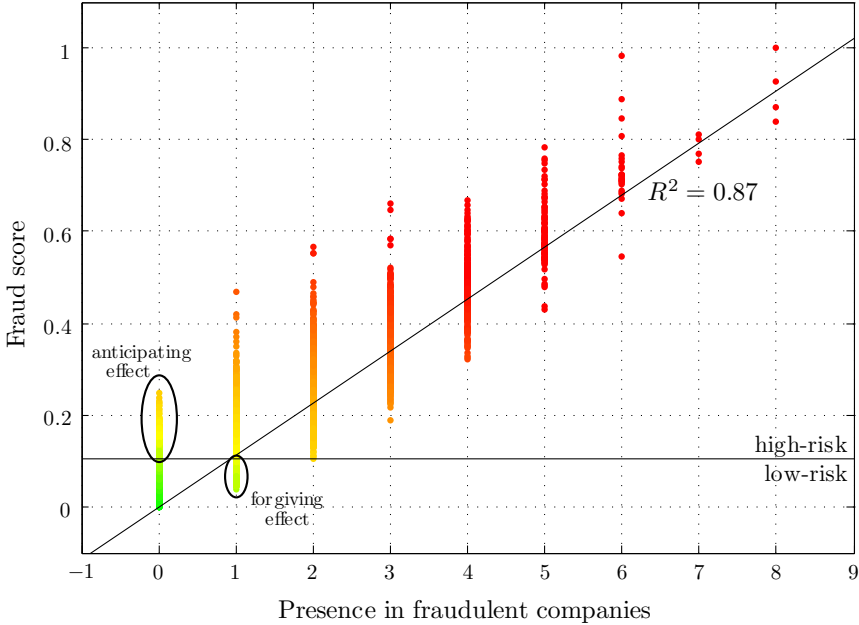


Figure 3.14: Each resource is associated with its propagated exposure score and its presence in fraudulent companies. The resources are colored according to their riskiness (red indicates high risk, green is low risk). The horizontal line represents the boundary dividing the resources in a low-risk and high-risk category. Note that only 0.28% of all resources are labeled as high-risk.

after k iterations, and $\vec{\xi}_0$ the initial distribution. Note that the final scores are independent of the initial values of $\vec{\xi}_0$ (Page, 2001). We repeat the process for 100 iterations in order to make sure that potential changes in the final exposure score are only marginal.

Apart from a company score, the GOTCHA! propagation algorithm also assigns an exposure score to each resource. Note that the interpretation of the exposure scores of both companies and resources is the same: it expresses the extent to which the company/resource is exposed to fraud. Figure 3.14 shows the exposure scores of the resources compared to their presence in fraudulent companies (for year t_0). In general, 87% of the variation in the resources' exposure score is

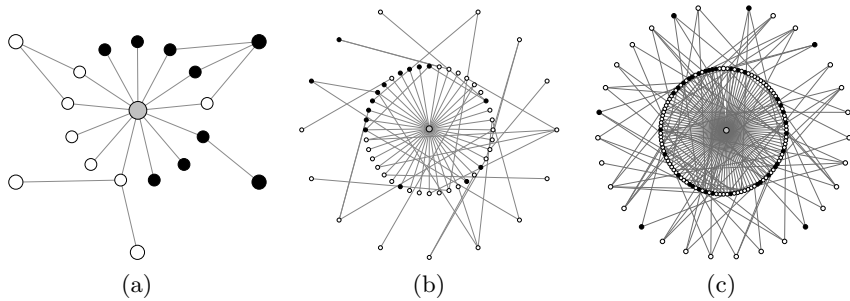


Figure 3.15: Various egonets for micro- (a), small- (b) and medium-sized (c) companies. The company is the center (i.e., the ego) of the egonet and is surrounded by its resources (i.e., the alters). High-risk resources are labeled in black, low-risk nodes are white-colored. All central companies (egos) are still active at the time of analysis.

explained by their presence in fraudulent companies. While certain resources were never associated with fraudulent companies before, they receive a relatively high exposure score. This means that, although those resources are not directly contaminated by fraudulent activities, they are surrounded by a huge amount of fraud. We call this the *anticipating effect* of GOTCHA!’s fraud propagation. On the other hand, some resources have been involved in fraudulent companies, but received a low fraud score. Due to the incorporation of the recency of fraud in the propagation algorithm, there is a *forgiving effect* present. When time evolves and resources were not involved in fraud again, their fraudulent influence decreases and is only marginal.

In agreement with social security fraud experts, GOTCHA! considers resources involved in at least two fraudulent companies always as high-risk. The minimum exposure score of the resource connected to at least two fraudulent companies is chosen as the cut-off value to distinguish between low- and high-risk resources. The horizontal line in Figure 3.14 illustrates this cut-off value. Resources located above the cut-off line are marked as high-risk. Note that this corresponds to only 0.28% of all resources.

Having an estimated probability of the riskiness of the resources, we are now able to characterize each company based on its connectivity to high- and low-risk resources.

Feature	Description	(u)	(b)	G
DIRECT FEATURES				
<i>Neighborhood Degree</i>	number of first-order neighbors that are of			
<i>high-risk</i>	- high-risk	X	X	X
<i>low-risk</i>	- low-risk	X	X	X
<i>relative</i>	proportion of high-risk neighbors	X	X	X
<i>Time-Weighted Degree</i>	time-weighted ⁸ number of first-order neighbors that are of			
<i>high-risk</i>	- high-risk			X
<i>low-risk</i>	- low-risk			X
<i>relative</i>	proportion of high-risk nodes, weighted in time			X
<i>Triangles</i>	number of closed triples in the neighborhood that contain			
<i>high-risk</i>	- at least one high-risk node	X		
<i>low-risk</i>	- no high-risk nodes	X		
<i>relative</i>	proportion of triples that contain at least one high-risk node	X		
<i>Quadrangles</i>	number of quadrangles in the extended neighborhood that contain			
<i>high-risk</i>	- at least one high-risk company node		X	X
<i>time-weighted</i>	- at least one high-risk company node, weighted in time			X
<i>low-risk</i>	- no high-risk company nodes		X	X
<i>time-weighted</i>	- no high-risk company nodes, weighted in time			X
<i>relative</i>	proportion of quadrangles that contain at least one high-risk company node		X	X
<i>time-weighted</i>	- weighted in time			X
<i>Quadrangle Frequency</i>	quadrangles in the extended neighborhood that contain the same two first-order neighbors, and have			
<i>mean (high-risk)</i>	- at least one high-risk company node, averaged		X	X
<i>time-weighted</i>	- at least one high-risk company node, averaged and weighted in time			X
<i>max (high-risk)</i>	- at least one high-risk company node, maximum		X	X
<i>time-weighted</i>	- at least one high-risk company node, maximum and weighted in time			X
<i>mean (low-risk)</i>	- no high-risk company nodes, averaged		X	X
<i>time-weighted</i>	- no high-risk company nodes, averaged and weighted in time			X
<i>max (low-risk)</i>	- no high-risk company nodes, maximum		X	X
<i>time-weighted</i>	- no high-risk company nodes, averaged and weighted in time			X
<i>Neighborhood Similarity</i>	count of similar neighbors	X	X	X
INDIRECT FEATURES				
<i>Exposure Score</i>	node's own exposure score	X	X	X
<i>Neighborhood Exposure</i>	first-order neighbors' exposure score			
<i>mean</i>	- averaged	X	X	X
<i>weighted mean</i>	- time-weighted			X
<i>maximum</i>	- maximum	X	X	X

Table 3.2: Network-based feature extraction for (u) Unipartite, (b) Bipartite, (G) GOTCHA!.

3.3.4 Network Feature Extraction

Given all legitimate companies at time t , we want to rank those companies according to their *fraud risk* – i.e., the probability that they will commit fraud in the near future. As this risk depends on a combination of intrinsic and network-based variables, we need to transform network information to a set of promising network-based features for each active company (Eliassi-Rad and Henderson, 2011). We infer two types of network-based features: direct and indirect features. The direct network features are derived from each company’s direct neighborhood. Given the bipartite structure of our network, for each company we take into account all nodes that are one and two hops removed from the center (i.e., a company’s associated resources and companies). Figure 3.15 illustrates the direct neighborhood of a company with varying neighborhood size. The indirect network features are derived from the exposure scores which use the whole network rather than a node’s neighborhood. Table 3.2 gives an overview of the features derived from the network.

Our approach GOTCHA! is evaluated against three baselines: (1) a model without network features, (2) a model with unipartite features, and (3) a model with bipartite features not time-weighted. In (2), companies are directly linked to each other. The link weight expresses the number of shared resources between both companies. Here, the direct features are derived from the first-order neighborhood as this explicitly comprises the associated companies. In (3), the network has a bipartite structure, but the links are not weighted in time. For each company, the unipartite model (2) extracts the following direct features: degree, triangles, neighborhood similarity. The degree counts the number of neighbors. Since the impact of high-risk neighbors is an important indicator of fraud, we distinguish between the number of first-order high-risk and low-risk neighbors, and the ratio hereof. Remark that a node is classified as high-risk if the node is a fraudulent company or if the node has a sufficient large exposure score as explained in Section 3.3.3. A triangle is defined as three nodes that are all connected to each other. We say that a triangle has a high-risk

⁸Time is included in the edge weight.

if at least one of the associated nodes is classified as high-risk. Neighborhood similarity measures the extent to which the characteristics of the neighbors are similar to the node of interest. Here, we compare companies based on location and sector-specific information, guided by expert expectations.

The indirect features include the company’s own exposure score and the exposure scores of the first-order neighborhood aggregated by the mean and maximum. The exposure score is computed according to Equation 3.2 where the restart vector incorporates fraud (Requirement 2). The bipartite model (3) derives the same set of features as the unipartite model, with the exception of triangles. In our bipartite network structure where companies (resources) are exclusively connected to resources (companies), no triangles exist. However, a shift of many resources from one company to another might indicate the existence of a spider construction. Hence, we count the number of quadrangles – i.e., a closed path of four nodes – in the extended neighborhood where we both include the first- and second-order neighborhood. We say that a quadrangle is of high risk if at least one high-risk company node is associated with the quadrangle. The quadrangle frequency establishes how many quadrangles are formed for each pair of resources. As the length of this feature value differs from company to company, we compute the mean and average amongst all pairs of resources. The features in GOTCHA! differ from those of (3) as they are time-weighted by the edges. For high-risk degree for example, this means that we sum the edge weight of the associated high-risk resources. The value of a weighted quadrangle is determined by the arithmetic mean of the link weights (Opsahl and Panzarasa, 2009). We also derive the weighted mean of the first-order neighbors’ exposure scores, weighing the impact of each node’s exposure score by the edge weight.

We construct the features for each timestamp $t \in \{t_0, t_1, t_2, t_3\}$ and hence take into account the time-evolving property of fraud. Together with the intrinsic features, these network-based features are fed into a learning algorithm. An overview of the features’ summary statistics for year t_3 can be found in Table 3.3.

Feature	Summary Statistics			
	Fraud		Non-Fraud	
	μ	σ	μ	σ
DIRECT FEATURES				
<i>Neighborhood Degree</i>				
<i>high-risk</i>	31.37	39.09	13.13	48.12
<i>low-risk</i>	2.00	8.43	5.55	18.44
<i>relative</i>	0.91	0.23	0.55	0.39
<i>Time-Weighted Degree</i>				
<i>high-risk</i>	19.15	22.01	6.70	17.86
<i>low-risk</i>	0.56	4.10	2.64	7.20
<i>relative</i>	0.93	0.23	0.56	0.45
<i>Quadrangles</i>				
<i>high-risk</i>	56.71	198.83	0.62	37.04
<i>time-weighted</i>	45.34	155.41	0.17	13.16
<i>low-risk</i>	131.30	294.40	375	108729.60
<i>time-weighted</i>	55.88	162.35	134	36197.67
<i>relative</i>	0.19	0.32	0.0040	0.0048
<i>time-weighted</i>	0.23	0.36	0.0036	0.0046
<i>Quadrangle Frequency</i>				
<i>mean (high-risk)</i>	0.64	0.87	0.03	0.23
<i>time-weighted</i>	0.28	0.40	0.0075	0.0641
<i>max (high-risk)</i>	1.52	2.46	0.0398	0.3328
<i>time-weighted</i>	0.62	0.98	0.0104	0.0956
<i>mean (low-risk)</i>	0.89	0.57	0.45	0.56
<i>time-weighted</i>	0.47	0.38	0.19	0.27
<i>max (low-risk)</i>	1.75	1.59	0.64	1.22
<i>time-weighted</i>	0.95	0.85	0.31	0.50
<i>Neighborhood Similarity</i>				
<i>Sector</i>	0.60	0.49	0.73	0.45
<i>Location I</i>	0.55	0.50	0.45	0.50
<i>Location II</i>	0.01	0.11	0.01	0.12
INDIRECT FEATURES				
<i>Exposure Score</i>	0.0027	0.0046	3.565e-5	2.569e-4
<i>Neighborhood Exposure</i>				
<i>mean</i>	0.0106	0.01756	3.381e-4	2.024e-3
<i>weighted mean</i>	8.49e-3	0.0156	1.93e-4	1.668e-3
<i>maximum</i>	0.0353	0.0467	0.0028	0.0119

Table 3.3: Network-based feature extraction.

3.4 Modeling Approach

The social security institution keeps track of fraudulent companies and labels them fraudulent as soon as suspicious activities are discovered. Having an extensive database containing time-related records, we are able to evaluate time-consistent models at different timestamps and time windows. In our analysis, we define four timestamps $t \in \{t_0, t_1, t_2, t_3\}$. For each timestamp, we specify within which time window the learning algorithm has to predict whether a company will be fraudulent or not. We evaluate the models on their detection of short-, medium- and long-term frauds. For instance, a short-term model estimates the probability of short-term fraud. The time windows are set to 6, 12 and 24 months, by experts' agreement.

A key challenge in predicting social security fraud is making the right trade-off between a small time window that accurately reflects current types of fraud, and a larger time window which provides more confirmed evidence of fraud and anticipates new fraudulent structures.

As mentioned, in order to evaluate the relevance of relational information in fraud prediction, we compare the GOTCHA! model with three baselines. The same instances are used in the training and test sets for the baselines and GOTCHA! network model. By doing so, we are able to determine the added value of incorporating relational knowledge (in terms of network-based features) on the performance of the prediction models. We discuss each of the models in more detail below.

Baseline - Intrinsic – is trained and tested with intrinsic-only variables. Relationships with other companies and resources are neglected in the analysis.

Baseline - Unipartite – integrates intrinsic and network-based variables into one model (see *Challenge II* in Section 3.2.2). The network only consists of companies that are linked to each other by means of resources. Link weight is defined as the number of resources that both companies share.

Baseline - Bipartite – integrates intrinsic and network-based variables into one model (see *Challenge II* in Section 3.2.2). The network includes both companies and resources. A binary link weight is imposed, defining whether a link exists between a company and a resource.

Proposed GOTCHA! model – enriches the bipartite model with time-weighted network features.

Both the intrinsic and relational features can be seen as different views from multiple data sources that describe the same problem. This is also referred to as multi-view learning (Xu et al., 2013). In the context of social security fraud, we collect data from three data sources: the factual and historical data sources that register a company’s declarations and reports; and the transactional data source which stores data from a real-time tool provided by the government where a company should report which resources it is currently using (see Figure 3.7). These three data sources are combined into two views: one that describes the company’s own characteristics, and one that specifies the company’s connections with other companies. The traditional approach to deal with multiple views of the same data, is to concatenate the feature vector of each view into one single feature vector. In this chapter, we focus on the concatenation of both intrinsic and relational features, and then apply single-view learning. Future research (see Chapter 7) should further elaborate on how to fully exploit the two views rather than to combine them into one single view. Co-training, for example, is a multi-view learning approach where a model is alternately learned for each view, where the results of one view contribute to the model development of the other view.

3.4.1 Rebalancing the data set

To address the extremely skewed data distribution (see *Challenge I* in Section 3.2.2), we use the *SMOTE* approach (Chawla et al., 2011) to rebalance the data set. *Synthetic Minority Oversampling Technique (SMOTE)* is a combination of oversampling the minority class and undersampling the majority class (Chawla et al., 2011). Based on the

experimental results of Chawla et al. (2011), we choose an oversampling and undersampling percentage of 400% and 200% respectively.

3.4.2 Learning algorithm

Random Logistic Forests and *Random Forests* are implemented to train the models. We opt for ensemble methods as individual logistic regression or decision trees often fail to appropriately weigh features based on their predictiveness (Gallagher et al., 2008), which our data set confirmed (see Section 3.5). Breiman (2001) proposed Random Forests, an ensemble of trees. Random Logistic Forests, as proposed by (Gallagher et al., 2008), is an ensemble of plain vanilla logistic regressions, where each classifier is fed with $\lfloor \log(N) + 1 \rfloor$ random features, with N the total number of features. The final label assigned to an instance is based on the majority vote of each individual model. We estimate an ensemble of 500 individual models, each with 6 random features.

Using ten-fold stratified cross-validation, we enforce the learning algorithms to use each instance once in the test set. Stratified sampling ensures that each sample represents the real fraud distribution. As such, we can average the results, obtaining more stable performance measures of each of the models and resulting in a better impression of the significance of the different types of variables.

In summary, our experiments are designed to answer the following questions: 1) Do network-based variables yield better performance over intrinsic-only variables? If so, by how much? 2) Is the incorporation of a bipartite, time-weighted network structure essential? 3) Are network models able to capture changes in the environment? Do they statistically perform better as the baselines over the different timestamps? 4) Are the network models able to identify companies that will perpetrate fraud in the near future and also on long term?

3.5 Results

In this section, we discuss the results of our GOTCHA! network model compared to the baselines. All models are evaluated in terms of

the AUC score (Area Under ROC Curve), precision and recall. We use an extensive time-dependent data set obtained from the Belgian Social Security Institution. For each timestamp, approximately 220,000 active companies and more than 5 million resources are registered. Our goal is to find companies that exhibit a high risk of perpetrating fraud. We extract intrinsic features that describe the current characteristics of a company, and network-based features that take into account the present and past relationships to the resources. We train and test models based on fraudulent companies found and confirmed by experts. We analyze the difference in performance between the baselines and the GOTCHA! model, as well as the difference in performance for the various time windows (i.e., short, medium and long term).

Do network-based features boost the performance of traditional models that only use intrinsic features? That is, does the GOTCHA! model significantly outperform the baselines? As opposed to existing methods (Chau et al., 2006; Šubelj et al., 2011) which bootstrap the network propagation algorithm with the output of an intrinsic model, we opt to include both intrinsic and domain-driven network-based features in the final model. There are two reasons. First, our approach indicates which variables (including intrinsic variables) contribute to fraudulent behavior, and as a consequence, experts will gain insights in the current fraud process. Second, we start from a set of confirmed fraudulent companies to initialize the propagation algorithm which other methods lack.

Table 3.4 and 3.5 outlines the average AUC score and corresponding p-values for the different estimated models, based on 10-fold cross validation. The results show that the intrinsic baseline can be improved by including network-based variables. The unipartite baseline (1) performs significantly better than the intrinsic baseline (2) at a significance level of 0.05 (except for year t_0 on short term and t_2 on long term for the Random Forests model). We conclude that network-based variables boost the performance of the fraud detection models. Remember, a link weight in a unipartite network represents the number of shared resources between two companies. Including resources as a separate entity in the network, allows us to integrate time in the

AUC Performance	Year t_0			Year t_1			Year t_2			Year t_3		
	ST	MT	LT	ST	MT	LT	ST	MT	LT	ST	MT	LT
<i>(1) Baseline - Intrinsic</i>												
<i>Random Log. Forests</i>	0.8438	0.8868	0.8232	0.8604	0.8310	0.7802	0.8473	0.8074	0.7540	0.7343	0.7381	0.7288
<i>Random Forests</i>	0.8619	0.8782	0.8183	0.8841	0.8514	0.7988	0.8247	0.8272	0.7938	0.7805	0.7792	0.7619
<i>(2) Baseline - Unipartite</i>												
<i>Random Log. Forests</i>	0.8962	0.9151	0.8650	0.9167	0.8715	0.8277	0.8953	0.8679	0.8076	0.7854	0.7702	0.7721
<i>Random Forests</i>	0.9056	0.9104	0.8691	0.9300	0.8924	0.8436	0.8816	0.8742	0.8159	0.8267	0.8125	0.8126
<i>(3) Baseline - Bipartite</i>												
<i>Random Log. Forests</i>	0.8749	0.8893	0.8517	0.8910	0.8652	0.8101	0.8698	0.8262	0.7826	0.7798	0.7652	0.7357
<i>Random Forests</i>	0.8907	0.8867	0.8726	0.9075	0.8910	0.8325	0.8670	0.8543	0.8095	0.8221	0.8250	0.7897
<i>(4) GOTCHA!</i>												
<i>Random Log. Forests</i>	<u>0.9233</u>	0.9281	0.9066	<u>0.9534</u>	0.9380	0.8943	0.9053	0.8953	0.8707	0.9035	0.8877	0.8567
<i>Random Forests</i>	0.9173	<u>0.9312</u>	<u>0.9246</u>	0.9507	<u>0.9409</u>	<u>0.9074</u>	<u>0.9069</u>	<u>0.9044</u>	<u>0.8755</u>	<u>0.9176</u>	<u>0.9114</u>	<u>0.8953</u>

Table 3.4: AUC scores of the baseline and GOTCHA! models.

AUC Performance	Year t_0			Year t_1			Year t_2			Year t_3		
	ST	MT	LT	ST	MT	LT	ST	MT	LT	ST	MT	LT
<i>(1) Intrinsic - Unipartite</i>												
<i>Random Log. Forests</i>	0.0287	0.0126	0.0119	0.0125	0.0020	0.0084	0.0014	0.0093	0.0032	0.0389	0.0006	0.0057
<i>Random Forests</i>	0.0055	0.0126	0.0071	0.0133	0.0122	0.0004	0.0165	0.0023	0.0053	0.0052	0.0147	0.0143
<i>(2) Unipartite - Bipartite</i>												
<i>Random Log. Forests</i>	0.9678	0.9957	0.9937	0.9798	0.9965	0.9931	0.9952	0.9928	0.9958	0.9535	0.9993	0.9956
<i>Random Forests</i>	0.9930	0.9870	0.9779	0.9721	0.9930	0.9997	0.9934	0.9984	0.9979	0.9978	0.9864	0.9880
<i>(3) Bipartite - GOTCHA!</i>												
<i>Random Log. Forests</i>	0.0760	0.0239	0.0029	0.0065	0.0081	0.0008	0.0325	0.0017	0.0009	0.0000	0.0115	0.0075
<i>Random Forests</i>	0.0576	0.0447	0.0177	0.0048	0.0129	0.0058	0.0184	0.0002	0.0001	0.0004	0.0005	0.0019

Table 3.5: P-values of the AUC scores.

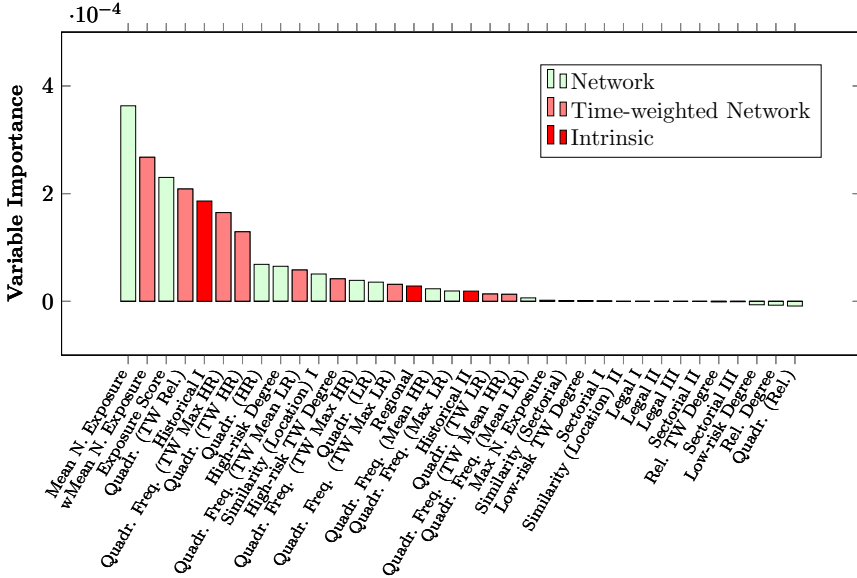


Figure 3.16: Variable Importance of Random Forests for timestamp t_3 . (TW = time-weighted; LR = low-risk; HR = high-risk; N = neighborhood).

bipartite network by the link weight between a resource and a company. We find that the bipartite baseline (3) without time-weighted edges does improve the intrinsic baseline (1), but does not outperform the unipartite baseline (2). The GOTCHA! model (4) significantly surpasses all baselines (1)-(3) in terms of AUC score from which we can conclude that features derived from a time-weighted bipartite network are an important enrichment for fraud detection models.

Ensemble methods perform better than the individual models. We compare a Decision Tree model to Random Forests, and Logistic Regression to Random Logistic Forests, and find that the highest performance in terms of AUC score is achieved with ensemble models. For brevity, we omit the model details.

Which variables (or variable categories) are mainly responsible for the performance of our GOTCHA! models? Figure 3.16 depicts the variable importance of Random Forests in year t_3 when we are testing long-term fraud. The figure shows that network-based variables are

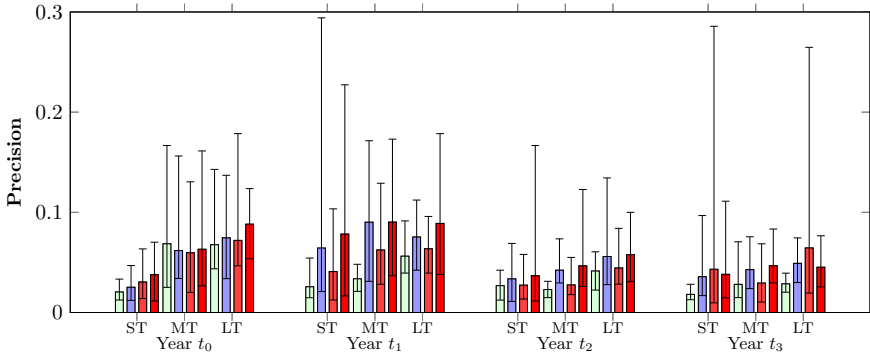
Feature	Year t_0			Year t_1			Year t_2			Year t_3		
	ST	MT	LT	ST	MT	LT	ST	MT	LT	ST	MT	LT
DIRECT FEATURES												
<i>Neighborhood Degree</i>												
<i>high-risk</i>				+		+	+	+	+	+		+
<i>low-risk</i>		+	-	-	-		-	-	-	-		+
<i>relative</i>	-	+	+	+		-		-	-			+
<i>Time-Weighted Degree</i>												
<i>high-risk</i>		-		-		-		-		-	-	-
<i>low-risk</i>	-	-		+		+	+	+	+		-	-
<i>relative</i>	+	-	-	-	-	+		+	+		-	-
<i>Quadrangles</i>												
<i>high-risk</i>		-	-					+		-		
<i>time-weighted</i>	-	+	+	+	+	+	+	+	+	+	+	
<i>low-risk</i>	+			-	-	-	-	-	-	-	-	-
<i>time-weighted</i>	-			+		+			+	+	+	+
<i>relative</i>		+	+	-		-	+		+	+	-	-
<i>time-weighted</i>	+		+	+							+	+
<i>Quadrangle Frequency</i>												
<i>mean (high-risk)</i>			-			-			-			
<i>time-weighted</i>	+			+	+	+		+	+	-		-
<i>max (high-risk)</i>			+	+				-	+		+	+
<i>time-weighted</i>	-		+	-		-	-	+	-	+	+	+
<i>mean (low-risk)</i>			+		+	-		-	-	-	-	-
<i>time-weighted</i>		-	-		-	+	-	+	+	+	+	+
<i>max (low-risk)</i>	-		-			-		-	-	+		
<i>time-weighted</i>	+	+	+	+	+	+	+	+	+	+		+
<i>Res. Similarity</i>												
<i>Sector</i>	-		+			-	+		-	-		
<i>Location (1)</i>			-		-	-	-		-	-	-	-
<i>Location (2)</i>				+	+		+		-			
INDIRECT FEATURES												
<i>Exposure Score</i>												
<i>Exposure Score</i>	+			+	+		+					
<i>Neighborhood Exposure</i>												
<i>mean</i>		-	-		+	+	-	+				+
<i>weighted mean</i>		+	+	+			+		+	+	+	
<i>maximum</i>		+	+	-		-				-		

Table 3.6: Variable importance and sign of the GOTCHA! model for social security fraud detection. A positive sign indicates a positive contribution of that variable to fraud. A negative sign means that the variable negatively impacts fraud.

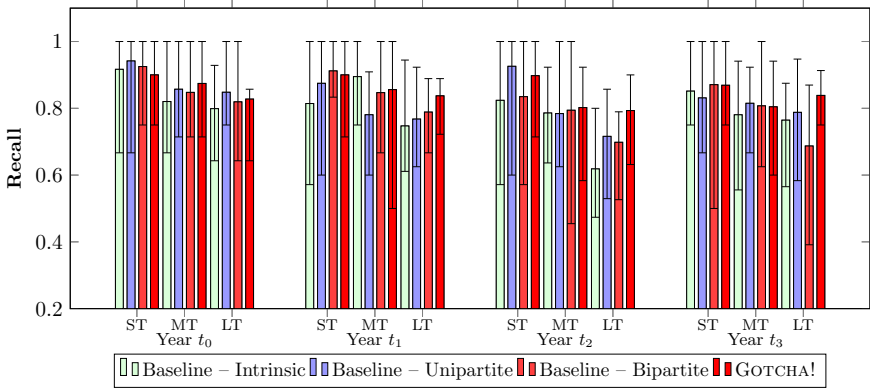
important indicators in fraud detection models. The most predictive features comprise features derived from the exposure scores and quadrangles. The exposure score captures the extent to which a node is influenced by fraud. According to Figure 3.16, aggregated features derived from the neighborhood exposure scores are more meaningful than the company's own exposure score. Quadrangles measure whether a pair of resources has been transferred between multiple companies before. The relative number of high-risk quadrangles plays an important role in the detection of fraud. Quadrangle frequency measures how many times a transfer between the same pair of resources occurred. The more companies the two resources have in common, the more suspicious the transfer is. This is in line with the process of a spider construction, where resources are continuously moved from one fraudulent company to another.

Table 3.6 summarizes the signs of the coefficients for the network parameters, based on Random Logistic Forests. Note that, in general, features aggregating high-risk characteristics are positively related with fraud, which complies with expert's intuition. One exception is the high-risk time-weighted degree which is overall negatively related with fraud. Remark that low-risk quadrangle frequency (maximum and time-weighted) positively impacts the suspiciousness of a company. This means that the shift of pairs of resources from multiple legitimate companies is anomalous which might indicate that the GOTCHA! model is able to find new spider constructions, and does not completely rely on high-risk influences from the surrounding environment. Based on the large parameter value, we find that the weighted mean neighborhood exposure score is a crucial element in the prediction of fraud, which is in accordance with Figure 3.16. We conclude that network-based features remain relevant to estimate fraud over time, irrespective of the timestamp and the time window.

Does the impact of network-based variables depend on the intrinsic variables of a company or are they independent of other intrinsic features (e.g., are network effects more pronounced for companies that operate in a high-risk sector or legal category)? We do not find significant interaction effects between intrinsic and network-based features. Network-based features play an important role, irrespective of the intrinsic characteristics of the company.



(a) Precision.



(b) Recall.

Figure 3.17: Precision and recall for the various models.

The companies outputted by our models are passed on to experts for further inspection. As experts' resources are limited, they require models that generate a short list (high precision) with as many possible fraud cases in the near future (high recall). In practice, however, we often need to make a trade-off between precision and recall. Figure 3.17 depicts the precision and recall for the baselines and the GOTCHA! model over various timestamps and time windows. Error bars indicate the minimum and maximum results achieved over the folds. Although the GOTCHA! model does not achieve a higher precision than the network models, it performs on average better than the intrinsic-only model. A pairwise t -test confirms that these results are significant ($\alpha = 0.1$), with the exception of the medium-term model for the intrinsic baseline in year t_0 . Although subtle, notice the step-wise increase in precision over the different time windows for almost all models across all timestamps and time windows. Overall we can say that shorter-term models achieve a slightly lower precision than models estimated on a longer time window. This can be explained by the lack of confirmed fraudulent cases to learn from.

In terms of recall, the GOTCHA! model and baselines follow a similar pattern: the ratio of detected companies decreases when the time window is extended. On short term, every model succeeds to identify all the fraudulent companies which is shown by the maximum error bars of one. This assesses the trade-off between recall and precision. Long-term models are more precise, which is penalized by a lower recall. Short-term models are able to identify all fraudulent companies at the expense of a lower precision.

3.5.1 Out-of-time Validation

Up until now, models were trained and validated on the same timestamp. Results prove the superiority of our proposed model compared to the baselines. However, in practice, models are trained on a previous timestamp and used in real-time. This section discusses our findings when implementing the models in this way. This is called *out-of-time validation*. The models are trained on year $t - 1$ and tested on year t .

Figure 3.18 represents a ROC analysis of an out-of-time validation

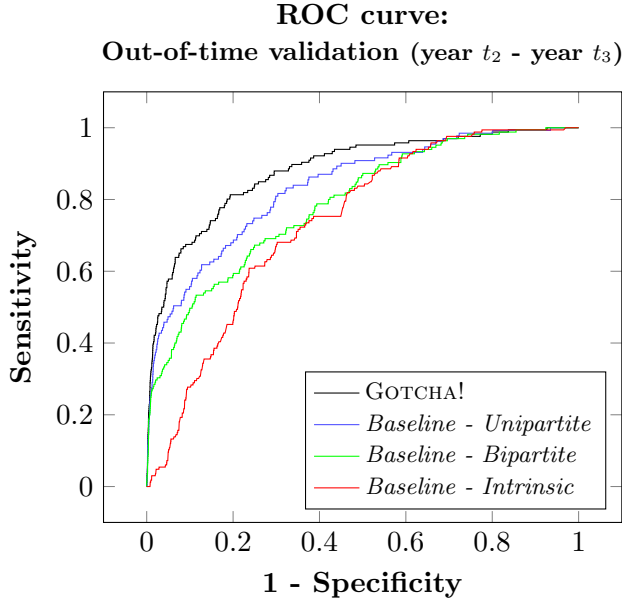


Figure 3.18: ROC analysis of the proposed approach applied in practice. All models are estimated on $year\ t_2$ and tested on $year\ t_3$.

on medium-term period between year t_2 and year t_3 (other timestamps perform similarly). The figure shows that the baselines already perform well. However, including network-based variables has a positive effect on the predictive power of the network models. In particular, when a network model gives a company a high score, it has a higher probability of begin truly fraudulent. This is shown in the steep increase in the beginning of the curve that represents the performance of the network models. Remark that the unipartite model outperforms the bipartite model. However, when we include time-weighted features, we achieve the best performance. This is shown by the GOTCHA! model.

The previous section illustrated that the medium- and long-term models perform better in terms of precision. It appears, however, that many companies detected by the short-term model will have solvency problems sometime in the future; this is shown in Table 3.7. The table represents the results of an out-of-time validation over the different timestamps, when the models are estimated on short-term fraud. We

	Total	ST Fraud	MT Fraud	LT Fraud	Fraud after analysis	Total Fraud	Bankrupt	Non- Active	Active	% detected	
t_1	<i>Baseline - Intrinsic</i>	100	4	1	5	1	11%	11	8	70	22%
	<i>Baseline - Unipartite</i>	100	15	7	2	7	31%	20	19	30	51%
	<i>Baseline - Bipartite</i>	100	16	9	4	7	36%	17	10	37	53%
	GOTCHA!	100	20	6	6	10	42%	29	10	19	71%
t_2	<i>Baseline - Intrinsic</i>	100	4	1	1	1	7%	8	7	78	15%
	<i>Baseline - Unipartite</i>	100	7	5	4	1	17%	26	12	45	43%
	<i>Baseline - Bipartite</i>	100	14	7	10	3	34%	24	7	35	58%
	GOTCHA!	100	17	4	12	7	40%	30	6	24	70%
t_3	<i>Baseline - Intrinsic</i>	100	2	0	1	0	3%	14	1	82	17%
	<i>Baseline - Unipartite</i>	100	15	3	3	0	21%	19	3	57	40%
	<i>Baseline - Bipartite</i>	100	24	6	3	0	33%	12	5	50	45%
	GOTCHA!	100	16	12	8	0	36%	20	4	40	56%

Table 3.7: Future lifecycle of detected companies. All the models are estimated on short-term fraud, but are able to identify high-risk companies after the predetermined time window.

analyze the top 100 most suspicious companies, as experts can only investigate maximum 100 companies during each time period which thus reflects model usage in practice. Table 3.7 indicates how many companies in the list commit fraud on short (*ST Fraud*), medium (*MT Fraud*) and long (*LT Fraud*) term. The model even identifies companies that will commit fraud after the time window of analysis (*Fraud after analysis*). Note that this effect diminishes over time due to the recency of the data used. GOTCHA! improves the intrinsic baseline by detecting 31%, 33% and 33% more fraudulent and high-risk cases for the respective timestamps, resulting in a higher precision and recall. The unipartite baseline is improved by 11%, 23% and 15%, respectively. The bipartite baseline is outperformed by an increase of 6%, 6% and 3% respectively. Recall that the ROC curve of the bipartite model (see Figure 3.18) did not achieve a better performance than the unipartite model. However, when analyzing the results by a limited set of the top 100 suspicious companies, we find that the bipartite model is more precise than the unipartite model. These results are consistent over all timestamps.

What happens to the other companies in the list? Some are still active (*Active*). Others are normally suspended (*Non-active*), and redeemed all their outstanding debts. Surprisingly, we see that 29%, 30% and 20% of these companies go bankrupt in the future. Although there is a lack of hard evidence and the time passed, experts are convinced that those companies are missed fraudsters. Assuming the expert is right in his/her expectation, this would mean that the detection model is able to reach higher levels of precision up to 71%, improving the intrinsic baseline by detecting up to 55% additional fraudsters over time (year t_2). There are thus reasons to believe that GOTCHA! is suitable for *retrospective* fraud detection. To summarize, by using GOTCHA!, experts can identify fraudulent companies much faster and more accurately, and potentially are still able to recover some of the losses occurring with fraud.

3.5.2 Curtailing newly originated spider constructions

Rather than detecting far-evolved spider constructions, experts are enticed to identify newly originated spider constructions as well, i.e., new fraudulent setups with only few fraudulent companies in it. In

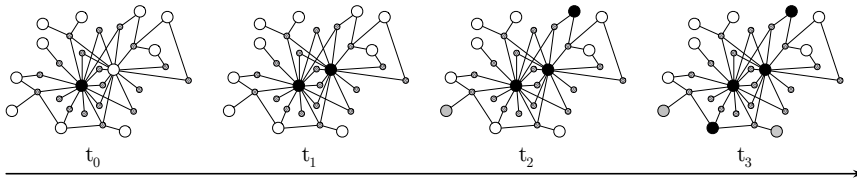


Figure 3.19: Evolution of a spider construction over time. The network represents the nodes and connections as observed at time t_0 . Only one company is fraudulent, but passes many resources to other companies. Future data shows that three extra companies will commit fraud, and two companies will go bankrupt. If we had applied GOTCHA!, our results show that we could have avoided the development of this spider construction at time t_0 .

Figure 3.19, we illustrate that GOTCHA! is also able to find such constructions. The figure shows a subgraph for timestamp *year* t_0 . One company has committed fraud during this timestamp. Note that almost all its resources flow towards another company. Indeed, this and two other companies commit fraud in the future, as well as two companies that went bankrupt. Our results show that if we had applied GOTCHA! during timestamp t_0 , we could have avoided the development of this spider construction as those companies would have appeared in the list generated on t_0 . It can be questioned whether the two bankruptcies are purely coincidental or that they are part of the fraud construction.

3.6 Conclusions

In this chapter, we improve the performance of traditional classification techniques for social security fraud detection by including domain-driven network information using GOTCHA!, a new fraud detection approach. We start by identifying the challenges that concur with fraud and design GOTCHA! such that it addresses each of these challenges to detect future fraud. In particular, we represent the network as a time-weighted bipartite graph, including two node types: companies and their resources. Starting from a limited set of confirmed fraudulent companies, we spread fraudulent influences of one node type through the network and infer an initial exposure score for both node types, i.e., the unlabeled companies

and resources. Our propagation algorithm inherits concepts from the Personalized PageRank algorithm as proposed by (Page et al., 1998), and is extended by making the following domain-dependent adjustments: (1) propagation for bipartite graphs (i.e., scoring both companies and resources), (2) emphasizing fraud, (3) dynamical behavior: use of a time-dependent weight to represent relationships between companies and resources, and to weigh the impact of fraud, (4) degree-independent propagation. The time-dependent weight allows to both anticipate and forgive the riskiness of the resources. For each company, we aggregate the properties of the direct and indirect neighborhood, and combine them with intrinsic features.

The Social Security Institution benefits from our developed approach in multiple ways: (1) *Guided search for fraud*. Instead of randomly investigating companies, the GOTCHA! algorithm produces an accurate list of companies that are worthwhile to investigate by experts. Our experiments show that our GOTCHA! network model exploits essential information for predicting future fraud more efficiently. Our model is compared to three baselines. The first one is an intrinsic-only baseline and uses only intrinsic features. The second one is a unipartite baseline, linking the companies directly to each other and aggregating resource information in the link weight. The third one extends the network representation to a bipartite graph but does not include time in the link weights. Results show that GOTCHA! produces more accurate results than the baselines in terms of their AUC score. We find that network models achieve a higher precision, although the recall is approximately the same. Hence, network-driven models reduce the set of high-risk companies passed on to the experts for further screening. (2) *Faster fraud detection*. The predictability of short-term models is surprising. Short-term models are not only able to accurately predict which companies will commit fraud in the near future, but also identify companies that perpetrate fraud many years later. This results in a higher overall precision compared to medium- and long-term models, favoring the short-term models in the fraud detection process. This also indicates that, so far, many fraudulent companies already radiate fraudulent behavior, which used to take several months, or even years, before

they were actually captured. (3) *Immediate feedback loop*. Findings of experts are immediately implemented in the models. The models update their detection process accordingly. Consequently, changes in the fraud environment are captured by the models. Our results show that models indeed use different sets of variables over time. Our future work will elaborate more on active learning, by updating the model using both correctly and incorrectly classified instances.

Although we applied our approach to social security fraud detection, the results in the Chapter 6 show that our proposed framework can be employed for the detection of other fraud types where the network can be represented as a higher order graph (n -partite graph). Chapter 6 will elaborate further on the application of this approach on credit card fraud where merchants are explicitly connected to buyers through the transactions they pursue. This work focused on finding individual companies. Another topic for future research is community detection which may find groups of suspicious companies. Community detection allows experts to gain a thorough understanding in the creation and development of spider constructions. The next chapter will discuss community detection into more details.

Chapter 4

GOTCHA'LL! Fraudulent clique detection

Given a labeled graph containing fraudulent and legitimate nodes, which nodes group together? How can we use the riskiness of node groups to infer a future label for new members of a group? This chapter focuses on social security fraud where companies are linked to the resources they use and share. The primary goal in social security fraud is to detect companies that intentionally fail to pay their contributions to the government. We aim to detect fraudulent companies by (1) propagating a time-dependent exposure score for each node based on its relationships to known fraud in the network (see Section 3.3.3); (2) deriving cliques of companies and resources, and labeling these cliques in terms of their fraud and bankruptcy involvement; and (3) characterizing each company using a combination of intrinsic and relational features and its membership in suspicious cliques. We show that clique-based features boost the performance of traditional relational models.

4.1 Introduction

So far, the fraud detection literature has mainly focused on analyzing *guilt-by-association* (Koutra et al., 2011), i.e. how relationships with fraudsters affect the probability that a person of interest will commit fraud. For example, suppose there are two fraudsters B and

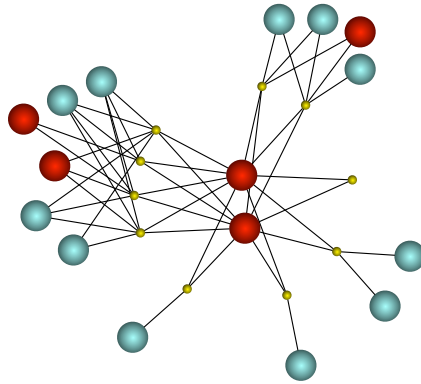


Figure 4.1: Subgraph of companies (large nodes) connected to their resources (small nodes). Fraudulent companies are dark-colored, currently-legitimate companies are light-colored. Companies form cliques (i.e., fully connected subgraphs) based on their use of the same set of resources.

C who are both connected to person A (let's say, by a friendship relation), then guilt-by-association analyzes each relationship to those neighbors separately. However, this approach does not take into account the relationships between the neighbors. In this work, we introduce *guilt-by-constellation* in which we derive suspicious cliques of nodes, and characterize each node in terms of its clique membership. A clique is a fully connected subgraph of the network where each node is connected to every other node in the subgraph. For example, suppose now that persons B and C also know each other and, as a consequence, persons A, B and C form a clique of friends. Guilt-by-constellation investigates whether this will have a stronger effect on the fraud probability of person A.

In this chapter, we address social security fraud and show a successful example of how clique-based features are an important element in inferring future fraud. We define fraud as those companies that intentionally go bankrupt in order to avoid paying tax contributions: their debt to the government will be unrecoverable. We observe that after a certain time period a new company is founded which uses almost the same resources as the previous company, like machinery, equipment, employees, address, buyers, suppliers, etc.

(see Figure 4.1). As opposed to many graph-related works, we exploit *bipartite* graphs, connecting two object types, i.e. companies and resources. We find that when a new company enters the market and inherits (a part of) the same set of resources previously associated with a fraudulent company (or companies), its fraud risk increases.

We introduce clique-based features which are shown to outperform previous approaches to this problem. In particular, we define both complete- and partial-cliques (i.e., companies share *all* or *part* of their resources with each other) and investigate: (1) Does the probability of perpetrating future fraud increase when fraudulent companies are closely connected to each other, i.e. they form a dense group where they all share the same (set of) resources? (2) If a new company enters such a group, what would we say about its probability to commit fraud? Based on these analyses and observations, we define relational and clique-based features using a graph representation. Relational features aggregate the characteristics of close neighbors by treating each of them as a separate individual regardless of their links to other neighbors (i.e., guilt-by-association). Clique-based features, on the other hand, also take into account the connectivity within the neighborhood (i.e., guilt-by-constellation). In addition to networked features (which capture *peer pressure*), we incorporate intrinsic features in our models. These intrinsic features are able to detect new types of fraud (e.g., ones that are not imitated). Remark that our models are dynamically updated, by extracting time-dependent individual and clique membership scores for each company and by re-estimating the corresponding models. We contribute by proposing a novel approach to detect fraud by:

- Defining cliques in a bipartite graph where one type of nodes (i.e., the companies) are connected to another type of nodes (i.e., the resources) (see Section 4.4.3).
- Using a time-dependent *individual exposure score* (Section 4.4.2) of every node to label cliques in the network and infer a *suspiciousness score* (Section 4.4.3) for that clique.
- Featurizing new instances based on the properties of the cliques

they belong to, and integrating the extracted features with intrinsic and relational features (see Section 4.4.4).

The remainder of the paper is organized as follows: background, related work, task description, empirical evaluation and conclusions.

4.2 Social Security Fraud

Our proposed approach will be applied to social security fraud detection. While this is only one application to integrate clique memberships in detection algorithms, we believe that a similar approach is promising on comparable application domains, like credit card fraud detection, insurance fraud, opinion fraud, and so on. In this paper, we study social security data acquired from the Belgian Social Security Institution.

Companies need to contribute employer and employee taxes to the government. We say that if a company *intentionally* goes bankrupt so as not to pay its tax contributions, the company is fraudulent. Remark that fraudulent companies often belong to a *web of fraud*, i.e. the resources of fraudulent companies are (partly) transferred to other companies which will commit fraud on their turn. E.g., fraudulent companies A, B and C operated at address p and used suppliers a and b. All those resources are now transferred to active company D. Company D is likely to commit fraud in the future. While experts have a great intuition in finding fraudulent companies, they expect that some bankruptcies classified as regular are in fact undetected fraudulent bankruptcies. We will use the network of companies and resources to judge the fraud probability or risk of a set of active companies. Resources move in bulk from one fraudulent company to another, leaving a trail of fraud. Using the company-resource network, we propose to capture *clique* behavior of the resources to cluster together companies. We will extract both a fraud and bankruptcy score for each clique: resource involvement in many *fraudulent* companies increases the fraud risk of future companies that use the same set of resources. Resource involvement in many *bankruptcies* might increase the fraud risk as well, as this may uncover an undiscovered group of fraudulent companies. We expect that currently-legitimate members

of cliques that are highly associated with fraud or bankruptcy, have a higher probability of committing fraud in the near future. In this work, we try to answer questions like (1) does *guilt-by-constellation* detect future fraud more efficiently (2) what effect does a suspicious (i.e., fraudulent) clique have on currently legitimate companies that are part of that clique? (3) what effect does a clique characterized only by (apparent) regular bankruptcies have on currently-legitimate companies that are part of that clique?

4.3 Related Work

While previous literature acknowledges the importance of network analysis in fraud detection, most research focuses on the so-called *guilt-by-association*. Many works aggregate relational information in features such as degree, proportion, count, etc. (Neville et al., 2005; Fast et al., 2007; Van Vlasselaer et al., 2013) or apply inference procedures to spread the fraudulent influence throughout the whole network (Pandit et al., 2007; Akoglu et al., 2013; Van Vlasselaer et al., under review; Akoglu et al., 2010). The aforementioned techniques neglect the density among the neighborhood of the node of interest, i.e. the extent to which the surrounding nodes are connected to each other as well. This is known as clusters, communities or cliques in the network (Newman, 2010). Cortes et al. (2001) formulated the idea to compute the *community of interest* (COI) centered around each node in the network and compare the overlap between COI's. A significant overlap with a fraudulent COI might indicate that the COI is also fraudulent. Fast et al. (2007) developed a fraud detection approach for the National Association of Securities Dealers (NASD) which uses *tribes* or clusters of representatives. The authors focused on suspicious pairs of representatives that do not comply with a normal pattern in the industry. Akoglu et al. (2013) proposed *FraudEagle*, a novel approach to spot fraudulent reviewers and reviews for opinion fraud detection. The authors used a co-clustering (Chakrabarti et al., 2004) technique to group together the top high-risk users for visualization purposes.

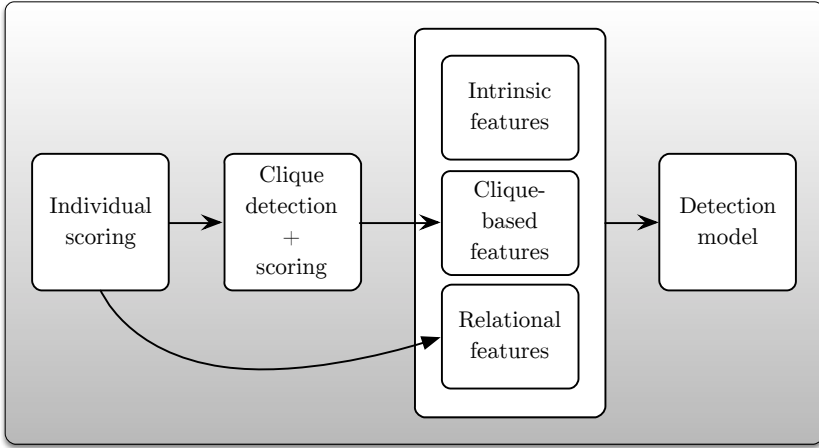


Figure 4.2: Flow-chart of detection process.

4.4 Proposed Method

4.4.1 Task description

The primary goal of this paper is to predict which currently active companies form a threat to perpetrate fraud in the future by estimating a detection model that consist of a combination of intrinsic, relational and clique-based features. Specifically, our approach consists of four steps, as illustrated in Figure 4.2:

1. *Individual scoring*: The influence of few known fraudulent (bankrupt) companies is spread through the network, deriving a time-dependent exposure score for every node. That is, each company and resource receive a score based on the presence of fraudulent (bankrupt) influence in their neighborhood.
2. *Clique detection and scoring*: Resources and companies that are frequently associated with each other are clustered in a clique. We aggregate the individual exposure scores of the involved companies and the resources to derive a suspiciousness score for each clique.

3. *Feature extraction:* We calculate the value of the features for each currently active company based on its clique memberships (31), and combine them with intrinsic (18) and relational (2) features. In total, we have 51 company characteristics.
4. *Model estimation:* We integrate all extracted features and try to predict which companies are highly sensitive to commit fraud in the future

Recall that a bipartite graph $\mathcal{G}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E})$ is a graph that connects nodes $v_1 \in \mathcal{V}_1$ to nodes $v_2 \in \mathcal{V}_2$, such that for each edge the following property holds:

$$e(v_1, v_2) \in \mathcal{E} | v_1 \in \mathcal{V}_1 \text{ and } v_2 \in \mathcal{V}_2 \quad (4.1)$$

Let \mathcal{V}_1 be the set of company nodes, and \mathcal{V}_2 the set of all resource nodes, then a company is uniquely connected to resources and vice versa. At a certain timestamp t , all companies are labeled according to their fraud involvement $\ell_f(v_i) \in \{\text{legitimate}, \text{fraud}\}$ and their bankruptcy involvement $\ell_b(v_i) \in \{\text{active}, \text{bankrupt}\}$. Those labels are used to infer an individual fraud and bankruptcy exposure score for every company and resource.

4.4.2 Individual Exposure Score

The individual exposure score is derived from Equation 3.6. Recall that the individual exposure score measures the extent to which each node is influenced by fraud. In this work, each clique is characterized by a fraud score and a bankruptcy score. In order to do so, we estimate two exposure scores: (a) one where the restart vector \vec{z} is bootstrapped with fraud, and (b) one where \vec{z} is bootstrapped with bankruptcy. Recall that (a) corresponds to the calculation in Section 3.3.3. The bankruptcy score is computed by changing the restart vector accordingly, and

$$\begin{cases} v_j = e^{-\beta h} & \text{if entry } j \text{ is a bankrupt company} \\ v_j = 0 & \text{otherwise} \end{cases}$$

with β the decay constant, and h the time passed since the company filed for bankruptcy where $h = 0$ represents a current fraud com-

pany. Afterwards, the restart vector \vec{v} is transformed to the degree-adapted starting vector \vec{z} by taking the element-wise product with the degree vector \vec{d} where d_i captures the degree of node i . The normalized degree-adapted vector is \vec{z}_{norm} which sums to 1.

4.4.3 Clique Detection and Scoring

Given present and past relationships of the companies and their resources, can we build cliques of companies and their associated resources, and score each clique based on the fraudulence or bankruptcy that resides in each clique? First, we define how we can extract all cliques in a bipartite graph. Second, we score each clique based on the exposure scores derived in the previous section.

Clique Detection According to Boccaletti et al. (2006), a community is defined as a tightly connected group of nodes or subgraph in the network. A *clique* is the strongest definition and requires that all objects of a subgraph are connected to each other. In bipartite graphs, we define a clique as a subgraph in which each type-one object is connected to each type-two object. This means that we induce a subgraph from the network in which all companies are connected to all resources and vice versa. Note that our approach only tends to find company cliques, and uses resources to associate the companies.

We apply a bottom-up approach to find all cliques in the network, which we describe in detail as follows. First, we start by enumerating all pairs of companies that share at least two resources. Since we are inclined to analyze strong relationships between companies, we require that each clique contains at least two companies and two resources. For each two companies in the data set, we list all of their shared resources. Next, we merge any two pairs of companies that share the same resources (or an intersection of the resources). If two pairs can be merged together in a complete-clique based on an *exact match* of all resources, the original pairs are deleted from the set of cliques. If the resources of two pairs of companies *partially overlap*, the two pairs are merged if both groups share at least two resources together. Those cliques are considered partial-cliques. The original pairs are kept in the set of cliques. This step is repeated until no newly created cliques can be merged together, i.e. until there is no exact or partial

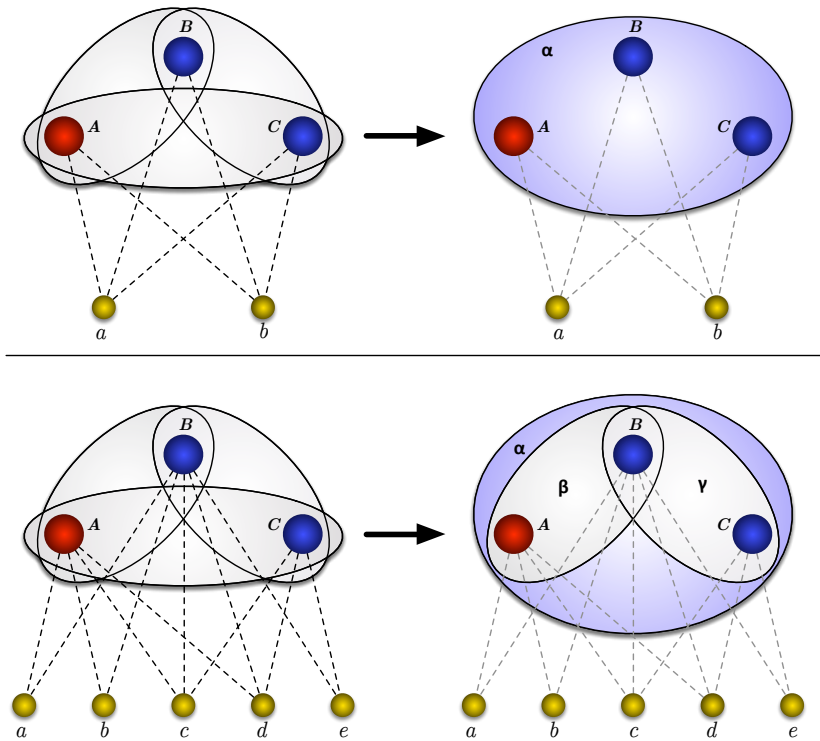


Figure 4.3: Clique detection process. Companies A , B and C share the same (set of) resources. The top figure illustrates the merging process for an exact match between pairs of companies. The original pairs are deleted from the final set. Only clique α is in the remaining set of cliques. The bottom figure represents a partial overlap between pairs of companies. Here, the original pairs β and γ , together with a new clique α are all added to the new set of cliques.

overlap between the new cliques in the set. We illustrate examples of the types of cliques this procedure creates in Figure 4.3.

Typically, a clique either consists of many companies that share only few resources or few companies with many resources. Since we do not delete partially overlapping groups, some cliques might be contained in other cliques (see the bottom figure in Figure 4.3). Thus, we are able to obtain insights in the intensity of the relationships between companies. For example, the bottom figure illustrates that company B is part of a “large” partial-clique α that connects it to companies A and C . This clique is formed based on two shared resources (c-d). Yet, company B is also contained in clique β based on four shared resources (a-d). As such, company A will have a larger influence on company B than company C , as company A is stronger connected to company B than to company C .

Clique scoring To score the cliques in terms of fraud and bankruptcy involvement, we use the individual propagated exposure score of each node. More concretely, given the known fraudulent and bankrupt companies, we characterize each clique by:

1. **COUNT**: The absolute number of fraudulent and bankrupt companies in the clique.
2. **PROPORTION**: Relative number of fraudulent and bankrupt companies in the clique.
3. **(WEIGHTED) SUM**: Sum of company (resource) fraud and bankruptcy exposure scores, optionally weighted by the number of companies (resources) in the clique.
4. **MAGNITUDE**: Total size of the clique (companies and resources) and the number of companies and the number of shared resources contained in the clique.

Note that most cliques are legitimate, not containing any company ever associated with fraud or bankruptcy before. Approximately 5% and 10% of all the identified cliques contain at least one company that was already labeled as fraudulent or bankrupt respectively. In the next section, we will introduce how we define clique-based features and characterize each company based on its clique memberships

4.4.4 Feature extraction

The detection algorithm should be able to identify high-risk companies rather than high-risk resources. Therefore, we extract features for each active company at a certain timestamp. In general, we define three sets of features: intrinsic, relational and clique-based features.

Intrinsic A company often exhibits suspicious characteristics without being influenced by others. Intrinsic features reflect company behavior as if the company was treated in isolation. Those features include a.o. sector, size, age, financial statements, etc.

Relational The fraud and bankruptcy exposure score embody the proximity of fraudulent or bankrupt influence in the company's neighborhood. A high *fraud score* indicates that many companies in the surrounding environment were already caught by perpetrating fraudulent activities. The *bankruptcy score* reveals the extent to which neighboring companies are bankrupt. These scores are computed in Section 4.4.2.

Clique-based While some companies are isolated, other companies highly interact with their neighborhood. Cliques of closely connected companies are interesting to analyze in a fraud detection context. We define three types of cliques: (1) *innocent* - this corresponds to the majority of the identified cliques (90%), (2) *bankruptcy* - approximately 10% of the cliques are associated to at least one bankrupt company, and (3) *fraudulent* - around 5% of the cliques is sensitive to fraud. The cliques captured in (3) are also part of the cliques identified in (2). Since a company can belong to multiple cliques, clique behavior is aggregated. That is, for each company we derive the following clique-based features:

1. **COUNT**: Number of cliques to which the company belongs.
2. **AVERAGE**: The characteristics as defined in Section 4.4.3 are averaged over all the cliques the company belongs to. For example, the average fraud count reflects the average number of fraudulent companies that reside in a clique.

3. **MAXIMUM:** The danger of considering the average values of all the associated cliques is that the effect of one highly suspicious clique can be filtered out by many innocent cliques. Therefore, we include the maximum value for each of the identified clique characteristics. For example, the maximum fraud count captures the maximum number of fraudulent companies that are within one clique.

In total, we create 31 clique-based features for each active company. Around 70% of all companies are not included in a clique, and have zero values for the clique-based features. While most companies are not included in a clique, approximately 75% of all fraudulent companies are member of at least one clique. All the aforementioned features are combined and passed to the detection process.

4.4.5 Detection model

The data set provided by the social security institution is a dynamic data set which includes past and present company characteristics and past and present relationships between companies and their resources. In order to validate the detection power over time, we choose to re-estimate the model for four timestamps and three time windows. More concretely, for every timestamp, we extract the features of all active companies according to Section 4.4.4, and infer a model to predict which companies will perpetrate fraud within a certain time window. We define three time windows: short, medium or long term. Based on experts' knowledge, we arbitrarily set the time windows to 6, 12 and 24 months. While short-term models are able to capture new fraud mechanisms, long-term models have more evidence to learn from. The models are re-estimated on a yearly basis, i.e. for timestamps year $t_0 - t_3$ (analogous to Chapter 3). Due to confidentiality issues, we do not specify the exact timestamp.

Fraud data sets commonly represent an extremely skewed distribution. This means that often less than 1% of the observations are fraudulent. In order to rebalance the data set, we apply SMOTE (Synthetic Minority Oversampling Technique) as proposed by Chawla et al. (2011) on the training set. Based on empirical evidence of Chawla et al. (2011), the oversampling and undersampling percent-

age are set to 400% and 200% respectively. Previous literature acknowledges that the featurization of network-related characteristics of an object might create a multitude of input features which can deteriorate the results, and suggests the use of ensemble methods to carefully select the most important features (Gallagher et al., 2008). Our models are estimated using Random Forests (Breiman, 2001). This ensemble method infers a set of decision trees by randomly selecting features. A voting process decides the label of each instance.

For each timestamp, the data set is randomly split into training and test sets. The training set is manipulated by SMOTE to address the imbalanced data distribution. The next section will discuss the results of our detection models. The results reflect the performance of the derived models on the test set.

4.5 Empirical Evaluation

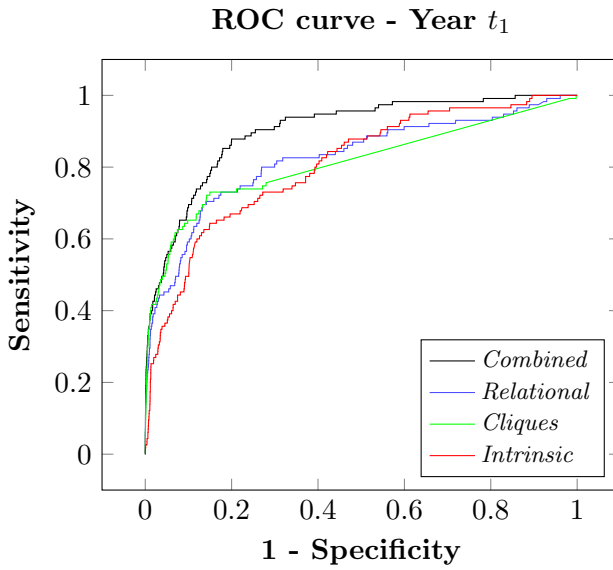
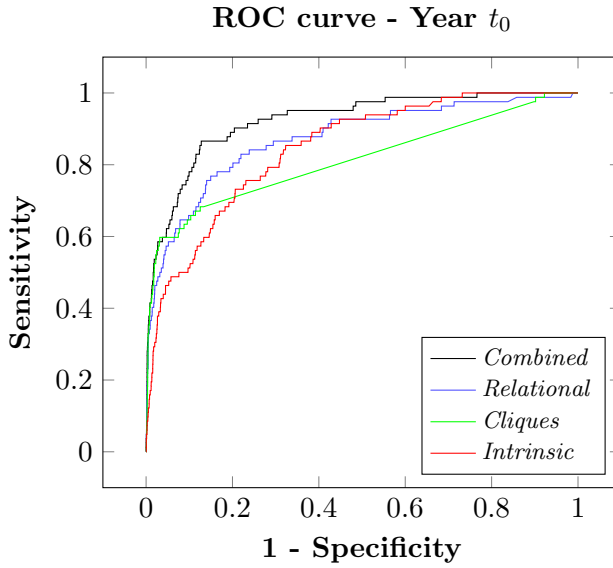
In this section, we evaluate our estimated models in terms of performance volatility over time, prediction power and precision on different time windows, and importance of the various sets of features.

4.5.1 Data Set

This approach is tested on data received from the Social Security Institution of Belgium. More details about the data can be found in Chapter 3.

4.5.2 Performance over time

Figure 4.4 depicts the ROC curves of the various timestamps of our analysis. All ROC curves present the model performance for a long-term time window. The ROC curves indicate that the combined models generate better results. In addition, a pairwise t-test confirms that the combined approach performs significantly better than the other models for all timestamps and time windows ($\alpha < 0.05$). Especially the steep slope of the curve clearly indicates that the combined model is particularly good in classifying companies as fraudulent that have



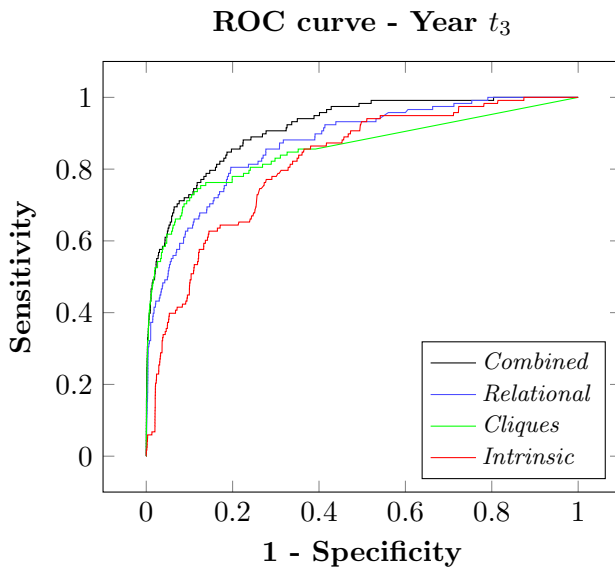
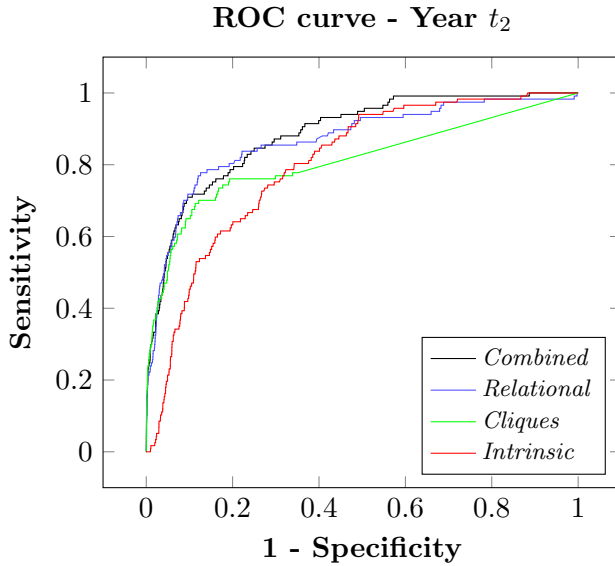


Figure 4.4: ROC curves for the different timestamps of our analysis. Notice that the combined model which includes all of the intrinsic, relational, and clique-based features outperforms the models using any one of those features alone.

AUC Performance	Year t_0			Year t_1		
	ST	MT	LT	ST	MT	LT
<i>Relational - Intrinsic</i>	0.2758	0.3500	0.0143	0.3890	0.0452	0.0012
<i>Cliques - Intrinsic</i>	0.9101	0.9941	0.7761	0.7327	0.2597	0.4968
<i>Cliques - Relational</i>	0.0888	0.0070	0.0003	0.0722	0.0764	0.0154
<i>Combined - Intrinsic</i>	0.0001	0.0041	0.0000	0.0018	0.0016	0.0000
<i>Combined - Cliques</i>	0.0012	0.0000	0.0000	0.0025	0.0001	0.0000
<i>Combined - Relational</i>	0.0117	0.0141	0.0019	0.0020	0.0003	0.0008

AUC Performance	Year t_2			Year t_3		
	ST	MT	LT	ST	MT	LT
<i>Relational - Intrinsic</i>	0.0464	0.0358	0.0125	0.0001	0.0000	0.0002
<i>Cliques - Intrinsic</i>	0.6689	0.8191	0.7526	0.0123	0.0203	0.0151
<i>Cliques - Relational</i>	0.0060	0.0027	0.0013	0.1484	0.0035	0.0026
<i>Combined - Intrinsic</i>	0.0011	0.0005	0.0001	0.0000	0.0000	0.0000
<i>Combined - Cliques</i>	0.0002	0.0000	0.0000	0.0009	0.0000	0.0000
<i>Combined - Relational</i>	0.0018	0.0232	0.0006	0.0077	0.0008	0.0000

Table 4.1: P-values of the AUC scores.

a high cut-off value (i.e., companies with a high fraud probability according to the model are in reality often sensitive to fraud). This high true positive rate is particularly important because experts have limited resources available to investigate high-risk companies, and are able to inspect only a few companies in each timestamp. Note from the figures that the clique-based and the combined model have a similar increase for high cut-off values. This might indicate that the clique-based features are mainly responsible for the high prediction power of the combined model when only a limited number of companies is selected. The relational model also follows a steep increase, but especially lifts up the curve of the combined model in the middle, when the clique-based model performs poorly.

Finally, even without network-based features, the model achieves a relatively high performance. This is illustrated by the intrinsic model in the figures. However, relational and clique-based features are an important element in boosting the performance, and should therefore be included in the detection models.

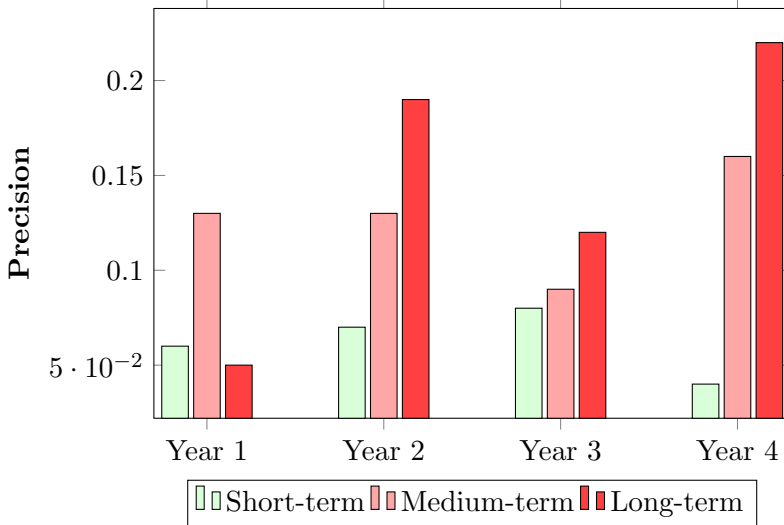


Figure 4.5: Precision of the top 100 most high-risk companies. Generally speaking, long-term models perform better than short- and medium-term models.

4.5.3 Precision

Fraud inspection is a time-consuming task and experts only select few companies for further investigation. Detection models should comply with these requirements. Given that the experts can only process approximately 100 companies in each timestamp, which companies should be inspected? Our results (from the previous section) showed that the combined model is preferred above the other models, but are the models equally *precise* in finding high-risk companies on short, medium and long term?

In Figure 4.5, we illustrate the precision for the combined model for each timestamp and each time window. Except for Year t_1 where we have limited networked data, long-term models have a higher precision. More than 20 out of 100 companies that are classified as fraudulent in Year t_4 , do indeed perpetrate fraud in the future. This means that high-risk companies already radiate suspicious behavior and characteristics even before they effectively perform fraudulent activities. The precision of the detection model is in general low. However, given

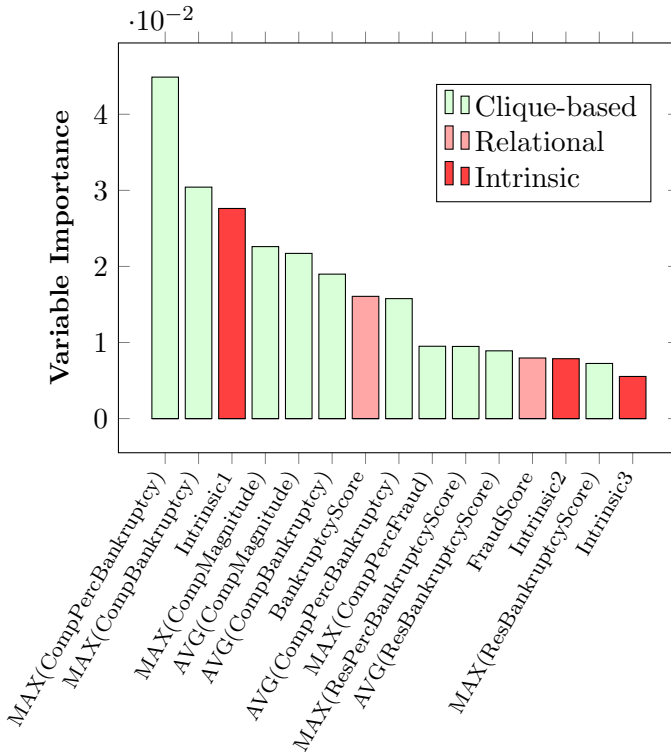


Figure 4.6: Variable Importance of the top 15 features in the combined model.

the extremely unbalanced data set of the social security institution, these are remarkable results. While our models are able to reach a precision of 22%, random classification would only result in a random precision of less than 0.2%.

4.5.4 Variable importance

We would like to assess which variables contribute to the high prediction power of the estimated detection models. Figure 4.6 illustrates that the top 15 most important variables are mainly clique-based features, although one of the intrinsic features also has a high explanatory power. Note that the most important clique-based variables are bankruptcy, rather than the fraud related variables. We can conclude

that an environment which is highly sensitive to bankruptcy might actually be a construction with hidden fraud.

4.6 Conclusions

While the challenge of fraudsters is to find the loopholes in the law, it becomes the challenge of the data analyst to characterize suspicious activities and to categorize new, similar activities as high-risk. In this work, instead of solely focusing on intrinsic behavior such as demographics, we choose to incorporate network-based features. First, we define an exposure score that quantifies both the fraudulent as well as the bankruptcy involvement of the neighborhood. Second, we form cliques of companies based on the resources they share, and score each clique in terms of the sensitivity of that clique to fraud and bankruptcy based on the computed exposure scores. For every defined timestamp, we derive features for each active company and learn a detection model to predict which companies exhibit a high risk of perpetrating fraud in the future. Our results indicate that the combination of clique-based, relational and intrinsic features achieves the best performance. Also, long-term models have a higher precision when we analyze the top 100 high-risk companies, as more data becomes available. In particular, our model is able to uncover 22% fraud cases, which is very high considering the extremely skewed class distribution ($< 0.2\%$). Moreover, we find that clique-based features have a high explanatory power and are an important indicator for future fraud.

Chapter 5

AFRAID: Active Fraud Investigation and Detection

Fraud is a social process that occurs over time. This chapter introduces a new approach, called AFRAID, which utilizes active inference to better detect fraud in time-varying social networks. That is, classify nodes as fraudulent vs. non-fraudulent. In active inference on social networks, a set of unlabeled nodes is given to an oracle (in this case one or more fraud inspectors) to label. These labels are used to seed the inference process on previously trained classifier(s). The challenge in active inference is to select a small set of unlabeled nodes that would lead to the highest classification performance. Since fraud is highly adaptive and dynamic, selecting such nodes is even more challenging than in other settings. AFRAID is applied to a real-life fraud data set obtained from the Belgian Social Security Institution to detect social security fraud, which is more thoroughly discussed in Chapter 3. Recall that, in this setting, fraud is defined as the intentional failing of companies to pay tax contributions to the government. Thus, the social network is composed of companies and the links between companies indicate shared resources. Results show that AFRAID outperforms the approaches that do not utilize active inference by up to 15% in terms of precision.

5.1 Introduction

Data mining techniques offer a good solution to find patterns in vast amounts of data. Human interaction is often an indispensable part of data mining in many critical application domains (Baesens, 2014; Baesens et al., 2015). Especially in fraud detection, inspectors are guided by the results of data mining models to obtain a primary indication where fraudulent behavior might situate. However, manual inspection is time-consuming and efficient techniques that dynamically adapt to a fast-changing environment are essential. Due to the limited resources of fraud inspectors, fraud detection models are required to output highly precise results, i.e. the hit rate of truly identified fraudsters should be maximal.

In this chapter, we investigate how *active inference* fosters the fraud detection process for business applications over time. Active inference is a subdomain of active learning where a network-based algorithm (e.g., collective inference) iteratively learns the label of a set of unknown nodes in the network in order to improve the classification performance. Given a graph at time t with few known fraudulent nodes, which k nodes should be probed – that is, inspected to confirm the true label – such that the misclassification cost of the collective inference (CI) algorithm is minimal. We consider *across-network* and *across-time* learning, as opposed to within-network learning (Kuwadekar and Neville, 2011). We combine the results of a CI algorithm with local-only features in order to learn a model at time t and predict which entities (i.e., nodes) are likely to commit fraud at time $t + 1$.

Each time period fraud inspectors have a limited budget b at their disposal to investigate suspicious instances. This budget might refer to time, money or the number of instances to be inspected. If we invest k of budget b to ask inspectors about the true label of a set of instances selected based on a selection criterion, will the total budget b be better spent? That is, do we achieve more precise results by investing a part of the budget (i.e., k) in learning an improved algorithm while the remaining budget $l = b - k$ is used to investigate the re-evaluated results, rather than by using the complete budget b to inspect the initial results without learning?

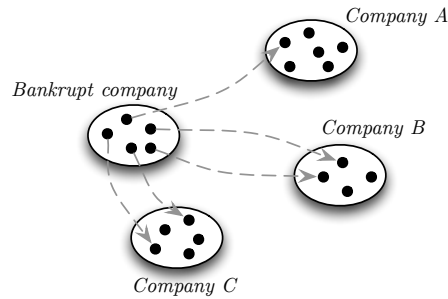


Figure 5.1: Fraud process: a fraudulent company files for bankruptcy in order to avoid paying taxes and transfers its resources to other companies that are part of the illegal setup, also known as a spider construction (see Section 3.2).

This chapter introduces AFRAID (short for: Active Fraud Invigation and Detection) which is applied to social security fraud. In social security fraud, companies set up illegal constructions in order to avoid paying tax contributions. While detection models can rapidly generate a list of suspicious companies, which k companies should be inspected such that the expected label of *all* other companies minimizes the tax losses due to fraud?

Our contributions are the following:

- Fraud is *dynamic* and evolves over time. AFRAID is a new approach for active inference in a timely manner by (1) using time-evolving graphs, and (2) weighing inspectors' decisions according to recency. (1) The influence that nodes exercise on each other varies over time. The extent of influence is captured in time-varying edge weights of the graph. Additionally, greater importance is attached to more recent fraud. (2) Given that an inspector labels a specific node as legitimate at time t , can we assume that the node is still legitimate at time $t + 1$? This chapter elaborates on how to temporarily integrate an inspector's decision in the network model, decreasing the value of the decision over time.
- A combination of *simple and fast probing strategies* is proposed to identify nodes that might possibly distort the results of a

collective inference approach. These strategies are applied to a large real-life fraud graph. Probing decisions made by (1) a committee of local classifiers, and (2) by insights provided by inspectors are evaluated. (1) A committee of local classifiers collectively votes for the most uncertain nodes without relying on domain expertise. (2) Inspectors use their intuition to formalize which nodes might distort the collective inference techniques.

- The benefits of investing k of the total budget b in learning a better model are investigated. Results show that active inference boosts the performance of the classifier in terms of precision and recall.

The remainder of the chapter is organized as follows: background (Section 5.2), network definition (Section 5.3), problem definition and active inference (Section 5.4), results (Section 5.5), related work (Section 5.6) and conclusion (Section 5.7).

5.2 Background

The data used in this study is obtained from the Belgian Social Security Institution, a federal governmental service that collects and manages employer and employee social contributions. We say that a company is fraudulent if the company is part of an illegal set up to avoid paying these taxes. Recent developments have shown that fraudulent companies do not operate by themselves, but rely on other associate companies (Van Vlasselaer et al., 2013, 2015). They often use an interconnected network, the so-called *spider constructions*, to perpetrate tax avoidance. Figure 5.1 illustrates the fraud process. A company that cannot fulfill its tax contributions to the government files for bankruptcy. If the company is part of an illegal setup, all its resources (e.g., address, machinery, employees, suppliers, buyers, etc.) are transferred to other companies within the setup. While the economical structure of the company is disbanded by means of bankruptcy, the technical structure is not changed as all resources are re-allocated to other companies and continue their current activities. Network analysis is thus a logical enrichment of traditional fraud detection techniques. For more details, see Section 3.2.1.

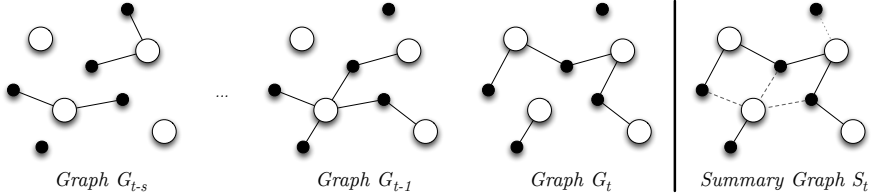


Figure 5.2: A summary graph \mathcal{S}_t at time t contains all nodes and edges observed between time t and time $t - s$.

5.3 Network definition

This section elaborates on how to use the temporal-relational data to create time-evolving fraud networks. Given relational data at time t , the corresponding graph is defined as $\mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t)$, with \mathcal{V}_t the set of nodes (or points, or vertices) and \mathcal{E}_t the set of edges (or lines, or links) observed at time t . Graph \mathcal{G}_t describes the static network at time t .

Besides current relationships, dynamic graphs keep track of the evolution of past information, e.g. nodes that are added to or removed from the network, edges that appear and disappear, edge weights that vary over time, etc. In order to include a time aspect in the network, we define the summary graph \mathcal{S}_t at time t as all the nodes and edges observed between time $t - s$ and t . Figure 5.2 depicts how a summary graph is created. For this problem setting, we include all historical information available ($s = t$), as fraud is often subtle and takes a while before the relational structure is exhibited. Although historical links hold important information about possible spread of fraud, their impact differs from more recent links. Based on work of Rossi and Neville (2012); Sharan and Neville (2007), we exponentially decay the edge weight over time as follows

$$w(i, j) = e^{-\gamma h} \quad (5.1)$$

with γ the decay value (here: $\gamma = 0.02$) and h the time passed since the relationship between node i and j occurred and where $h = 0$ depicts a current relationship. Mathematically, a network is repre-

sented by an adjacency matrix \mathbf{A} of size $n \times n$ where

$$\begin{cases} a_{i,j} = w(i,j) & \text{if } i \text{ and } j \text{ are connected} \\ a_{i,j} = 0 & \text{otherwise} \end{cases} \quad (5.2)$$

Since companies are explicitly connected to the resources they use, our fraud graph has a dual structure: every edge in the network connects a company to a resource. The network composed of n companies and m resources is called a *bipartite* network, and is of size $n \times m$. The corresponding adjacency matrix is $\mathbf{B}_{n \times m}$. As we know when a resource was assigned to a company, the edge weight corresponds to the recency of their relationship, exponentially decayed over time. In case multiple relationships exist between a company and a resource, we only include the most recent one. An edge weight with maximum value 1 refers to a current assignment.

5.4 Active Inference

Collective inference is a network analysis technique where the label of a node in the network is said to depend on the label of the neighboring nodes. In social network analysis, this is often referred to as *homophily* (McPherson et al., 2001), where one tends to adopt the same behavior as one’s associates (e.g., committing fraud if all your friends are fraudsters). A change in the label of one node might cause the label of the neighboring nodes to change which in turn can affect the label of their neighbors, and so on. As a consequence, a wrong expectation of one node strongly affects the estimated label of the other nodes. Active inference is analogous to active learning. It selects an observation to be labeled in order to improve the classification performance. While active learning iteratively re-learns and updates a classifier by the newly acquired label, active inference re-evaluates the labels of the neighboring nodes using an existing model. For a profound literature survey of active learning, we refer the reader to Settles (2009).

In this chapter, we train a set of out-of-time local classifiers $\vec{\mathcal{L}}$ at time t where each observation i is composed of a set of features \vec{x}_i derived at time $t - 1$ and the corresponding label $\mathcal{L}_i = \{\text{fraud, non-fraud}\}$ observed at time t . The set of features consists

Algorithm 1: Active inference for time-varying fraud graphs.

input : Summary graph \mathcal{S}_{t-1} and \mathcal{S}_t where $\mathcal{S}_t(\mathcal{V}_{s,t}, \mathcal{E}_{s,t})$,
time-weighted collective inference algorithm GOTCHA!,
budget k , set of labeled fraudulent nodes \mathcal{L}_{t-1} and \mathcal{L}_t .
output: Labeled nodes \mathcal{L}_{t+1} .

Initialize \mathbf{LC}_t

$\vec{\xi}_{t-1} \leftarrow \text{GOTCHA!}(\mathcal{S}_{t-1}, \mathcal{L}_{t-1});$ 5.4.1

$\mathbf{LC}_t \leftarrow \text{LC}(\vec{x}_{t-1}[\vec{a}_{t-1}, \text{aggr}(\vec{\mathcal{N}}_{t-1}), \vec{\xi}_{t-1}], \mathcal{L}_t);$

Active inference

$\ell \leftarrow 0$

while $\ell < k$ **do**

$\vec{\xi}_t \leftarrow \text{GOTCHA!}(\mathcal{S}_t, \mathcal{L}_t);$

$\mathcal{L}_{t+1} \leftarrow \text{LC}_t(\vec{x}_t[\vec{a}_t, \text{aggr}(\vec{\mathcal{N}}_t), \xi_t], \mathcal{L}_t);$

 Select node v_i to probe; 5.4.2

if $y(v_i) = \text{fraudulent}$ **then**

$\mathcal{L}_t(v_i) \leftarrow (\text{fraud}, t);$ 5.4.3

else if $y(v_i) = \text{non-fraudulent}$ **then**

$\forall v_j \in \mathcal{N}_i : w(j, i) = 0;$ 5.4.3

end

$\ell \leftarrow \ell + 1$

end

of (1) intrinsic features \vec{a}_i , and (2) neighborhood features (see Section 5.4.1). Intrinsic features are features that occur in isolation and do not depend on the neighborhood. The intrinsic features that describe the companies in this analysis include age, sector, financial statements, legal seat, etc. The neighborhood features are derived by a collective inference technique. We apply each classifier LC_m to observations from time t in order to predict which observations are likely to commit fraud at time $t + 1$. In active inference, inspectors are asked to provide their most probable label at time $t + 1$ which is directly integrated in the current network setting to infer a new expectation of the neighbors' label. This is *across-time* and *across-network* learning. Recall that inspectors have a total budget b at their disposal each timestamp, and are able to invest $k < b$ budget in improving the current collective inference algorithm. Using the updated feature set, the LC re-learns a new estimate of each of the nodes' fraud probability. However, as inspectors' decisions are only temporarily valid, we temporally weigh the belief in a decision, by decreasing its value in time. Algorithm 1 provides more details on the procedure for active inference in time-varying fraudulent networks, and will be discussed in the remainder of this section.

5.4.1 Collective Inference Technique

Many collective inference algorithms have been proposed in the literature (see Sen et al. (2008) for an overview). In this chapter, we employ a set of local classifiers that evaluates the classification decision on both intrinsic and neighborhood features. For the neighborhood features, we make a distinction between (1) local neighborhood features and (2) a global neighborhood feature. The local neighborhood features are based on the labels of the direct neighbors. Recall that in our bipartite graph only the labels of the companies are known, and that the first order neighborhood of each company is composed of its resources. We define the direct neighborhood of a company as the company's resources and their associations. As the number of neighbors for each node differs, the neighborhood labels are aggregated in a fixed-length feature vector (here: length = 3) (Sen et al., 2008). The following aggregated features $\text{aggr}(\mathcal{N}_i)$ are derived from the network for each company i .

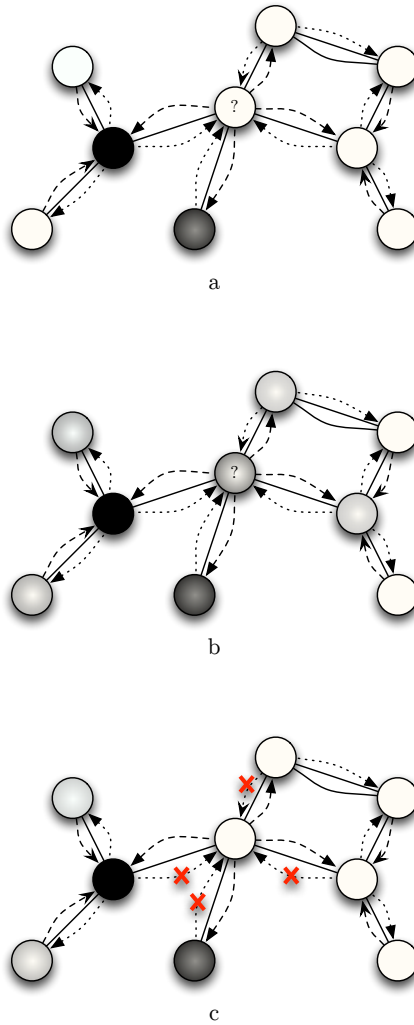


Figure 5.3: Time-weighted collective inference algorithm. (a) At time t , two companies in the subgraph are fraudulent. The intensity of the color refers to the recency of the fraudulence. (b) Propagation of fraud through the network by GOTCHA!'s propagation algorithm. (c) Cutting the incoming edges after probing node '?' and confirming its non-fraudulent label.

- **Weighted Sum:** the number of fraudulent companies associated through a similar resource, weighted by the edges.
- **Weighted Proportion:** the fraction of fraudulent companies associated through a similar resource, weighted by the edges.
- **Weighted Mode:** binary indicator whether the neighborhood is mainly fraudulent or non-fraudulent.

Note that the weighted sum and proportion correspond to the time-weighted degree and the relative degree in Section 3.3.4. In order to reduce complexity, no features are included that capture the neighborhood’s connectivity (e.g., triangles, quadrangles).

If – due to probing companies – the label of one of the neighbors changes, the local neighborhood is directly impacted. After each iteration of Algorithm 1, the local neighborhood features are updated.

The global neighborhood feature is inferred using GOTCHA!’s propagation algorithm which is more thoroughly discussed in Section 3.3.3. Figure 5.3a and 5.3b illustrate how fraud propagates through a network.

5.4.2 Probing strategies

Given a set of observations with an estimated label by a local classifier LC, which observation should be probed (i.e., checked for its true label) such that the predicted label of the other observations are maximally improved? Recall that the feature set of each observation from which the LC estimates the label depends on the neighborhood of that observation. Any change made in the label of one node has a direct impact on the feature set of the neighbors. We define five probing strategies: committee-based, entropy-based, density-based, combined and random strategy.

(1) Committee-based strategy

Rather than to rely on the decision of one LC, many LCs decide on which node to pick in a committee-based strategy. An often used

approach is uncertainty sampling. That is, sample that observation about which all the members of the committee are the most uncertain. Our committee is composed of the set of local classifiers \mathcal{LC} . Each local classifier \mathcal{LC}_m expresses how confident it is in the estimated label of each observation by means of a probability. Sharma and Bilgic (2013) distinguishes between two types of uncertainty: most-surely and least-surely uncertainty. The most-surely uncertain node is that node for which the estimated probabilities of the local classifiers provide equally strong evidence for each class. For example, when half of the committee members vote for **fraud**, and the other half vote for **non-fraud**, we say that the committee is most-surely uncertain about the node's label. Least-surely uncertainty refers to that node for which the estimated probabilities do not have significant evidence for either class. The committee is least-surely uncertain about a node's label if the probability of the node to belong to a class is close to 0.5 for many classifiers. Based on Sharma and Bilgic (2013), we combine positive (i.e., belonging to class **fraud**) and negative (i.e., belonging to class **non-fraud**) evidence learned from the set of models. Each local classifier \mathcal{LC}_m assigns a fraud estimate to each node x . A model is in favor for a positive label of node x when $P_x(+|\mathcal{LC}_m) > P_x(-|\mathcal{LC}_m)$, then $\mathcal{LC}_m \in \mathcal{P}$ for node x , otherwise $\mathcal{LC}_m \in \mathcal{N}$. Evidence in favor of node x being fraudulent is

$$E^+(x) = \prod_{\mathcal{LC}_m \in \mathcal{P}} \frac{P_x(+|\mathcal{LC}_m)}{P_x(-|\mathcal{LC}_m)} \quad (5.3)$$

Evidence in favor of node x being a non-fraudulent is

$$E^-(x) = \prod_{\mathcal{LC}_m \in \mathcal{N}} \frac{P_x(-|\mathcal{LC}_m)}{P_x(+|\mathcal{LC}_m)} \quad (5.4)$$

The most-surely uncertain node (MSU) in the set of unlabeled nodes \mathcal{U} is the node which has the maximal combined evidence.

$$x^* = \operatorname{argmax}_{x \in \mathcal{U}} E(x) = E^+(x) \times E^-(x) \quad (5.5)$$

The least-surely uncertain node (LSU) is the node which has the

minimal combined evidence.

$$x^* = \operatorname{argmin}_{x \in \mathcal{U}} E(x) = E^+(x) \times E^-(x) \quad (5.6)$$

We define four types of committee-based strategies to sample nodes: (1) most-surely uncertain (MSU), (2) least-surely uncertain (LSU), (3) most-surely uncertain using the best performing local classifiers (MSU+) and (4) least-surely uncertain using the best performing local classifiers (LSU+). We implemented sampling strategy (3) and (4), as we found that some poorly performing classifiers fail to appropriately weigh the feature set and distort the results of the uncertainty sampling. Therefore, in MSU+ and LSU+, only well-performing committee members (i.e., above average precision of all local classifiers) are allowed to vote on the node to be probed.

(2) Entropy-based strategy

Fraud is highly imbalanced, having only a limited set of confirmed fraudulent nodes available. However, our network exhibits statistically significant signs of homophily (p -value < 0.02) which indicates that fraudulent nodes tend to cluster together. Some non-fraudulent nodes lie on the boundary between a cluster of fraudulent and non-fraudulent nodes. The entropy-based strategy measures the impurity of the neighbors' labels and identifies these nodes that are associated with a similar amount of fraudulent and non-fraudulent nodes, and

$$\begin{aligned} x^* &= \operatorname{argmax}_{x \in \mathcal{U}} \operatorname{Entropy}(x) \\ &= -d_{rel,x}^{(2)} \log(d_{rel,x}^{(2)}) - (1 - d_{rel,x}^{(2)}) \log(1 - d_{rel,x}^{(2)}) \end{aligned} \quad (5.7)$$

with $d_{rel,x}^{(2)}$ the fraction of fraudulent nodes associated with node x in the second-order neighborhood (i.e., the companies) at time t .

(3) Density-based strategy

Spider constructions are subgraphs in the network that are more densely connected than other subgraphs. The density-based strat-

egy aims to find those nodes of which the neighborhood is highly interconnected.

$$x^* = \operatorname{argmax}_{x \in \mathcal{U}} \frac{\# \text{ of observed edges}}{\# \text{ of all possible edges}} \quad (5.8)$$

(4) Combined strategy

Based on experts' expertise, the combined strategy searches for companies that are located in (1) a dense neighborhood (= high density), and (2) an impure neighborhood (= high entropy). Evidence is aggregated by multiplication (Sharma and Bilgic, 2013). The node with the maximum value for the combined strategy is selected for probing, and

$$x^* = \operatorname{argmax}_{x \in \mathcal{U}} \text{Combined}(x) = \text{Entropy}(x) \times \text{Density}(x) \quad (5.9)$$

(5) Random strategy

The random probing strategy randomly picks a node in the network for probing.

Probing strategy (1) does not rely on domain expertise, while (2)-(4) are guided by experts' insights. Strategy (5) is employed as baseline.

5.4.3 Temporal weighing of label acquisition

Based on previous selection technique, the probed node is sent to inspectors for further investigation. Inspectors will confirm the true label of the node. Recall that in our setting only companies can be directly attributed to fraud, resources cannot be passed to the inspectors for investigation. Inspectors will thus only label company nodes. At label acquisition, two scenarios can occur for each node that is probed:

Classified as fraudulent

In this case, the node is added to the bag of fraudulent nodes, and affects (1) the local neighborhood features of the neighbors, and (2)

the global neighborhood feature of all nodes. (1) Up until now, the sampled node was considered to be non-fraudulent. Hence, we locally update the feature set of the company’s neighbors. (2) The starting vector of GOTCHA!’s propagation algorithm (see Section 3.3.3) is re-created, treating the node as a fraudulent one. Assume that node x^* is probed and classified as fraudulent, then

$$v_{x^*} = 1$$

and the starting vector \vec{z} is updated accordingly, such that

$$\vec{z} = \vec{v} \odot \vec{d}$$

with \vec{d} the degree vector. The normalized vector is \vec{z}_{norm} . The global neighborhood feature for each node is then updated by Equation 3.6.

Classified as non-fraudulent

Inspectors do not find any evidence that this node will be involved in fraud at time $t + 1$. However, this does not imply that the node will always be non-fraudulent. The inspectors’ decision is only valid for a limited time period. This decision does not impact the local neighborhood features, as the node was treated as non-fraudulent before. It only temporarily affects the exposure scores computed by GOTCHA!’s propagation algorithm. If we know for certain that node i is legitimate at time t – based on e.g., inspectors’ labeling – the node should block any fraudulent influence passing through. By temporarily cutting all the incoming edges to node i , node i will not receive any fraudulent influences, and as a result cannot pass fraudulent influences to its neighbors. The edge weight in the adjacency matrix \mathbf{M} is changed as follows:

$$\forall j \in \mathcal{N}_i : w(j, i) = (1 - e^{-\beta d})e^{-\alpha h} \quad (5.10)$$

with α and β decay values, t the time passed since the decision that i is non-fraudulent where $d = 0$ if it is a current decision, and h

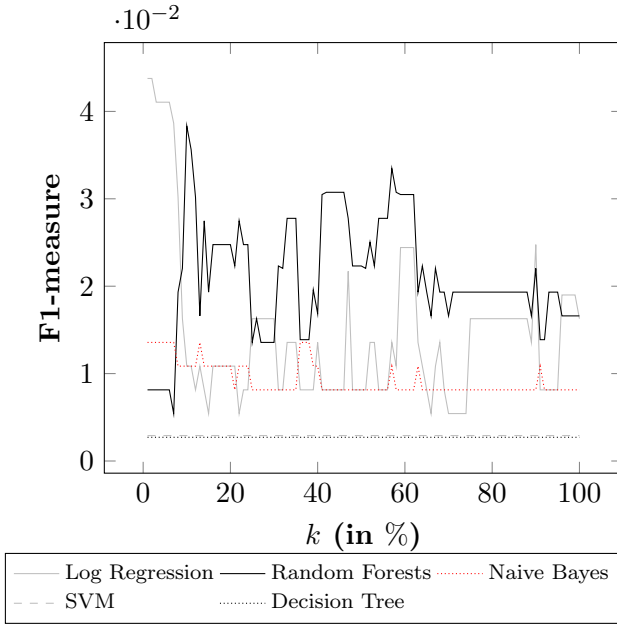


Figure 5.4: Model performance of active inference on time t for probing strategy MSU+.

is the time passed since a relation between i and j occurred. Remark that only incoming edges are cut from the non-fraudulent node. The outgoing edges are still intact. This mitigates the effect of fraud on its neighbors. This is illustrated in Figure 5.3c.

5.5 Results

AFRAID, a new approach for active inference in time-evolving graphs is applied to a real-life data set obtained from the Belgian Social Security Institution. We use historical data for evaluation, allowing us to appropriately interpret results and the value of active inference for our application domain. We trained five local classifiers (i.e., Logistic Regression, Random Forests, Naive Bayes, SVM and Decision Tree) for two timestamps t_1 and t_2 . Due to a non-disclosure agreement, the exact timestamps of analysis is omitted. The local classifiers $\mathcal{L}\mathcal{C}$ of time t_1 are learned using data features of time t_0 and their correspond-

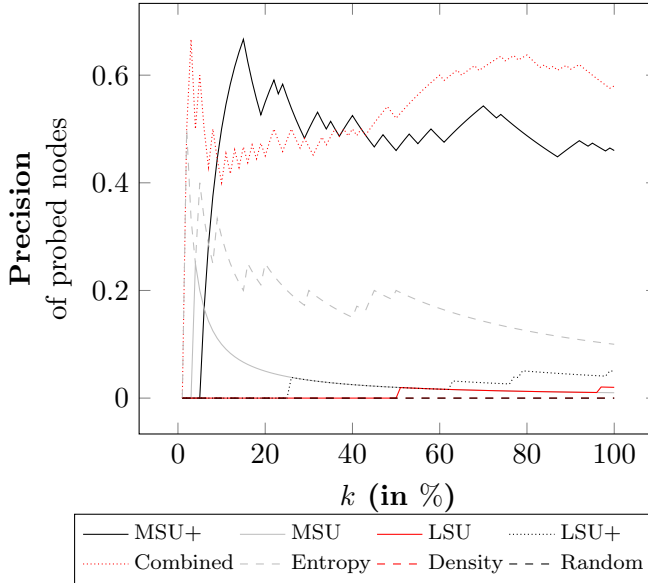


Figure 5.5: Precision achieved by the probing strategies.

ing label at time t_1 . The model is tested on data features of time t_1 aiming to predict the corresponding label at time t_2 . Because inspection is time-consuming, the number of companies that are passed on for further inspection is limited. In this problem setting, we focus on the top 100 most probable fraudulent companies, out of more than 200k active companies, and evaluate model performance on precision, recall and F1-measure.

Figure 5.4 shows the F1-measure of the local classifiers obtained when investigating the top 100 most likely fraudulent companies in function of the percentage of companies labeled of the budget b . Precision and recall follow a similar pattern, as the total number of companies that committed fraud between t_1 and t_2 reaches approximately 200 ($< 1\%$). The probing strategy used is (MSU+). While Naive Bayes, SVM and Decision Tree are not significantly impacted, the probing strategy is able to identify nodes that change the top 100 most probable frauds for Logistic Regression and Random Forests. Although the benefits for Logistic Regression are not pronounced, the precision achieved by Random Forests increases from 3% up to 15%.

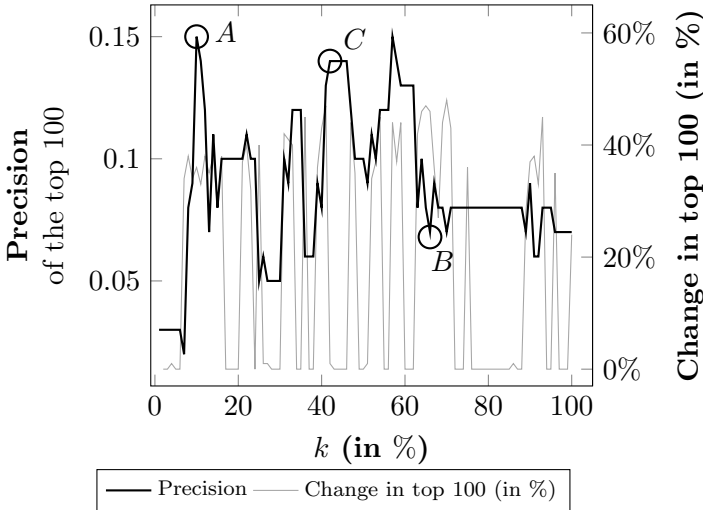


Figure 5.6: Changes in precision for Random Forests compared to changes in the evaluation set when using probing strategy MSU+.

Figure 5.5 depicts the precision achieved by the probing strategies themselves. On average, more than 50% of the probed nodes are labeled by the inspectors as fraudulent. Considering that there are only 200 out of 200k companies that commit fraud during the next time period, this is translated in an increase of approximately 25% recall. These results indicate that the probing strategy on its own is a powerful approach to detect many frauds.

Remark that the curves in Figure 5.4 vary a lot. This is mainly due to the shift in the top 100 companies, depending on which node is probed. Figure 5.6 illustrates how the changes in precision (black curve) can be explained by changes in the top 100 most suspicious companies (gray curve, in %). We distinguish three scenarios, as indicated in the figure: (A) The sampled node causes an increase in precision. The sampled node is labeled as non-fraudulent hereby correctly blocking fraudulent influence to the rest of its neighborhood, or the sampled node is labeled as fraudulent intensifying the spread of fraud towards its neighborhood. (B) The sampled node deludes the CI technique. This can be explained by the innocent resources often attached to illegal setups. (C) The sampled node does not have any

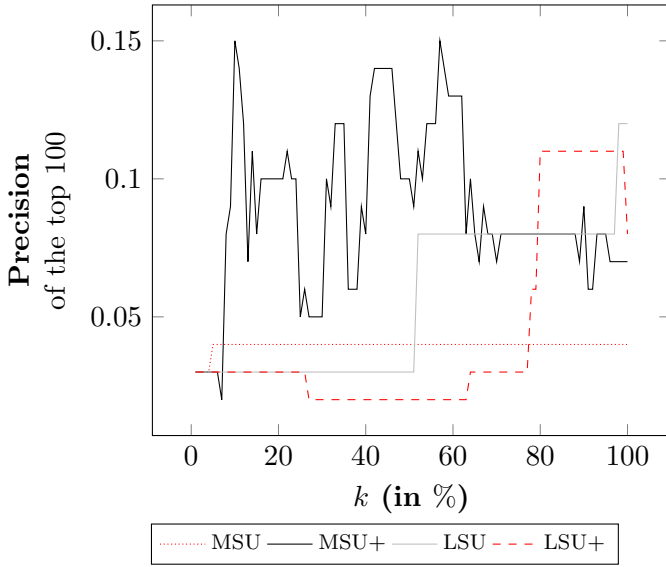


Figure 5.7: Precision of the committee-based probing strategies.

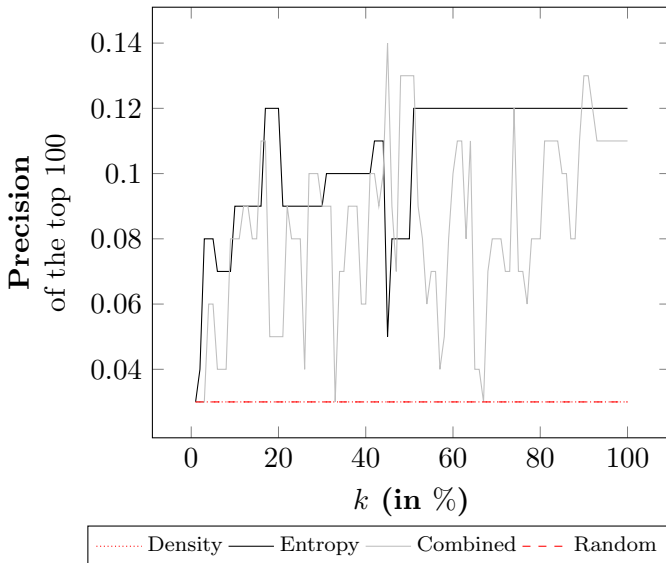


Figure 5.8: Precision of the experts-based probing strategies.

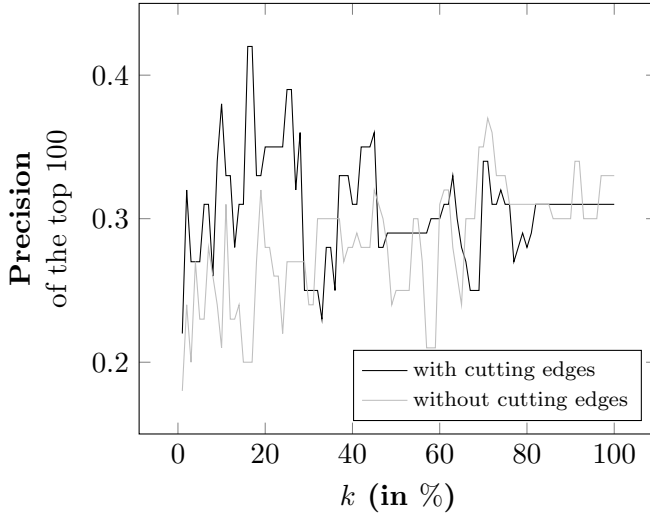


Figure 5.9: Precision when inspectors’ decisions are weighted in time.

influence on the top 100, and can be seen as a lost effort.

Figure 5.7 and 5.8 compare the different probing strategies. We distinguish between committee-based strategies (Figure 5.7) and strategies using experts’ experience (Figure 5.8). In general, MSU+, the Entropy-based and Combined strategy achieve approximately the same precision. Consistent with the results of Sharma and Bilgic (2013), the probing strategies LSU and LSU+ do not contribute to learning, as well as the Density and Random strategy. Surprisingly, we observed that the MSU strategy which uses all classifiers does not perform well. When we apply a committee-based strategy composed of the best members or advanced experts’ strategies (i.e., Entropy-based and Combined), we achieve the best performance. We can conclude that a committee of local classifiers can mimic experts’ insights, which is often preferred in order to make unbiased inspection decisions.

Finally, we evaluate how model performance is affected by cutting the edges, and gradually re-integrating their influence in time. Figure 5.9 shows the precision at time t_2 with and without integrating the edge cuts of time t_1 . Especially when the probing budget is limited, the precision is positively impacted. When more budget is in-

vested in probing, the effects of transferring decisions of the previous timestamps are similar to the results achieved when no previous edge cuts are taken into account. A p-value of < 0.001 shows that if the budget is lower than 30, the performance is significantly higher when cutting the edges. If more budget than 30 is used, the difference is not significant anymore.

5.6 Related Work

Active learning iteratively learns a classifier by selecting unlabeled observations to be labeled, and update the classifier accordingly. The label is assigned by an “oracle”, which often refers to human interaction present in the learning process. Although active learning is widely explored in the literature (see Settles (2009) for an overview), it is only recently applied to networked data. As network-based features often rely on the neighbors, an update in the neighborhood causes some features to change. This is collective classification, and is proven to be useful for fraud detection in (Pandit et al., 2007; Akoglu et al., 2013). Active inference refers to the process of iteratively sampling nodes such that the collective classification prediction of all other nodes is optimized. Most studies focus on within-network classification, in which a subset of the nodes are labeled and the labels of the other nodes need to be decided. The goal is to select the most informative (set of) nodes to sample. Rattigan et al. (2007) suggest to select those nodes for probing that lie central in the network and impact other nodes more significantly. Macskassy (2009) uses the Empirical Risk Minimization (ERM) measure such that the expected classification error is reduced. The Reflect and Correct (RAC) strategy (Bilgic and Getoor, 2008, 2009) tries to find misclassified islands of nodes by learning the likelihood of each node belonging to such island. In (Bilgic et al., 2010), the authors propose ALFNET combining both content-only (or intrinsic) features with features derived from the network. They use local disagreement between a content-only and combined classifier to decide which node to probe in a cluster. As opposed to within-network learning, Kuwadekar and Neville (2011) applied active inference to across-network learning. That is, their Relational Active Learning (RAL) algorithm is bootstrapped on a fully-labeled

network and then applied to a new unlabeled network. Samples are chosen based on a utility score that expresses the disagreement within an ensemble classifier.

5.7 Conclusion

This chapter discussed how active inference can foster classification in time-varying networks. A new active inference approach for time-evolving graphs, called AFRAID, is applied to a real-life data set obtained from the Belgian Social Security Institution with as goal to detect companies that are likely to commit fraud in the next time period. Fraud is defined as those companies that intentionally do not pay their taxes. Given a time-varying network, we extracted (1) intrinsic features and (2) neighborhood features. A change in the label of one node might impact the feature set of the neighbors. This is collective classification. We investigated the effect on the overall performance of a set of classifiers, when we are able to select a limited set of nodes to be labeled. Although the domain requirements are rather strict (i.e., only 100 out of >200k companies can be investigated each time period), Random Forests benefit the most from active inference, achieving an increase in precision up to 15%. We investigated different probing strategies to select the most informative nodes in the network and evaluate (1) committee-based and (2) expert-based strategies. We find that committee-based strategies using high-performing classifiers result in a slightly better classification performance than expert-based strategies which is often preferred in order to obtain an unbiased set of companies for investigation. We see that the probing strategies on their own are able to identify those companies with the most uncertainty, resulting in a total precision of up to 45%.

Chapter 6

APATE: Anomaly Prevention using Advanced Transaction Exploration

In the last decade, the ease of online payment has opened up many new opportunities for e-commerce, lowering the geographical boundaries for retail. While e-commerce is still gaining popularity, it is also the playground of fraudsters who try to misuse the transparency of online purchases and the transfer of credit card records. This chapter proposes APATE, a novel approach to detect fraudulent credit card transactions conducted in online stores. APATE combines (1) intrinsic features derived from the characteristics of incoming transactions and the customer spending history using the fundamentals of RFM (Recency - Frequency - Monetary); and (2) network-based features by exploiting the network of credit card holders and merchants and deriving a time-dependent suspiciousness score for each network object. Results show that both intrinsic and network-based features are two strongly intertwined sides of the same picture. The combination of these two types of features leads to the best performing models which reach AUC-scores higher than 0.98.

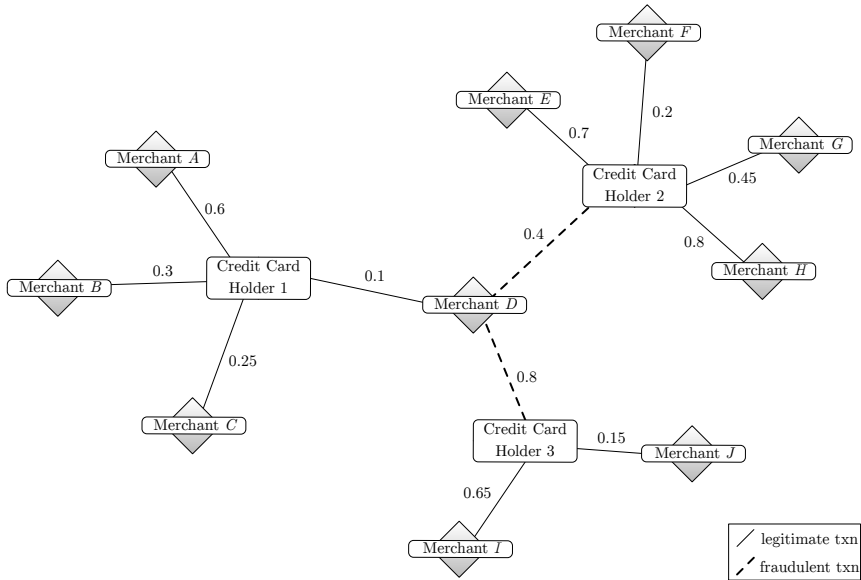


Figure 6.1: Toy example of a credit card fraud network. Weights depict the recency of the transaction between the merchant and credit card holder.

6.1 Introduction

In recent years, e-commerce has gained a lot in popularity mainly due to the ease of cross-border purchases and online credit card transactions. Customers are no longer bound by the offers and conditions of local retailers, but can choose between a multitude of retailers all over the world and are able to compare their products, offered quality, price, services, etc. in just a few clicks. While e-commerce is already a mature business with many players, security for online payment lags behind. Recently, the European Central Bank (ECB) reported that the value of card fraud increased in 2012 by 14.8% compared to the year before ECB (2014). The main reason is the strong growth in online sales, resulting in many “card-not-present” transactions (CNP), a means of payment that catches the attention of illicit people who try to mislead the system by pretending to be someone else. As a consequence, credit card issuers need an automated system that prevents the pursue of an

incoming transaction if that transaction is highly sensitive to fraud, i.e. the transaction does not correspond to normal customer behavior.

This chapter focuses on *automatically* detecting online fraudulent transactions. Data mining offers a plethora of techniques to find patterns in data, distinguishing normal from suspicious transactions. A key challenge in fraud is to appropriately deal with the atypical character of fraud. That is, there are many legitimate transactions and only few evidence of fraudulent transactions to learn from, which complicates the detection process. Carefully thinking about and creating significant characteristics that are able to capture irregular behavior, is an essential step in an efficient fraud detection process. In this chapter, both intrinsic and network-related features are again combined. Intrinsic features analyze the transaction as if it is an isolated entity, and compare whether the transaction fits in the normal customer profile. RFM attributes – Recency, Frequency and Monetary Value – of the credit card holder’s past transactions are derived to create those features. Network-based features, on the other hand, characterize each transaction by creating and analyzing a network consisting of credit card holders and merchants which are related by means of transactions. A sample network is given in Figure 6.1. Some merchants and credit card holders are frequently involved in fraud.

Inspired by Equation 3.6, a collective inference algorithm is developed to spread fraudulent influence through the network by using a limited set of confirmed fraudulent transactions and decide upon the suspiciousness of each network object by deriving an exposure score – i.e. the extent to which the transaction, the associated account holder and the merchant are exposed to past fraudulent influences.

Hence, APATE is proposed, a novel, automated and real-time approach to tackle credit card transaction fraud by mapping past purchasing patterns and customer behavior into meaningful features and compare those features with the characteristics of a new, incoming transaction. Supervised data mining techniques are applied to uncover fraudulent patterns from a real-life credit card transaction data set obtained from a large credit card issuer in Belgium. APATE complies with the six-second rule, i.e. within six seconds the algorithm needs

to decide whether the transaction should or should not be pursued. This chapter contributes by propping a new propagation algorithm to propagation fraud from the network edges (i.e., the transactions) towards all the network components (i.e., the credit card holders and merchants) and derive for each transaction network-based features. Those features are combined with a set of intrinsic features to feed the learning algorithms. The APATE fraud detection model is able to *dynamically* adapt to a changing environment and prevents that fraudsters invent new ways to perpetrate their illegal activities.

Throughout this chapter, the following questions will be answered: (1) Is a new incoming transaction in line with normal customer behavior, i.e. does it correspond to regular spending patterns of that customer in terms of (a) *frequency* or the average number of transactions over a certain time window (b) *recency* or the average time in between the current and previous transaction and (c) *monetary value* or the amount spent on that transaction? (2) Which merchants, credit cards and transactions are sensitive to fraud? Given past network-based information between merchants and credit card holders through the transactions made, how can a suspiciousness score be derived for (a) merchants indicating which merchants are often related to fraud, and as a consequence, form a risk of pursuing future fraudulent transactions; (b) credit card holders who act irregularly or whose credit card is stolen and (c) transactions by combining evidence of the associated credit card holder and merchant; (3) Does APATE, a new detection approach which combines both intrinsic and network-based features, significantly boost the performance over traditional intrinsic-only models, and if so, which specific set of features contribute in detecting efficiently fraud?

We propose APATE (short for: Anomaly Prevention using Advanced Transaction Exploration), a novel, automated and real-time approach to tackle credit card transaction fraud by mapping past purchasing patterns and customer behavior into meaningful features and compare those features with the characteristics of a new, incoming transaction. Supervised data mining techniques are applied in order to uncover fraudulent patterns from a real-life credit card transaction data set obtained from a large credit card issuer

in Belgium. This approach complies with the six-seconds rule, i.e. within six seconds the APATE algorithm needs to decide whether the transaction should or should not be pursued. We contribute by proposing a new propagation algorithm to propagate fraud from the network edges (i.e., the transactions) towards all the network components (i.e., the credit card holders and merchants) and derive for each transaction network-based features. Those features are combined with a set of intrinsic features to feed the learning algorithms. The developed fraud detection model is able to *dynamically* adapt to a changing environment and continues to operate under the condition that fraudsters invent new ways to perpetrate their illegal activities.

The remainder of this chapter is organized as follows. The credit card fraud domain is introduced in Section 6.2. Section 6.3 discusses the proposed methodology, and focuses on intrinsic and network-based feature extraction (Sections 6.3.1 and 6.3.2). In Section 6.4, the obtained results are summarized and analyzed more thoroughly. Section 6.5 concludes this chapter.

6.2 Credit Card Transaction Fraud

6.2.1 Background

Credit card fraud detection is a widely studied research domain. Bhatla et al. (2003) and Delamaire et al. (2009) distinguishes between various types of fraud like application fraud (i.e., acquiring a credit card with false information), stolen or lost card, counterfeit card (i.e., card copying or using a card which does not belong to the owner) and card-not-present (CNP) fraud (i.e., using credit card details to make distance purchases). This chapter focuses on CNP fraud perpetrated through online credit card transactions.

As manually processing credit card transactions is a time-consuming and resource-demanding task, credit card issuers search for high-performing and efficient algorithms that *automatically* look for anomalies in the set of incoming transactions. Data mining is a well-known and often suitable solution to big data problems involving risk such as credit risk modelling (Baesens et al., 2003a), churn prediction (Verbeke et al., 2011) and survival analysis (Backiel et al., 2014).

Nevertheless, fraud detection in general is an atypical prediction task which requires a tailored approach to address and predict future fraud. According to Definition 3.1, fraud is an uncommon, well-considered, imperceptibly concealed, time-evolving and often carefully organized crime which appears in many types and forms. Each property holds for credit card fraud:

- ***Uncommon*** The number of legitimate transactions outnumbers the number of fraudulent transactions drastically. Many credit card fraud detection studies report a fraud ratio of less than 0.5% (see Table 6.1).
- ***Well-considered*** Once fraudsters find a way to swindle, they exploit it until that type of fraud is discovered and prevention actions are taken. Extracting the right features and minimizing the opportunities of fraudsters to perpetrate fraud without being caught is an essential step in the fraud detection process.
- ***Imperceptibly concealed*** Fraudulent transactions often exhibit the same characteristics as legitimate transactions. Maes et al. (2002) formulated this as the presence of *overlapping data*. While many studies solely focus on customer profiling – *intra-account equivalence*, i.e. the extent to which the current behavior differs from previous customer behavior – models should take advantage of the knowledge sprouted from previous accounts used by fraudsters and compare this with currently legitimate customer – *inter-account equivalence*, i.e. the extent to which the customer profile differs from fraudulent profiles.
- ***Time-evolving*** An efficient fraud detection process is dynamic. There are two reasons. First, fraudsters change their way of working. Models should be fed with the most recent data to capture new types of fraud, and at the same time, should be able to prevent “existing” fraud. Second, customer changes in lifestyle might affect the spending patterns. Models that contrast a new transaction against the customer’s transaction history mark changes in spending patterns as suspiciously.
- ***Carefully organized*** Once a credit card is stolen, it is used in many fraudulent transactions. Analogously, certain merchants

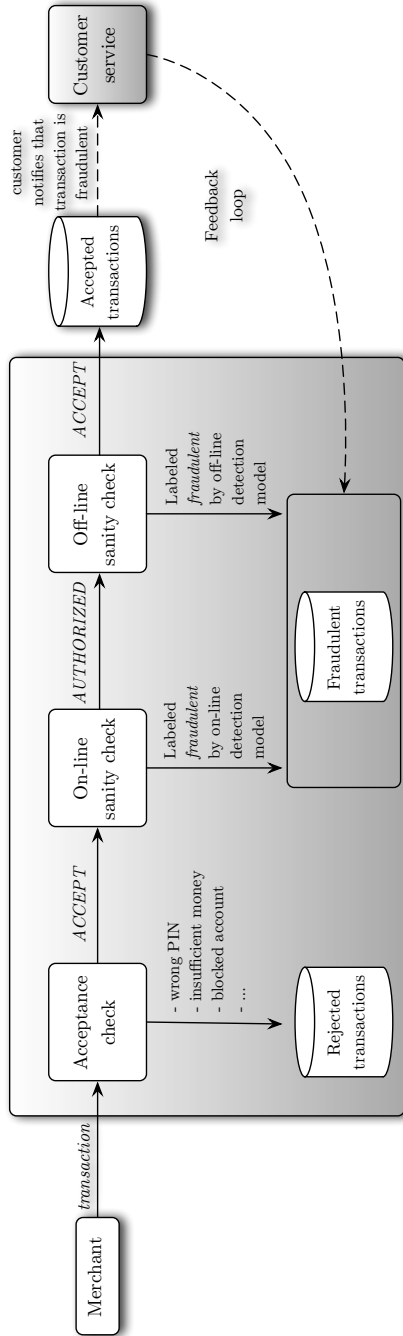


Figure 6.2: Credit Card Detection Process.

are more sensitive to fraud – merchants that perpetrate fraud by themselves or that are easily accessible by fraudsters. Efficient detection models need to exploit the relational structure among credit card holders and merchants.

Each of the aforementioned requirements need to be addressed before a detection model can efficiently work in practice. In the remainder of this chapter, a solution is formulated that systematically incorporates all of these requirements.

6.2.2 Credit Card Fraud Detection Process

The credit card detection process is summarized in Figure 6.2. The ultimate goal of such detection processes is to prevent the pursue of all transactions that do not comply with the imposed regularities. When a new transaction arrives in the system, a series of acceptance checks is performed. The transaction processing system checks for example whether the user entered the right PIN or whether the spending amount is yet sufficient. If the transaction clears the acceptance checks, it is passed on to the sanity check of the detection system. Here, the system computes the probability that the transaction is fraudulent, e.g. by applying a detection model learned from past transactions. If the probability exceeds a certain threshold, the transaction does not proceed and is aborted. The sanity check has both an on-line (i.e., in real time) and off-line module. The data set under consideration consists of all processed transactions by World-line Belgium. A transaction is fraudulent if the transaction does not pass the (a) on-line or (b) the off-line sanity check, or (c) by customer notification. While (a) is known in real-time, (b) and (c) can take up to one week.

The on-line detection process is liable to the “six-seconds rule” of decision. This means that both the acceptance check and the on-line sanity check need to be processed within six seconds. APATE operates as part of the sanity check, and can be implemented both in an on-line as an off-line environment.

#	Reference	Detection Technique	Method	Fraud Ratio
1	(Bolton and Hand, 2001)	<i>Peer group analysis and break point analysis</i>	U	N.A.
2	(Zaslavsky and Strizhak, 2006)	<i>Self-organizing maps (SOM)</i>	U	N.A.
3	(Quah and Sriganesh, 2008)	<i>Self-organizing maps (SOM)</i>	U	N.A.
4	(Weston et al., 2008)	<i>Peer group analysis</i>	U	N.A.
5	(Ghosh and Reilly, 1994)	<i>Neural networks</i>	S	?
6	(Aleskerov et al., 1997)	<i>Neural networks</i>	S	?
7	(Dorransoro et al., 1997)	<i>Neural networks</i>	S	0.6%
8	(Stolfo et al., 1997)	<i>Meta-learning</i>	S	20%
9	(Brause et al., 1999)	<i>Neural networks in combination with a rule-based association system</i>	S	0.2%
10	(Chan et al., 1999)	<i>Meta-learning</i>	S	20%
11	(Wheeler and Aitken, 2000)	<i>Case-based reasoning</i>	S	14%
12	(Maes et al., 2002)	<i>Neural networks and Bayesian belief networks</i>	S	?
13	(Syeda et al., 2002)	<i>Fuzzy neural networks</i>	S	?
14	(Shen et al., 2007)	<i>Decision trees, neural networks and logistic regression</i>	S	0.07%
15	(Srivastava et al., 2008)	<i>Hidden Markov Model</i>	S	?
16	(Whitrow et al., 2009)	<i>Transaction aggregation using a variety of models</i>	S	?
17	(Sánchez et al., 2009)	<i>Association rules</i>	S	0.3%
18	(Bhattacharyya et al., 2011)	<i>SVM, Random Forests and logistic regression</i>	S	0.5%
19	(Duman and Elikucuk, 2013)	<i>Migrating Birds Optimization and GASS algorithm (genetic algorithm)</i>	S	<0.01%
20	(Bahnsen et al., 2013)	<i>Bayes Minimum Risk</i>	S	0.025%
21	(Dal Pozzolo et al., 2014)	<i>Multiple models</i>	S	0.4%
22	(Bahnsen et al., 2014)	<i>Bayes Minimum Risk</i>	S	0.025%

Table 6.1: Overview of published papers in the credit card fraud detection domain.

6.2.3 Related Work

Although fraud detection in the credit card industry is a much-discussed topic which receives a lot of attention, the number of publicly available works is rather limited. One of the reasons is that credit card issuers protect the sharing of data sources and most algorithms are produced in-house concealing the model’s details. An overview of the literature is given in Table 6.1. In particular, credit card fraud detection techniques can be divided into two broad categories: supervised (S) and unsupervised (U) methods. Unsupervised methods solely use the customer (or transaction) characteristics to group them into small, similar clusters while maximizing the difference between the extracted clusters. If a new transaction of a certain customer is not allocated to the normal customer group, then an alarm is raised for that transaction (Bolton and Hand, 2001). Unsupervised

techniques include peer group analysis (Bolton and Hand, 2001; Weston et al., 2008) and self-organizing maps (Zaslavsky and Strizhak, 2006; Quah and Sriganesh, 2008). More studies focus on supervised techniques using evidence of past fraudulent transactions to infer the suspiciousness of future transactions. The most prevalent technique for supervised credit card fraud detection is artificial neural networks (ANN's) (Ghosh and Reilly, 1994; Aleskerov et al., 1997; Dorransoro et al., 1997; Brause et al., 1999; Maes et al., 2002; Syeda et al., 2002; Shen et al., 2007). While ANN's generally achieve a high performance, they are black box models which lack interpretability. Recently, the use of ensemble methods like Random Forests is found to perform well in credit card fraud (Whitrow et al., 2009; Bhattacharyya et al., 2011; Dal Pozzolo et al., 2014). Random Forests work especially well when there are many input features to learn from, which is often the case in network-related classification problems (Henderson et al., 2011). Other techniques for supervised learning in fraud are meta-learning (Chan et al., 1999), case-based reasoning (Wheeler and Aitken, 2000), Bayesian belief networks (Maes et al., 2002), decision trees (Shen et al., 2007), logistic regression (Shen et al., 2007; Bhattacharyya et al., 2011), hidden Markov models (Srivastava et al., 2008), association rules (Sánchez et al., 2009), support vector machines (Bhattacharyya et al., 2011), Bayes minimum risk (Bahnsen et al., 2013, 2014) and genetic algorithms (Duman and Elikucuk, 2013).

An important element in credit card fraud detection is the derivation of useful and meaningful features. Ghosh and Reilly (1994); Sánchez et al. (2009); Whitrow et al. (2009); Bhattacharyya et al. (2011); Bahnsen et al. (2013); Dal Pozzolo et al. (2014) defined three sets of features: (1) current transaction descriptors such as amount, type and timestamp of transaction, country of purchase, merchant info, etc.; (2) transaction history descriptors like number of transactions in last hour, amount spent on the transactions, typical merchant group, etc.; (3) client descriptors, like spending limit, gender, region, etc.

As no client information is available, only features from categories (1) and (2) are extracted which are enriched with network-based variables, exploiting the interrelationships between credit card holders and merchants.

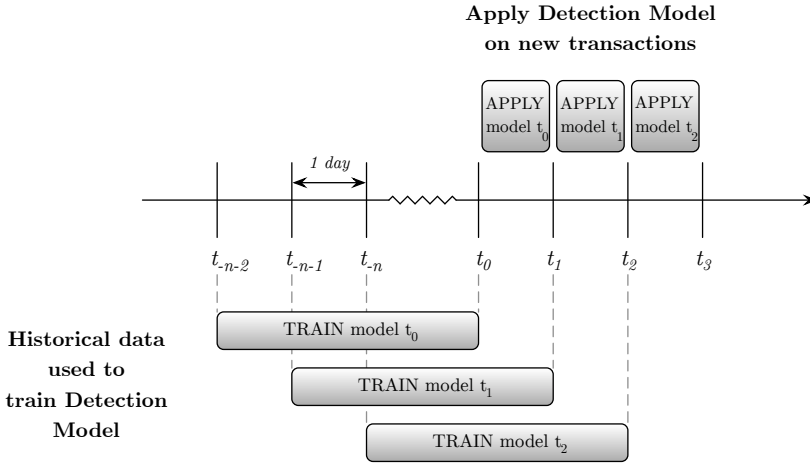


Figure 6.3: APATE’s re-estimation process of the detection models using a sliding window.

6.3 Proposed Methodology

In this section, we discuss how the APATE detection process is implemented. Note that the detection process comprises the sanity check as illustrated in Figure 6.2. Particularly, starting from a list of time stamped, labeled transactions, a model is learned to infer future fraudulent transactions. As fraud detection models should adapt dynamically to a changing environment, a sliding time window is introduced to characterize a transaction based on current (i.e., short term), and normal (i.e., medium and long term) customer’s past behavior. Both intrinsic and network-based features are derived using those three time windows. Since model estimation often cannot be executed within six seconds, the detection models are re-estimated on a daily basis at midnight the day before. Transactions made during the next day are evaluated using the model trained on data of the day before. The transaction features are extracted at real-time and fed into the model. This is depicted in Figure 6.3.

The APATE fraud detection process consists of two featurization steps:

1. ***Intrinsic feature extraction*** How does the incoming transaction differ from the previous transactions performed by that credit card holder?
2. ***Network-based feature extraction*** APATE exploits the relationships between credit card holders and merchants by means of transactions. The set of network-related features measures the exposure of each network object to fraud.

All features are summarized in Table 6.2. Remark that the transaction features are independent of the time window. In the remainder of this section, the featurization process of APATE to create intrinsic and network-based features is discussed in more details.

6.3.1 Intrinsic Feature Extraction

Traditionally, attempting to predict fraud using supervised data mining has been supported by the characterization of the purchase patterns that the customers present previous to the fraud event (Bhattacharyya et al., 2011). Most models are constructed using an aggregation of the transactions and their value. Krivko (2010) uses both the number of transactions and the monetary value of them to estimate rolling windows that are then used to train the model, Whitrow et al. (2009) use several aggregation techniques on the data and study the effects of the aggregation on the results, and Jha et al. (2012) construct a detailed data set that contains several transaction aggregations, plus information on the country in which the transaction occurred, to name a few. The literature seems to agree that there are three conditions that assist in predicting fraud: the transaction details, the time framework, and the location in which they occur.

Following this, the first set of variables that is proposed for studying this problem are a mixture of literature variables, plus some other indicators that arose during research which refers to the characteristics of the transactions themselves. Variables are construction by using inspiration from transaction analysis, including all variables that could be replicated from the studies in the literature. These variables include the number of transactions that occur in a given time framework (frequency), the amount of money spent in those transactions

Variable	Description	Summary statistics					
		ST		MT		LT	
		μ	σ	μ	σ	μ	σ
Transaction features							
Location (dummy)	Issuing region						
Belgium		0.16	0.37				
EU		0.76	0.43				
MC Category (dummy)	Category sensitivity to fraud						
Low		0.67	0.47				
Medium		0.31	0.47				
Amount	Amount of transaction	78.7	202.9				
Recency							
MC	Time passed since last transaction at the merchant	8.95	13.01	235.69	417.62	2483.8	2887.5
MC Category	at the merchant category	8.97	13.04	232.9	415.2	2652.03	2988.6
Global	across all transactions	10.31	14.39	318.3	455.8	2996.4	3011.0
Country	in the same country	9.29	13.63	242.6	420.1	2582.0	2950.7
Currency	with the same currency	9.99	14.13	292.9	446.4	2957.6	3034.2
Frequency							
MC	Total number of transactions at the merchant	0.12	0.70	0.25	1.54	0.85	5.85
MC Category	at the merchant category	0.13	0.74	0.28	1.63	0.92	6.11
Global	across all transactions	0.23	1.53	0.52	2.62	1.82	9.99
Country	in the same country	0.17	0.98	0.37	2.09	1.27	9.48
Currency	with the same currency	0.19	1.04	0.42	2.18	1.52	9.79
Monetary Value							
MC	Average amount of transactions at the merchant	5.24	120.84	9.64	158.21	30.08	558.6
MC Category	at the merchant category	6.54	139.09	13.39	198.45	47.05	783.66
Global	across all transactions	18.49	259.54	60.1	1083.7	288.63	7041.3
Country	in the same country	11.56	199.12	43.26	1002.4	227.67	6619.9
Currency	with the same currency	13.57	220.06	50.45	1068.8	261.9	7033.4
Event occurrence							
MC	First purchase? at the merchant	0.93	0.24	0.89	0.30	0.80	0.40
MC Category	at the merchant category	0.93	0.26	0.89	0.32	0.78	0.41
Global	across all transactions	0.89	0.31	0.80	0.40	0.52	0.50
Country	in the same country	0.91	0.29	0.86	0.35	0.72	0.45
Currency	with the same currency	0.90	0.30	0.83	0.38	0.60	0.49
Average Transactions							
Global	Average per time frame and level across all transactions					78.5	181.09
Merchant	at the same merchant					78.3	199.26
Exposure Score							
Transaction (TXN)	Extent to which transaction (TXN), merchant (MC) and credit card holder (CCH) are influenced by fraud given the network.	0.11e-2	0.018	0.46e-3	0.45e-2	0.40e-4	0.29e-3
Merchant (MC)		0.063	0.500	0.092	0.390	0.141	0.259
Credit card holder (CCH)		0.26e-4	0.85e-2	0.34e-4	0.36e-2	0.27e-4	0.76e-3

Table 6.2: Summary of input features on short (ST), medium (MT) and long (LT) term.

(monetary value), and the time between two subsequent transactions in a particular time period if any (recency).

The variables fit within the Recency - Frequency - Monetary Value (RFM) framework, which is widely used in marketing (Blattberg et al., 2008). There is no agreement in the literature regarding which one is an appropriate time framework to estimate these variables, ranging from hourly to averages over three months, this chapter studies both short, medium and long term: the last hour of transactions (attempting to capture cards that are heavily used and then dropped), the last day of transactions (attempting to capture specific, consumption-prone days), and the last week of transactions (attempting to capture the normal behavior of the customer). As will be shown in the experimental part (Section 6.4), only one month of transactions are available, so analysis of longer time periods were not possible. Jha et al. (2012) suggests that useful information can be extracted regarding the merchant at which the purchases occur. Data that is available, and that will be used to aggregate the merchants, concerns the merchant itself, a gross category in which the spending occurs (i.e. supermarkets, clothing stores, etc.), and an aggregated global variable with all merchants. The literature (e.g., (Bhattacharyya et al., 2011)) seems to suggest that performing the RFM analysis segmented by the currency and the country in which the transactions occurred would also bring information relevant to the study.

An additional set of binary variables was created to mark for when no purchase has occurred. These variables (FirstPurchase) mark if the transaction is the first one in that measured time frame, for each of the dimensions that are measured (see Table 6.2). This information is relevant mostly to a generalized linear model such as logistic regression, as discussed in Allison (2001). We construct 15 variables accounting for each level of aggregation and time period.

In summary, using three time periods, three types of RFM variables, and five types of transaction aggregations (single merchant, category, country, currency and global), a set of 60 ($3 \times 3 \times 5 + 3 \times 5$) variables aggregating the past transactions is developed. All variables have the following naming scheme: Level of Aggregation, RFM Type, Time Period. So for example, GlobalRecencyHour refers to the Recency (time between consecutive purchases) within one hour, when

Region	% of Transactions	% Fraudulent
Belgium	16.13%	0.05%
European Union	75.39%	0.45%
ROW	8.48%	5.36%
Total	100%	0.78%

Table 6.3: Transactions per Region and Fraud Percentage.

considering all available merchants.

The second step is to characterize the transaction itself using the location in which it occurred and the merchant info. Given the characteristics of the European credit card users, there is a strong pattern of credit card use European Union-wide, rather than in the country where the card is emitted. Transactions that occur outside the EU (mostly in the US) are rarer. Dummy variables for these three zones (EU, Belgium, and Rest-of-World, ROW) are able to capture this information. Table 6.3 shows relevant information supporting this segmentation for the data set available for this work.

We completed this part of the data set with the variables from the literature that do not fit the RFM framework. There were some variables from the literature which could not be implemented in this study, given the availability of data: some works in the literature use three months of data Bhattacharyya et al. (2011), but only one month is provided so it was impossible. Also, only online transactions of one issuer were available, so bank-related and POS related variables are not applicable, such as in Sánchez et al. (2009) and Whitrow et al. (2009). Dummy variables are included representing the currency in which the transaction occurred, categorizing them in euros, US dollars, and other currencies. We also included variables regarding the average amount of the transactions during the last week, as suggested by Bhattacharyya et al. (2011) and Whitrow et al. (2009), which were estimated both at global transaction level and at merchant level.

The last set of constructed variables deals with the categories of merchants. The data provider manifested that there were suspicions that fraudulent transactions tended to accumulate in certain categories. Using this information, the available categories (19) were segmented into three large categories using the individual categories'

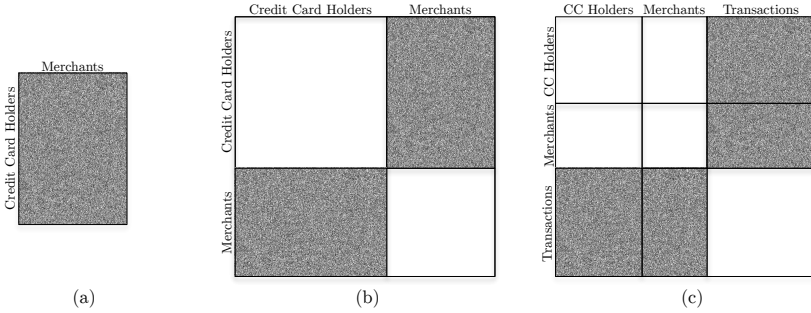


Figure 6.4: Adjacency matrix of the bipartite graph (a), transformed bipartite-to-unipartite graph (b) and transformed tripartite-to-unipartite graph (c).

fraud percentage. This leads to three dummy variables (CategoryLow, CategoryMid, and CategoryHigh) capturing this assumption.

After constructing the data set along the intrinsic (transaction related) variables, the information is complemented by exploring a novel approach of network analysis, as described in the next subsection.

6.3.2 Network Feature Extraction

Network definition

A graph that represents heterogeneous node types, is a multipartite graph. In particular, the credit card fraud network in this work is represented as a *bipartite* graph $\mathcal{G}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E})$, containing two node types – i.e. credit card holders and merchants – and satisfies the following property:

$$e(v_1, v_2) \text{ subset of } \mathcal{V}_1 \times \mathcal{V}_2 \quad (6.1)$$

with $v_1 \in \mathcal{V}_1$ the set of credit card holder nodes, and $v_2 \in \mathcal{V}_2$ the set of merchant nodes. Property 6.1 enforces that a transaction can only exist between different node types, i.e. credit card holders and merchants. A toy example of the credit card network is shown in Figure 6.1.

The corresponding adjacency matrix $\mathbf{A}_{\mathbf{c} \times \mathbf{m}} = (a_{i,j})$ of a bipartite graph is a matrix of size $c \times m$ with c and m the total number

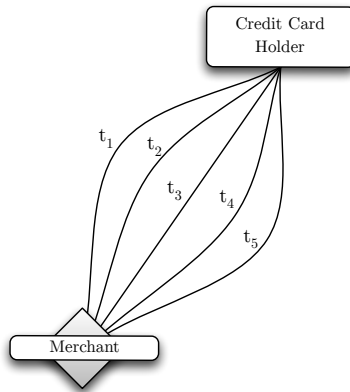


Figure 6.5: Example of a multi-edge subnetwork. The credit card holder made several transactions at the same merchant.

of credit card holder and merchant nodes respectively. The weight matrix $\mathbf{W}_{\mathbf{c} \times \mathbf{m}} = (w_{i,j})$ represents the weighted graph as a matrix.

In order to address the dynamic character of fraud, we integrate time into the network such that the edges express the *recency* of the transaction. Inspired by the half-life decay of atoms, we exponentially decay the intensity of a relationship in time, where:

$$\begin{cases} w_{i,j} = e^{-\gamma h} & \text{if a relationship exists between node } i \text{ and } j \\ w_{i,j} = 0 & \text{otherwise} \end{cases}$$

with γ the decay constant and h the time passed since the transaction pursued, measured according to the interval being studied (minutes for short term, hours for medium term and days for long term). The decay constant γ is chosen such that the edge weight is close to zero after one month (long-term: $\gamma = 0.0001$), one week (medium-term: $\gamma = 0.004$) and one day (short-term: $\gamma = 0.03$) respectively. A high weight represents a recent transaction. An example of a bipartite adjacency matrix is shown in Figure 6.4a.

We note that multiple transactions can occur between the same credit card holder and merchant. In that case, we say that the graph has *multi-edges* between two nodes. This is depicted in Figure 6.5. As adjacency matrices only represent the intensity between two nodes as

if they were connected by one edge, multi-edge information is often aggregated (e.g., sum, max, mean, etc.).

Network fraud propagation

Given a credit card network, how can we use the fraud label of the edges – i.e. the transactions – to infer a score for each network object? That is, we want to infer a score for each credit card holder, merchant and transaction. The derived score expresses the extent to which the network object is exposed to fraud, and is therefore called the *exposure* or *suspiciousness score*.

So far, all studies start from a limited set of labeled *nodes*, and infer a ranking for the remaining nodes. This chapter proposes a propagation algorithm that start from a limited set of labeled *edges* in order to label the remaining network objects.

Remark that Chapter 3 (see Section 3.3.3) introduced GOTCHA!’s fraud propagation algorithm for social security fraud to analyze bipartite graphs. GOTCHA!’s propagation algorithm is an iterative fraud scoring algorithm that is designed such that it scores two node types (cfr. bipartite graphs) based on the label of one node type. Assume that a graph consists of c type-one nodes and m type-two nodes. After k iterations, the vector containing the exposure scores of each node equals:

$$\vec{\xi}_k = \alpha \cdot \mathbf{Q}_{norm} \cdot \vec{\xi}_{k-1} + (1 - \alpha) \cdot \vec{z}_{norm} \quad (6.2)$$

with $\vec{\xi}_k$ the $(c + m)$ -vector containing the exposure scores of each node after k iterations, $\vec{\xi}_0$ a random vector with values between $[0, 1]$, $(1 - \alpha)$ the restart probability (according to Page et al. (1998), we choose $\alpha = 0.85$), \mathbf{Q}_{norm} the column-normalized weight matrix of size $((c + m) \times (c + m))$, and \vec{z}_{norm} the normalized degree-adapted starting vector of size $(c + m)$. Both \mathbf{Q}_{norm} and \vec{z}_{norm} are weighted in time to address the dynamic characteristic of fraud. The bipartite adjacency matrix as illustrated in Figure 6.4a is transformed into a symmetric, unipartite matrix with $q_{i,j} = 0$ if node i and j are both credit card holders or merchants (see Figure 6.4b)

Equation 6.2 starts from a limited set of labeled *nodes* to infer a score for the remaining *nodes*. However, in credit card fraud, we

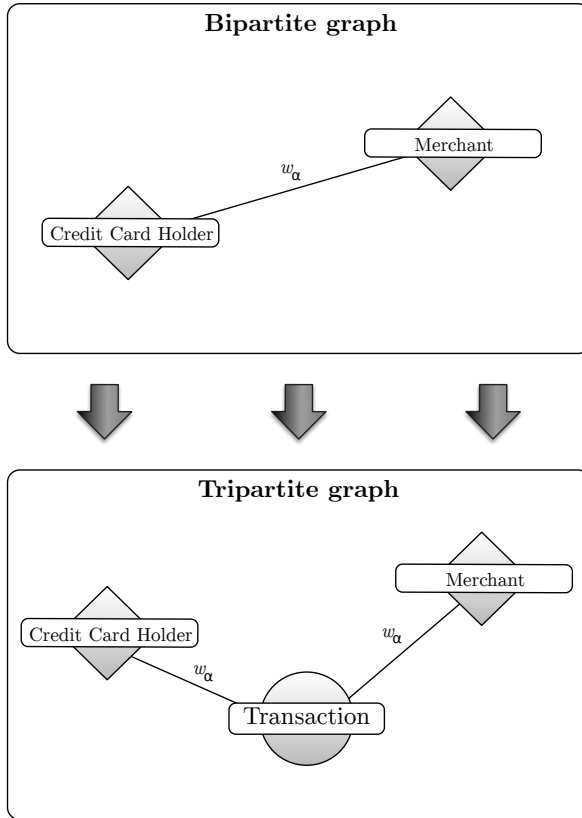


Figure 6.6: Transformation process from a bipartite to a tripartite graph by representing the edges as a separate node in the network.

require to start from a limited set of labeled *edges* to derive a score for both the *edges* and *nodes*. Therefore, APATE's network propagation algorithm adapts Equation 6.2 by making two changes: (1) \mathbf{Q}_{norm} is transformed into a tripartite graph including transactions as a node in the network; (2) \vec{z}_{norm} is a time-dependent normalized vector indicating the fraudulent transactions. These adaptations are discussed next.

(1) *Edge-to-node transformation*

In order to be able to propagate influence from edges, the edges are included as a separate entity in the network. That is, the edges are transformed into nodes and create a tripartite graph $\mathcal{G}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \mathcal{E})$ with $\mathcal{E} \subseteq (\mathcal{V}_1 \times \mathcal{V}_3) \cup (\mathcal{V}_2 \times \mathcal{V}_3)$, such that the following property holds:

$$\begin{aligned} \forall v_3 \in \mathcal{V}_3 : \exists! v_1 \in \mathcal{V}_1 \mid e(v_1, v_3) \in \mathcal{E} \\ \& \exists! v_2 \in \mathcal{V}_2 \mid e(v_2, v_3) \in \mathcal{E}. \end{aligned} \quad (6.3)$$

with $v_1 \in \mathcal{V}_1$ the set of credit card holder nodes, $v_2 \in \mathcal{V}_2$ the set of merchant nodes and $v_3 \in \mathcal{V}_3$ the set of transaction nodes. Property 6.3 enforces that credit card holder nodes and merchant nodes can only be connected to transaction nodes. An example transformation is illustrated in Figure 6.6. We note that the edge weight in the tripartite graph between the transaction and both the credit card holder and merchant is equal to the edge weight between the credit card holder and the merchant in the original bipartite graph $\mathbf{W}_{c \times m}$. Let's say that c , m and t are the total number of credit card holder nodes, merchant nodes and transaction nodes respectively, then the weighted matrix $\mathbf{M}_{(c+m) \times t}$ is the mathematical representation of the tripartite graph which is exponentially decayed over time, and

$$\begin{cases} w_{i,j} = e^{-\gamma h} & \text{if a relationship exists between node } i \text{ and } j \\ w_{i,j} = 0 & \text{otherwise} \end{cases}$$

Adding the transactions as separate nodes in the graph, enables us to easily integrate multi-edges from the bipartite graph into the tripartite graph. As each transaction edge in the bipartite graph is transformed into a transaction node in the tripartite graph, the weighted matrix creates for each transaction node a separate column. There is no need to aggregate multi-edge information.

As Equation 6.2 requires a symmetric matrix, the tripartite graph is transformed into a symmetric unipartite graph, as illustrated in Figure 6.4c. Mathematically,

$$\mathbf{Q}^{\text{tri}} = \begin{pmatrix} 0_{(c+m) \times (c+m)} & \mathbf{M} \\ \mathbf{M}' & 0_{t \times t} \end{pmatrix} \quad (6.4)$$

Matrix \mathbf{Q}^{tri} is a matrix with $c + m + t$ rows and columns. After normalizing the columns such that each column sums up to 1, the resulting matrix is $\mathbf{Q}_{\text{norm}}^{\text{tri}}$.

(2) *Starting vector*

The starting vector is originally created to personalize the ranking of web pages by guiding the algorithm with the user’s interests Page et al. (1998). Rather than initializing the starting vector as a uniformly distributed vector, the starting vector can be used to emphasize the influence of certain nodes on the final ranking. The same reasoning holds for fraud. As we are not interested in any influence to propagate through the network, but only in fraudulent influence, we guide the algorithm by specifying the confirmed fraudulent transactions using the starting vector. That is, the starting vector \mathbf{z}^{tri} of size $(c + m + t)$ equals:

$$\begin{cases} z_i^{\text{tri}} = e^{-\beta h} & \text{if node } i \text{ is a fraudulent transaction} \\ z_i^{\text{tri}} = 0 & \text{otherwise} \end{cases}$$

with β the decay constant, and h the time passed since the transaction is labeled as fraudulent. Dependent on the time window of analysis, the fraudulent influence is exponentially decayed on long ($\beta = 0.0001$), medium ($\beta = 0.004$) or short ($\beta = 0.03$) term. All credit card holder and merchant nodes have a zero weight for the starting vector. Remark that a higher weight is assigned to fraudulent transactions that occurred more recently.

The starting vector is normalized to $\mathbf{z}_{\text{norm}}^{\text{tri}}$, summing up to 1.

Using the previous modification to the bipartite propagation algorithm as stated in Equation 6.2, we derive APATE’s propagation

algorithm for edge and node labeling, where:

$$\vec{\xi}_k = \alpha \cdot \mathbf{Q}_{norm}^{tri} \cdot \vec{\xi}_{k-1} + (1 - \alpha) \cdot \vec{z}_{norm}^{tri} \quad (6.5)$$

The resulting score $\vec{\xi}_k$ is computed using the power-iteration method, iterating until convergence. Convergence is reached after a maximum number of iteration steps or when the change in the scores is marginal.

Feature extraction

As a long-, medium- and short-term time window is used in the analysis, matrix \mathbf{Q}_{norm}^{tri} and \vec{z}_{norm}^{tri} in Equation 6.5 are computed with different α values ($\alpha = 0.0001, 0.004, 0.03$) to infer an exposure score of each node and edge using information up until one month, week and day respectively. For example, the long-term exposure score indicates the extent to which the transaction (or merchant, or credit card holder) is sensitive to fraud during the last month. In general, the higher the exposure score of a network object, the more the node or edge is surrounded by fraud in its neighborhood.

For each new incoming transaction, the following features are computed: (a) credit card holder exposure score (CCHScore), (b) merchants exposure score (MCSScore) and (c) transaction (TXScore) exposure score on long (LT), medium (MT) and short (ST) term. We re-estimate the exposure scores for every network object each day at midnight in order to extract the evidential features for transactions that occur the next day.

The credit card holder and merchant exposure score are derived from Equation 6.5. If the credit card holder or merchant did not yet appear in the network – i.e., he/she did not perform any transaction during the time period of analysis – a score of zero is assigned to that node, as they are not yet exposed to fraudulent influences.

The transaction exposure score combines the influence of the associated credit card holder and merchant. If a transaction already occurred between the credit card holder and the merchant, the exposure score as calculated in Equation 6.5 is used. If multiple transactions occurred between the same credit card holder and merchant,

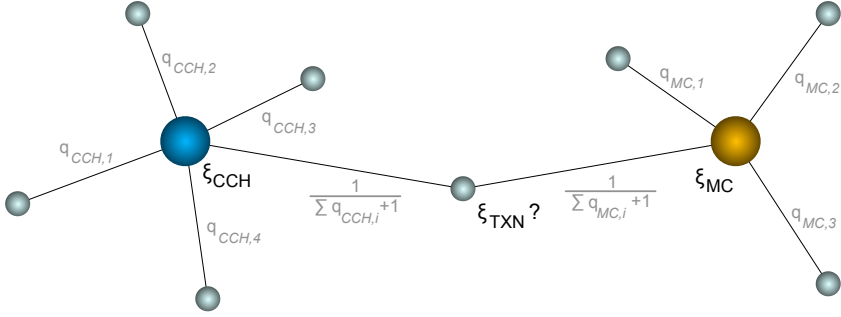


Figure 6.7: Local updating process when new transaction appears in the network.

we use the score assigned to the most recent transaction. When a transaction did not yet happen between a certain credit card holder and merchant, the exposure score of that transaction is computed by using the exposure scores of its direct neighborhood. Therefore, we say that the exposure scores are **locally updated** in the network, where:

$$\text{TXN}_{i,k,\text{score}} = \frac{1}{\sum_{j=1}^n w_{i,j} + 1} \text{CCH}_{i,\text{score}} + \frac{1}{\sum_{j=1}^m w_{k,j} + 1} \text{MC}_{k,\text{score}} \quad (6.6)$$

with $\text{TXN}_{i,k,\text{score}}$ the exposure score of a transaction between credit card holder i and merchant k , $\text{CCH}_{i,\text{score}}$ the exposure score of credit card holder i , $\text{MC}_{k,\text{score}}$ the exposure score of merchant k , $w_{x,y}$ the link weight between node x and y , and n and m the total number of links from credit card holder i and merchant k respectively. The local updating algorithm redivides the fraudulent influences. Instead of propagating the exposure score of the credit card holder/merchant only among the past transactions, the exposure score is now partly absorbed by the newly added transaction. We note that the edge weight of the new transaction is set to 1, as it represents a current relationship. This is depicted in Figure 6.7.

The network feature extraction step results in 9 features for each transaction: long-, medium- and short-term exposure scores for the transaction, associated credit card holder and merchant.

In the following section, analytic models are estimated using three sets of variables: intrinsic (18), network-based (9) and demographics (5); and measure the capabilities they have for predicting fraud. In all cases, the objective is to estimate the probability of fraud given the variables available, that is:

$$P(Y = \textit{fraud} | X_{\textit{Intrinsic}}, X_{\textit{Network}}, X_{\textit{Demographics}}) \quad (6.7)$$

6.4 Results

To test the proposed approach, a unique data set of approximately 3.3M transactions from a large Belgian credit card issuer has been used. The data consists of a supervised data set with all the information related to transactions occurring during five consecutive weeks, plus a fraud or no fraud mark added for each transaction by the company after suspicious transactions were investigated (after two weeks at most). The data set is highly imbalanced, with only 48 000 frauds among the transactions (< %1).

In this section, the following three questions are answered: What is the best model for the approach? How can the model be applied in a real life-situation? And finally, what is the added value of using network variables for this problem? For all questions, an out-of-time test set is created consisting of all transactions that occur in the last week (approximately 500k), while the first two weeks will be used as the data pool for creating the RFM and network variables for the following two weeks of data (the training set).

During data cleansing and pre-processing, all transactions that were rejected due to normal banking reasons (wrong PIN, input errors, and other non-purchase related reasons) were eliminated from the data set. These transactions account for 15% of all transactions. Additionally, all transactions over 5000 EUR were also dropped from the data set, to avoid distortions in the set. These transactions are clear outliers: they consist of less than 1% of all transactions (none of them fraudulent) and they were almost 25 standard deviations from regular transactions, as shown in Table 6.2, so eliminating them leads to more stable models. The final training set consists of 2.2M transactions, and the final test set consists of 500k transactions. For each

case, the variables described in Section 6.3 are calculated, accounting for 78 different variables, 9 which are network-based, 60 RFM variables, and the remaining variables being the non-RFM literature variables, or demographic and location-related.

6.4.1 Prediction Results

According to the findings of related research (see Section 6.2.3), three models will be benchmarked to each other: logistic regression, the standard general linear model for classification used in many banking related activities, which is the less powerful of the group in terms of predictive capabilities, but is very simple to understand; a feed-forward, one hidden layer, neural network, one of the most powerful non-linear models, but that is considered a black box; and a Random Forest, a very powerful ensemble of decision trees which has brought very good results in many publications dealing with multiple applications.

To tackle the imbalance problem, standard case weighting for neural networks and logistic regression is applied. For Random Forests, the sub-sampling capabilities of Random Forests are used, with each tree constructed using all fraudulent transactions and a randomly selected subset of the non-fraudulent ones such that they account for two times the number of fraudulent ones, as explained in Chen et al. (2004a). The Random Forests model is trained using 500 trees, which gives non-fraudulent cases an *a priori* chance of being selected similar to the one of simple random sampling. For parameter tuning, in the case of neural networks, 20% of the training data set was reserved for tuning the parameters, selecting the best combination of epochs and number of neurons over the grid given by $(Neurons, Epochs) \in [16, 156] \times [100, 1000]$, with the epochs increased in increments of 50, and the neurons in increments of one.

Model	AUC	Accuracy
Logistic Regression	0.972	95.92%
Neural Networks	0.974	93.84%
Random Forests	0.986	98.77%

Table 6.4: Comparison of models.

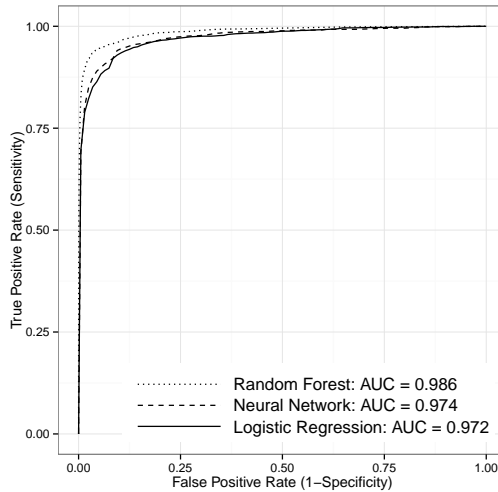


Figure 6.8: ROC Curve for Different Models.

The results, in Table 6.4, show a very high accuracy and Area-Under-the-ROC-Curve (AUC) values. The models are almost perfect, correctly predicting 98.7% of cases in the case of the R Random Forests (the highest value), and with an AUC of 0.987. The relatively lower accuracy in the other two models is caused by a higher fraud detection rate when contrasted with false positives: the models are good at detecting frauds, but that comes at a cost of some extra non-fraudulent transactions being detected as fraudulent, which does not occur with Random Forests. This hits accuracy given the high imbalance of the data set. The very high AUC obtained can be seen in Figure 6.8.

To make a fairer comparison, possibly closer to a real application of the model, we will study the case when at most a 1% false positives are acceptable. The rationale behind this is that there is a reputational cost whenever a false positive occurs, given that users of the credit card will get a rejection on a non-fraudulent transaction, with all the consequences and annoyances that such an action brings. Table 6.5 shows the obtained results.

The results continue to be very good, but now the effects of the highly imbalanced problem are apparent. Random Forests is the best

Model	Cut-Off	Balanced Acc.	Specificity
Logistic Regression	0.85	87.4%	75.7%
Neural Networks	0.99	87.9%	76.8%
Random Forests	0.53	93.2%	87.4%

Table 6.5: Accuracy and AUC (test set) at 1% Maximum False Positive Rate.

model overall, with an 87.4% accuracy in the positive (fraudulent) cases, and a balanced accuracy of 93.2%. It is followed by neural networks, with a 76.8% specificity. The results hint at a highly non-linear problem, since there is a clear advantage when using non-linear models, which can be as large as the 12% increase in specificity when comparing Random Forests with logistic regression. The difference between neural networks and Random Forests also suggests that the problem is not only highly non-linear, but that it is necessary to apply an ensemble model that searches for patterns in the sub-spaces that arise when applying a Random Forest. In any case, the results are very good. A user could use the model and detect close to 90% of all fraudulent transactions, flagging incorrectly only 1% of non-fraudulent ones.

6.4.2 Variable Importance and Network Variable Impact

Type	AUC	Accuracy
Only RFM	0.953	97.83%
Literature	0.955	97.87%
All Variables - First transaction	0.971	99.46%
Only Social Networks	0.920	94.37%
All variables	0.986	98.77%

Table 6.6: AUC for Different Subsets of Variables.

The final question that needs to be answered is which variables are more important, and try to measure their effect in the model overall. There are three main sets of variables in the problem: The RFM and demographic variables, the variables that are suggested in the literature that extend the RFM methodology, and the network

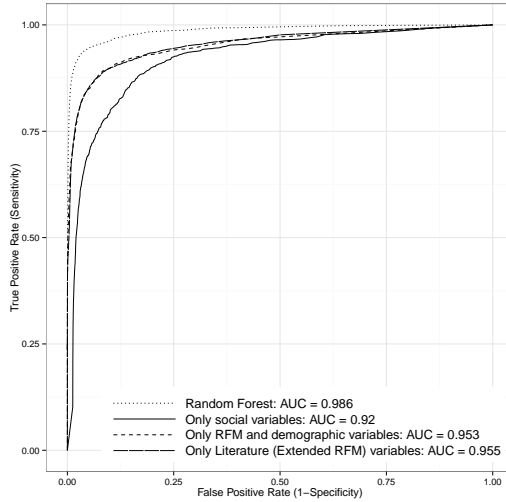


Figure 6.9: ROC Curve for Different Subsets of Variables.

variables. In order to contrast these sets three additional Random Forests are estimated – since they give the best results –, one for each subset of variables. The results of these models can be seen in Table 6.6.

It can be seen that only using the 9 network variables available the model reaches an AUC of 0.920. A model with only the RFM and demographic variables reaches an AUC 0.953, slightly higher. The inclusion of currency and country variables, together with the transaction averages (from the literature) make the AUC increase only slightly to 0.955. From these results we can conclude that the RFM variables are a very good set of variables to predict fraud, permitting to reach a very high AUC measure. The inclusion of the extended literature variables increase only slightly the AUC from a pure RFM approach, which might be caused due to regional behavior described in the data set we have available: variables representing currency and country do not present a strictly different behavior in Europe – with a unified currency, small travel distances and an integrated market, which might be even stronger when considering online sales – than what it might occur on different regions, such as North America or Australia. We can conclude that the inclusion of transaction averages,

currency, and country variables has a minor, albeit positive, impact on the description of fraud for our data set.

The inclusion of social network variables in combination with all the RFM variables has a very strong impact on the prediction results, reaching an AUC of 0.987. The main conclusion that can be derived from this result is that, considering that the social network variables have a very small correlation with respect to the other sets (the largest is 0.1), the information that these variables bring allows increasing the capabilities of the data set, interacting multidimensionally with the other two sets of variables, which translates into an increase of 5% in the AUC of the model. The ROC curves of the three different models (Figure 6.9) show that the models perform similarly in terms of the separation of false positives and false negatives, but the full model has less false positives in the early stages of the model, and that gain comes from the combination of the data sets.

When dealing with fraud, it is common to see several transactions that occur in a very short period of time, with a very high accumulated monetary value. As such, detecting the first transaction that is fraudulent is an interesting problem. Table 6.6 displays the AUC of first transactions only (the ones with `GlobalFrequencyHour` equal to zero). It can be seen that the AUC, although lower, is still very high, which suggests that the purchasing patterns that precede fraud in the long term are the most relevant for predicting it, or, conversely, that it is the contrast between current and past behaviors that allow to correctly estimate fraud, and this is correctly captured by the variables in the model.

The exact relevance of the variable can also be extracted from the Random Forests model, and sheds light on the multidimensional increase in predictive capabilities of the model. Figure 6.10 shows the relative importance of each variable according to the Random Forests. It is interesting to note that the top two (with very similar importance) are one for each set, and are both related to the merchant at which the purchase occurs: `AvgAmountMerchantWeek` corresponds to the average monetary value per week before the current transaction at the merchant, so it shows the normal behavior on any given week, whereas `LT.TXScore` shows the long-term behavior of the network associated with the transaction itself, representing the normal, long-term, rela-

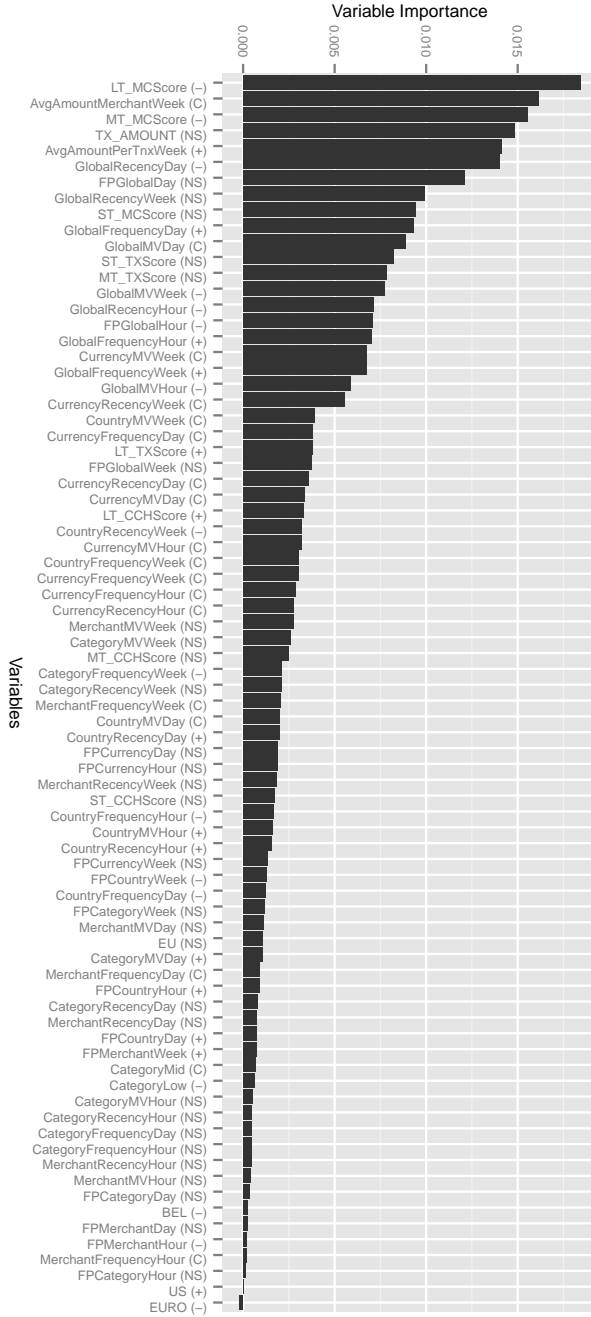


Figure 6.10: Importance of each variable for the Random Forests built using all available variables. In parentheses after each variable is the relevant information from the logistic regression output: A + or - sign of significant coefficients, NS when the variable was not significant, and C when it was highly correlated (greater than 0.9) with another variable in the data set.

tion between merchants and the user of the credit card, weighting in the expected patterns of both fraud and non-fraud given the structure of the network. The next set of variables are again some literature and RFM variables mixed with network variables, but now referring to the medium-term (day) purchases, followed by the short-term network scores for both the merchant and the transaction. The variables representing transactions during the last hour seem to be of lower importance, and the currency variables close the list, which suggest, as shown before, that the purchase pattern in Europe is marked by the euro, so effects on currency that were present in previous works in the literature are annulled. It follows that it is from these relationships between the social network variables and the purchase patterns that the learning process is able to extract a significant amount of information that allows for a very high accuracy and AUC. As was expected, the first purchase variables are of limited importance in the Random Forests, but they can be significant in the logistic regression, considering the fact that the information of those variables was included in a mixture of information from other variables which cannot be recovered easily in generalized linear models.

Regarding the signs and significance, most short-term variables are non-significant in the logistic model, which suggests that the hourly behavior requires a deeper multivariate analysis that Random Forests delivers. All currency-related, and many country-related variables are highly correlated with other variables in the data set which suggest that the purchasing patterns of the data set are highly localized. The signs of the significant variables show that the hourly behavior tends to have a positive sign, which increases the odds of fraud, showing that when there are short-term increases in purchasing there is a higher risk of fraud. Something similar happens with the global variables: a higher global frequency is related to a higher fraud probability, but a higher monetary value is related to lower odds of fraud. All long-term social network variables are relevant, with varying signs: the long-term merchant score has a negative sign, showing that there are less risky merchants when dealing with fraud, but the transaction and customer long-term score have positive signs, which suggests that there are riskier customers, more prone to be subject to fraudulent activities.

6.5 Conclusions

In summary, this chapter tackles credit card transaction fraud by proposing a novel, automated and real-time approach APATE (short for: Anomaly Prevention using Advanced Transaction Exploration). For each new incoming transaction, APATE decides whether the transaction might hint towards fraud and whether or not it should be pursued. A major component of APATE is the feature extraction part, where both intrinsic and network-based attributes are combined. Our approach uses the RFM framework (Recency - Frequency - Monetary Value) complemented with demographic information of the transaction to define intrinsic features. As opposed to many previous studies, the detection models are enriched with network variables. The credit card fraud network consists of a network where credit card holders are connected to the merchants through the transactions they make. In particular, this chapter discusses a new technique for fraud propagation through the network starting from a limited set of labeled edges (i.e., fraudulent transactions) and inferring a score for all the network components (i.e., credit card holders, merchants and transactions).

APATE is tested on a company data set with more than three million transactions, by estimating a logistic regression, a neural network and a Random Forests model. Results show that the proposed approach leads to a very high AUC score and accuracy, especially for Random Forests. Even after adjusting the model to only allow 1% false positives, a high specificity is achieved, meaning that our models efficiently identify fraudulent transactions. Although each set of features separately results in a good model performance, the best results are reached when both intrinsic and network variables are combined, which suggest that there is a multidimensional component that is inherent of the combinations of the RFM and network approaches, potentially capturing both a short-term change in behavior – contrasting the short-term purchase pattern with the normal ones, either daily or weekly one – and a long-term structure of the transactions, which arises from analyzing the different networks that can be inferred from the data. Finally, this chapter shows that APATE is not only able to find almost all fraudulent transactions, but also accurately pick out the first transaction in a series of fraudulent transactions, which is an important requirement in curtailing credit card transaction fraud.

Chapter 7

Conclusions

*“One never notices what has been done;
one can only see what remains to be done.”*

— Marie Curie, 1894

In this dissertation a set of new approaches are developed on how to use network analysis in a fraud detection context. All approaches are tested on real-life data sets obtained from (a) the Belgian Social Security Institution which aims to detect and prevent corporate tax evasion, and (b) Worldline Belgium, a credit card issuer that tries to minimize the losses due to unlawful use of credit cards by fraudsters. All approaches are developed from the point of view of the application, whilst simultaneously being as generic as possible. Hence, this dissertation serves as an intertwinement of theory and practice. This last chapter consists of two parts. In the first part, the main findings and conclusions of this dissertation are recapitulated. The second part elaborates further on future research ideas.

7.1 Conclusions

This dissertation is situated in the research domain of data science. The introductory chapter elaborates further on data science, its requirements and its applications. Data science encompasses every theory, strategy and action undertaken that use data, ranging from data definition, storage and collection to the interpretation,

implementation and evaluation of knowledge derived from data. The main objective is to develop automated detection algorithms that are capable of processing massive amounts of data in a limited time span which generate a highly accurate, meaningful and precise output. These algorithms help fraud fighters in the investigation process. This chapter also zooms in on the motives of people why they commit fraud. The fraud triangle helps to identify three drivers: pressure, opportunity and rationalization. While the first driver focuses on the incentives (e.g., money, prestige, greed, etc.) of fraud, the opportunity refers to the possibilities created by the system (e.g., a company, society, environment...) or by the fraudsters themselves. Rationalization specifies the fact that fraudsters resign themselves to their crimes. Many types of fraud exist. This chapter provides a non-exhaustive list of different fraud categories. Especially, credit card fraud and tax evasion are highlighted. All approaches in this dissertation are applied on credit card fraud and tax evasion.

The second chapter serves as a formal basis for network notation and representation which are used in the next chapters. The potential strength of state-of-the-art network analysis is discussed, together with a brief introduction to applications of network analysis in a fraud detection setting. Furthermore, this chapter elaborates on how a network can be represented in a mathematically interesting manner in order to derive useful statistics and meaningful features from the network in a scalable way. In terms of scalability, egonets are presented. An egonet is the one-hop induced neighborhood centered around a node of interest. We argue that analyzing each node's egonet, can result in a powerful set of features. In addition, we contrast the various options to decide upon the weight of edges which are able to quantify the intensity of relationships. Homophily is explained, a concept borrowed from sociology, which states that people have the tendency to connect to other people that are similar to themselves (e.g., interests, hobbies, propensity to commit fraud, etc.). In a network that exhibits statistically significant signs of homophily, nearby social neighbors are alike. In this chapter, homophily is mainly approached from a fraud perspective, so to serve as a primary indicator whether a fraud detection model might

benefit from network analysis. The second chapter is concluded by entering into the featurization process. The featurization process defines how unstructured network information can be mapped into a set of structured features. Neighborhood metrics, centrality metrics and collective inference algorithms are presented.

In Chapter 3, a new fraud detection approach GOTCHA! is introduced. The approach is tested on data from the Belgian Social Security Institution which is responsible for the collection and distribution of social contributions. Fraud is defined as those companies that intentionally go bankrupt in order to not pay their taxes. The aim hereby is to improve the performance of traditional classification techniques for social security fraud by including information from a *time-weighted, bipartite* network. Fraud is dynamic and evolves over time. The network exhibits time in the edge weight. The bipartite structure is imposed by domain requirements. This means that network includes two node types (here: companies and resources). Starting from a limited set of fraudulent companies, fraudulent influences of one node type are spread through the network in a viral like manner, so to infer an initial exposure score for both node types, i.e., the unlabeled companies and resources. The propagation algorithm inherits concepts from the Personalized PageRank approach as proposed by Page et al. (1998), and is extended by making the following domain-dependent adjustments: (1) propagation for bipartite graphs (i.e., scoring both companies and resources), (2) emphasizing fraud, (3) dynamical procedure: use of time-dependent weight to represent relationships between companies and resources, and to weight the impact of fraud, (4) degree-independent propagation. The time-dependent weight allows both to anticipate and forgive the riskiness of the resources. For each company, we aggregate the properties of the direct neighborhood, and combine them with intrinsic features.

The Social Security Institution and other similar fraud applications benefit from the developed approach in multiple ways: (1) *Guided search for fraud*. Instead of randomly investigating companies, GOTCHA! produces an accurate list of companies that are worthwhile

to investigate by experts. Experiments show that GOTCHA! exploits essential information from the network predicting future fraud more efficiently. GOTCHA! is compared to three other baselines. The first one is an intrinsic-only baseline and uses only intrinsic features. The second one is a unipartite baseline, linking the companies directly to each other and aggregating resource information in the link weight. Hence, the network is not time-weighted and only contains one node type. The third one extends the network representation to bipartite graphs, as often imposed in a fraud setting, but does not include time in the link weights. Results show that GOTCHA! significantly produces more accurate results than the baselines in terms of AUC score. We find that network models achieve a higher precision, although recall is approximately the same. Hence, network-driven models reduce the set of high-risk companies passed on to the experts for further screening. (2) *Faster fraud detection.* The predictability of short-term models is surprising. Short-term models are not only capable to accurately predict which companies will commit fraud in the near future, but also identify companies that perpetrate fraud many years later. This results in a higher overall precision compared to medium- and long-term models, favoring the short-term models in the fraud detection process.

A similar network model is used in practice by the Belgian Social Security Institution to guide fraud experts and inspections.

While the focus of Chapter 3 was to identify individual fraudsters, the objective of Chapter 4 is to find undetected groups in the network. Rather than relying on confirmed fraud, GOTCHA'!!! aims to learn from the structure of cliques, and the local properties of the clique members. Again, starting from a bipartite network, the graph is split up in cliques. A clique is a fully connected subgraph where each node is connected to every other node. However, the bipartite structure where companies are uniquely connected to their resources does not allow to find such cliques. Hence, the definition of a clique is relaxed, such that we aim to find subgraphs where each company node is connected to each resource node, and vice versa. Every clique is assigned a score in terms of the sensitivity of that clique to fraud and bankruptcy based on the computed

exposure scores. Clique-based features are enriched with intrinsic and relational features for each clique member. Results indicate that the combination of clique-based, relational and intrinsic features achieves the best performance. It is shown that bankruptcy is an important indicator and often comes along with fraud. Compared to GOTCHA!, we find that the same intrinsic features are highlighted as important predictors. In particular, GOTCHA'!! is able to uncover 22% fraud cases, which is very high considering the extremely skewed class distribution ($< 0.2\%$). Remark that these results differ from the results achieved in GOTCHA! for the following reasons: (1) no direct features are computed for GOTCHA'!!, (2) GOTCHA! does not include bankruptcy-based features, (3) GOTCHA! comprises a more thorough post-analysis taking into account suspicious bankruptcies and frauds that occur even after the adopted analysis period, and (4) the results in GOTCHA! are evaluated based on an out-of-time validation, whilst the results of GOTCHA'!! are evaluated on the same time stamp.

The usefulness of this algorithm is its capability to uncover undetected groups in the network, and finally curtail growth of such fraudulent subgraphs. Furthermore, the fundamentals of this approach are used for CAW (Check-in @ Work) registrations of the Belgian Social Security Institution, a new check-in system for employees at construction sites, operational from April 2015.

Chapter 5 elaborates on how to improve classification output given the limited budget b of inspectors. The budget b corresponds to the number of companies that can be inspected by fraud experts, and is often extremely low. The research objective is formulated as follows: If we invest an amount k of total budget b to ask inspectors about the true label of a set of companies selected based on some selection criterion, and we use these labels to re-learn a new model, will we achieve more precise results using the new model than by using the complete budget b to inspect the initial results without re-learning? Network models often rely on collective inference algorithms, where the label of one node is said to depend on the label of the other node. A change in the label of one node might have an impact on the label of its neighbors, which might have an

impact on their neighbors in turn, and so on. Active inference is a subdomain of active learning where a network-based algorithm iteratively learns the label of a set of unknown nodes in the network in order to improve the classification performance. Although the domain requirements are rather strict, Random Forests benefit the most from active inference, achieving an increase in precision up to 15%. We investigated different probing (or selection) strategies to select the most informative nodes in the network and evaluate (1) expert-based and (2) committee-based strategies which are often preferred in order to obtain an unbiased set of companies for investigation. Results show that probing strategies on their own are able to identify those companies with the most uncertainty, resulting in a total precision of up to 45%.

In Chapter 6, an approach is developed to tackle credit card fraud. Although initially the domain requirements of credit card fraud seem similar to the domain requirements imposed by social security fraud (i.e., both start from a bipartite graph of credit card holders/merchants or companies/resources respectively), there is one main difference. In social security fraud, companies were attributed to fraud. These are the nodes in the corresponding network. In credit card fraud, the transactions or edges between merchants and credit card holders are labeled. In this chapter, we start from a limited set of labeled edge (i.e., fraudulent transactions) to infer an exposure score for all the network components (i.e., the credit card holders, merchants and new incoming transactions). The approach is tested on a data set from a credit card issuer with more than three million transactions. The best results are reached when both intrinsic and network variables are combined, which suggest that there is a multidimensional component that is inherent to the combinations of the RFM and network approaches, potentially capturing both short-term changes in behavior – contrasting the short-term purchase patterns with normal ones, either daily or weekly – and a long-term structure of the transactions, which arises from analyzing the different networks that can be inferred from the data. Results show a very high AUC score and accuracy, especially for Random Forests. Even after adjusting the model to only allow 1% false positives, a high specificity is

achieved, meaning that the proposed model efficiently identifies fraudulent transactions. Finally, this chapter showed that we are not only able to find almost all fraudulent transactions, but also accurately pick out the first transaction in a series of fraudulent transactions, which is an important requirement in curtailing credit card transaction fraud.

7.2 Future research

7.2.1 Network dynamics

According to Definition 3.1, one of the challenges that concur with fraud is the time-evolving property. The aforementioned approaches implemented this property in the link weight between nodes of the network whereby a higher weight is assigned to more recent relationships. Moreover, the amount of fraud propagated through the network by a fraudulent node depends on its recency. Again, the more recently the node was captured as a fraud, the more influence it has on the neighboring nodes. Results show that time is an important element to boost the detection models. Future research should elaborate further upon the time property and should answer questions like: how much history should be taken into account? What is the optimal time horizon to predict future fraud? How should past relationships and past fraud be weighted in time, and what is the best weighing method? The GOTCHA! model as described in Chapter 3 incorporates all history to decide upon future fraud. Is it recommended to use all this history, or would it be better to only consider a limited timespan? And if so, how much history should be used? A thorough domain-wide study should address these questions, and should evaluate the optimal decision strategy in terms of (1) performance, and (2) scalability. Applications that have to deal with streaming data (like credit card transactions), collect and process huge amounts of data. Due to scalability issues, it is far from straightforward to include all history as this might result in networks of huge sizes.

Given the incorporation of historic information - regardless of the timespan, how should historical relationships be weighted in time? In this dissertation, the strength of relationships between the network nodes - by means of the link weight - is exponentially decayed upon its recency. The decay constant is γ (for decaying edge weight) and

β (for decaying past fraud). Due to a non-disclosure agreement, we cannot further elaborate on the exact value of γ and β , but we can say that it is a fixed, arbitrarily chosen value, based on experts' intuition. What is the effect if we change this value. More generally speaking, how should the optimal value be chosen? Do other decay strategies, like a linear or stepwise decay, perform similarly?

All models developed for the Social Security Institution (Chapter 3 - 5) are evaluated on their performance for three time windows: short, medium and long term. So far, the time window is arbitrarily chosen, and set on 6 months (short term), 12 months (medium term) and 24 months (long term). It is shown that, although long-term models seem to perform best, short-term models are not only capable of identifying companies sensitive to commit fraud in the near future, but also anticipate companies that are likely to commit fraud on long term. As such, short-term models are preferred. A more profound study should indicate what the optimal time horizon is to evaluate the models, and which strategy should be used to decide on the right time window.

Extracting a set of features that captures a company's behavior over time, might result in a better performance and gain important insights. *Survival analysis* for fraud detection can help in the study of a company's behavior over time. In terms of network analysis, this may refer to the probability that (1) a label (e.g., non-fraud or fraud) of a node in the network does not change after a certain timestamp t , or (2) a new link between two nodes in the network exists at time t , or (3) even a new node pops up in the network after time t .

How do fraudulent subgraphs or cliques evolve over time? The analyses discussed in this dissertation extracted cliques at a predetermined timestamp, neglecting how such cliques have been developed in the past and how their structure continues to expand in the future. *Time-dependent co-clustering* (i.e., a technique to split the network in subgraphs) should be further explored. One possibility is to add history nodes to the network, representing (a) past version(s) of each current node in the network and its connections with other history nodes. Each history node is connected with its corresponding successor node. Based on such network representation, cliques

generated by co-clustering or similar approaches might contain both history as current nodes indicating (1) which nodes are currently part of the clique and since when (both the the current node and its history nodes are part of the clique); and (2) which nodes were part of the clique since/until when (only history nodes are part of the clique).

Another issue that applies on many fraud detection problems, is the existence of repeated frauds over time. This refers to the fact that the label of an instance (let's say, a person) changes over time. A person might commit fraud on time t_x and t_{x+s} , but acts legally in between. This is indisputably connected to the *anticipating* and *forgiving* effect of the GOTCHA! propagation algorithm that anticipates fraudulent behavior of resources, and forgives their association with fraudulent companies over time. Although the main objective of GOTCHA! is to catch companies – whose label cannot change over time: a company is active and legitimate, or a company is fraudulent and bankrupt – a more thorough analysis should be perpetrated to investigate the effect of fraud from the point of view of the resource. The following questions should be answered: When is a resource enticed to commit fraud *again*? What are the underlying motives that trigger fraudulent behavior? Can we predict which resources are coincidentally related to fraudulent companies, and which resources are rather responsible to infect other companies with fraud?

7.2.2 Multi-view learning

In many applications, each data sample can be described by data collected from different domains or obtained from various feature extractors which exhibit heterogeneous properties (Xu et al., 2013). A view is exactly the set of features extracted from one specific domain. In the light of this dissertation, both the intrinsic feature set and the relational feature set can be seen as a separate view of the same problem. This work mainly focuses on the concatenation of the two views into one single view. That is, the combination of both intrinsic and relational features which are then used as one single feature set for learning. It is shown, however, that models learned from one single view are more sensitive to over-fitting (Xu et al., 2013).

Other approaches exist to deal with multiple views. Future research should further elaborate on how *co-training* – a multi-view learning approach – is able to improve the classification performance of the current models. In co-training, a model for each view is learned and each model is alternately updated by using the results of the model in the other view. In such a way, the knowledge obtained from one view has impact on the model in the other view, and vice versa. This process is repeated until the mutual agreement between the classifiers on unlabeled data is maximal. Another, more simplistic approach is to apply ensemble learning, where we learn a model or *base classifier* for each view and then use the output of each base classifier to learn a so-called *meta-classifier*. The meta-classifier determines the label of each sample as a combination of the base classifiers. Rather than using a meta-classifier to combine the results of the base classifiers, rank aggregation heuristics such as local Kemenization or α -approximation are able to derive a global ranking based on the local rankings of each base classifier. For a more detailed overview of multi-view learning, we refer to Xu et al. (2013).

7.2.3 Rationales

This dissertation is an intertwinement between theory and practice. The development of all approaches in this dissertation starts from a thorough understanding of the problem by closely interacting with domain experts. Definition 3.1 underlines the importance of domain knowledge in fraud detection. Inspectors often have a good intuition where to look and how fraud is perpetrated. While this work only uses domain knowledge in the development of the models, recent work (Sharma et al., 2015) proposed to integrate experts' rationales – i.e., the reason why they label a certain instance as fraud or legitimate – into the models and update the model accordingly.

More specifically, in Chapter 5, we introduced active inference – a subdomain of active learning – where we ask inspectors about the true label of a set of companies such that the expected label of all other companies is optimized. However, we only require from the experts to label instances without specifying why they assigned a specific label to these instances. Nevertheless, the rationale behind an expert's decision is valuable and can positively impact the prediction perfor-

mance of other, similar instances. Sharma et al. (2015) proposed a classifier-agnostic approach to integrate rationales in document classification. That is, the authors asked the labeler to (1) label a set of documents, and (2) highlight the words or phrases that affected their decision.

In this context, we have to ask inspectors (1) about the true label of a company, and (2) the rationale behind their decision (e.g., which features determine the label). Future research should investigate the following questions: Do rationales positively impact the performance of the fraud detection models? How should rationales be integrated in the feature set (over time)? Are rationales useful for every instance?

7.2.4 Internet of Things (IoT)

With the rise of Internet of Things (IoT), many opportunities open up for network analysis in fraud detection. IoT is the existence of one big, interconnected world of electronic devices, sensors, software, IT infrastructure, etc. (Baesens et al., 2015). The majority of devices is no longer operated by humans, but they rather function automatically. As a result, much more data can be collected and analyzed. In a fraud detection context, this means that both the fraudsters and the fraud fighters benefit from this new technology. From the fraudsters' point of view, since IoT exceeds ever known sizes, it is easy to find and take advantage of loopholes in the interconnected network. On the other hand, it creates emerging opportunities for fraud fighters as they are able to investigate fraud from multiple point of views (e.g., the combination of smart meters, weather forecasts, traffic maps, etc.). However, it is far from straightforward how to deal with this big network of interrelated things. This dissertation focused on how to extract data from unipartite (as a baseline in Chapter 3), bipartite (social security fraud) and tripartite (credit card fraud) graphs, where a graph contains one, two or three node types respectively. Internet of Things extends current approaches to multipartite or n -partite graphs. Such networks can quickly grow to immense sizes. How do we connect n node types with each other, in a scalable manner and such that we are not penalized in interpretability? Advances in analytics of IoT strongly depend on advances in other research domains (e.g., face/speech recognition, smart grids, text/opinion mining, etc.). As

we are still at the beginning of the expansion of IoT, for now, it should be seen and researched as a “plug-and-playground” where extra input can be systematically added in a flexible way. From this perspective, state-of-the-art advances in data science – and in particular in network analytics – with regard to IoT, can gradually grow together.

List of Figures

1.1	Knowledge Discovery in Databases (KDD) process.	2
1.2	The fraud triangle.	4
2.1	Köningsberg bridges.	20
2.2	Identity theft. The frequent contact list (C1-C6) of a person is suddenly extended with other contacts (dark nodes). This might indicate that a fraudster (dark node in the center) took over that customer's account and "shares" his/her contacts.	22
2.3	Network representation.	23
2.4	Example of a (un)directed graph.	24
2.5	Follower-followee relationships in a Twitter network.	25
2.6	Edge representation.	26
2.7	Example of a fraudulent network.	27
2.8	An egonet. The ego is surrounded by 6 alters of whom 2 are legitimate (light-colored) and 4 are fraudulent (dark-colored).	28
2.9	Example of credit card transaction data.	29
2.10	Toy example of credit card fraud.	30
2.11	Mathematical representation of (a) a sample network : (b) the adjacency or connectivity matrix; (c) the weight matrix; (d) the adjacency list and (e) the weight list.	31
2.12	A homophilic network.	33
2.13	Illustration of the degree distribution for a real-life network of social security fraud. The degree distribution follows a power law (log-log axes).	36

3.1	Example of a spider construction. Company 1 and 4 are fraudulent. Resources are transferred towards other companies (solid line). The key company organizes the fraudulent setup, but its links to other companies are hidden (dashed line).	45
3.2	Bipartite graph of a spider construction. Companies are indirectly connected to each other through the resources. .	46
3.3	Overview of the total number of active companies (blue curve) and fraudulent companies (red curve). The number of active companies is consistently growing. A similar trend can be noticed in the number of fraudulent companies. . .	48
3.4	Overview of the different stages a company can go through when ending its economic lifecycle. Even though it is hard to detect fraud <i>ex ante</i> , it is also a challenging task to define fraud <i>ex post</i> . Bankruptcy corresponds to the disability of paying back debts to the social security institution. It is not straightforward which companies are evidence of regular economic failure and which companies went bankrupt due to some fraudulent structure.	49
3.5	Real-life example of fraud propagating through a sub-network over time. Legitimate companies are unfilled, fraudulent companies are filled. The initial situation is represented in (a). When time passes, more nodes are influenced by fraudulent behavior of their neighbors (b), ultimately infecting almost the whole subgraph (c). This confirms the contagious effect of fraud.	53
3.6	Fraud detection process for the social security institution. .	56
3.7	Proposed GOTCHA! framework for social security fraud detection.	58
3.8	Example of a preprocessed data set.	59
3.9	Overview of a unipartite (a) and a bipartite (b) graph. . .	60
3.10	Exponentially weighting the recency of the relationships between companies and resources to determine the tie strength using different values of γ	62

-
- 3.11 Overview of propagation task. Only a limited number of companies is labeled. Using a propagation algorithm we infer an exposure score for each company and resource in the network, representing the extent to which a company/resource is exposed to fraud. 64
- 3.12 Illustration of GOTCHA!'s propagation algorithm. The dark node in the center propagates its fraudulent influence to its neighbors (step 1) The neighbors absorb the influence and propagate on their turn their fraudulent influence to their neighbors (step 1 + 2). The iterations are repeated several times until convergence. 65
- 3.13 The exposure score for each node depends on (a) the exposure scores of the node's neighborhood (left figure) and (b) a random jump towards another node in the network (right figure). 66
- 3.14 Each resource is associated with its propagated exposure score and its presence in fraudulent companies. The resources are colored according to their riskiness (red indicates high risk, green is low risk). The horizontal line represents the boundary dividing the resources in a low-risk and high-risk category. Note that only 0.28% of all resources are labeled as high-risk. 69
- 3.15 Various egonets for micro- (a), small- (b) and medium-sized (c) companies. The company is the center (i.e., the ego) of the egonet and is surrounded by its resources (i.e., the alters). High-risk resources are labeled in black, low-risk nodes are white-colored. All central companies (egos) are still active at the time of analysis. 70
- 3.16 Variable Importance of Random Forests for timestamp t_3 . (TW = time-weighted; LR = low-risk; HR = high-risk; N = neighborhood). 81
- 3.17 Precision and recall for the various models. 84
- 3.18 ROC analysis of the proposed approach applied in practice. All models are estimated on *year* t_2 and tested on *year* t_3 86

3.19	Evolution of a spider construction over time. The network represents the nodes and connections as observed at time t_0 . Only one company is fraudulent, but passes many resources to other companies. Future data shows that three extra companies will commit fraud, and two companies will go bankrupt. If we had applied GOTCHA!, our results show that we could have avoided the development of this spider construction at time t_0	89
4.1	Subgraph of companies (large nodes) connected to their resources (small nodes). Fraudulent companies are dark-colored, currently-legitimate companies are light-colored. Companies form cliques (i.e., fully connected subgraphs) based on their use of the same set of resources.	94
4.2	Flow-chart of detection process.	98
4.3	Clique detection process. Companies A , B and C share the same (set of) resources. The top figure illustrates the merging process for an exact match between pairs of companies. The original pairs are deleted from the final set. Only clique α is in the remaining set of cliques. The bottom figure represents a partial overlap between pairs of companies. Here, the original pairs β and γ , together with a new clique α are all added to the new set of cliques.	101
4.4	ROC curves for the different timestamps of our analysis. Notice that the combined model which includes all of the intrinsic, relational, and clique-based features outperforms the models using any one of those features alone.	106
4.5	Precision of the top 100 most high-risk companies. Generally speaking, long-term models perform better than short- and medium-term models.	109
4.6	Variable Importance of the top 15 features in the combined model.	110
5.1	Fraud process: a fraudulent company files for bankruptcy in order to avoid paying taxes and transfers its resources to other companies that are part of the illegal setup, also known as a spider construction (see Section 3.2).	115

5.2	A summary graph \mathcal{S}_t at time t contains all nodes and edges observed between time t and time $t - s$	117
5.3	Time-weighted collective inference algorithm. (a) At time t , two companies in the subgraph are fraudulent. The intensity of the color refers to the recency of the fraudulence. (b) Propagation of fraud through the network by GOTCHA!'s propagation algorithm. (c) Cutting the incoming edges after probing node '?' and confirming its non-fraudulent label.	121
5.4	Model performance of active inference on time t for probing strategy MSU+.	127
5.5	Precision achieved by the probing strategies.	128
5.6	Changes in precision for Random Forests compared to changes in the evaluation set when using probing strategy MSU+.	129
5.7	Precision of the committee-based probing strategies.	130
5.8	Precision of the experts-based probing strategies.	130
5.9	Precision when inspectors' decisions are weighted in time.	131
6.1	Toy example of a credit card fraud network. Weights depict the recency of the transaction between the merchant and credit card holder.	136
6.2	Credit Card Detection Process.	141
6.3	APATE's re-estimation process of the detection models using a sliding window.	145
6.4	Adjacency matrix of the bipartite graph (a), transformed bipartite-to-unipartite graph (b) and transformed tripartite-to-unipartite graph (c).	150
6.5	Example of a multi-edge subnetwork. The credit card holder made several transactions at the same merchant.	151
6.6	Transformation process from a bipartite to a tripartite graph by representing the edges as a separate node in the network.	153
6.7	Local updating process when new transaction appears in the network.	157
6.8	ROC Curve for Different Models.	160
6.9	ROC Curve for Different Subsets of Variables.	162

- 6.10 Importance of each variable for the Random Forests built using all available variables. In parentheses after each variable is the relevant information from the logistic regression output: A + or - sign of significant coefficients, NS when the variable was not significant, and C when it was highly correlated (greater than 0.9) with another variable in the data set. 164

List of Tables

2.1	Overview of neighborhood metrics.	37
2.2	Overview of centrality metrics.	38
3.1	Overview of all published papers related to fraud detection using network analytics.	55
3.2	Network-based feature extraction for (u) Unipartite, (b) Bipartite, (G) GOTCHA!.	71
3.3	Network-based feature extraction.	74
3.4	AUC scores of the baseline and GOTCHA! models.	79
3.5	P-values of the AUC scores.	80
3.6	Variable importance and sign of the GOTCHA! model for social security fraud detection. A positive sign indicates a positive contribution of that variable to fraud. A negative sign means that the variable negatively impacts fraud.	82
3.7	Future lifecycle of detected companies. All the models are estimated on short-term fraud, but are able to identify high-risk companies after the predetermined time window.	87
4.1	P-values of the AUC scores.	108
6.1	Overview of published papers in the credit card fraud detection domain.	143
6.2	Summary of input features on short (ST), medium (MT) and long (LT) term.	147
6.3	Transactions per Region and Fraud Percentage.	149
6.4	Comparison of models.	159
6.5	Accuracy and AUC (test set) at 1% Maximum False Positive Rate.	161

6.6	AUC for Different Subsets of Variables.	161
-----	---	-----

Bibliography

- Leman Akoglu, Mary McGlohon, and Christos Faloutsos. Oddball: Spotting anomalies in weighted graphs. In *Proceedings of the 14th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD)*, pages 410–421. Springer, 2010.
- Leman Akoglu, Rishi Chandy, and Christos Faloutsos. Opinion fraud detection in online reviews by network effects. In *Proceedings of the 7th International Conference on Weblogs and Social Media (ICWSM)*, pages 1–11. AAAI, 2013.
- Emin Aleskerov, Bernd Freisleben, and Bharat Rao. Cardwatch: A neural network based database mining system for credit card fraud detection. In *Proceedings of the 1997 IEEE/IAFE Computational Intelligence for Financial Engineering (CIFER)*, pages 220–226. IEEE, 1997.
- Paul D. Allison. *Missing data. Quantitative Applications in the Social Sciences*, volume 136. Sage University Paper, 2001.
- Sinan Aral, Lev Muchnik, and Arun Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences*, 106(51):21544–21549, 2009.
- Aimée Backiel, Bart Baesens, and Gerda Claeskens. Mining telecommunication networks to enhance customer lifetime predictions. In *Proceedings of the 14th International Conference on Artificial Intelligence and Soft Computing (ICAISC)*, pages 15–26. Springer, 2014.

- Bart Baesens. *Analytics in a Big Data World: The Essential Guide to Data Science and Its Applications*. John Wiley & Sons, 2014.
- Bart Baesens, Rudy Setiono, Christophe Mues, and Jan Vanthienen. Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science*, 49(3):312–329, 2003a.
- Bart Baesens, Tony Van Gestel, Stijn Viaene, Maria Stepanova, Johan Suykens, and Jan Vanthienen. Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, 54(6):627–635, 2003b.
- Bart Baesens, Véronique Van Vlasselaer, and Wouter Verbeke. *Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection*. John Wiley & Sons, 2015.
- Alejandro Correa Bahnsen, Aleksandar Stojanovic, Djamila Aouada, and Bjorn Ottersten. Cost-sensitive credit card fraud detection using bayes minimum risk. In *Proceedings of the 12th International Conference on Machine Learning and Applications (ICMLA)*, volume 1, pages 333–338. IEEE, 2013.
- Alejandro Correa Bahnsen, Aleksandar Stojanovic, Djamila Aouada, and Björn Ottersten. Improving credit card fraud detection with calibrated probabilities. In *Proceedings of the 2014 SIAM International Conference on Data Mining (SDM)*, 2014.
- Ravi Bapna and Akhmed Umyarov. Do your online friends make you pay? A randomized field experiment in an online music social network. *NBER Working Paper*, 2012.
- Tej Paul Bhatla, Vikram Prabhu, and Amit Dua. Understanding credit card frauds. *Cards Business Review*, 1(6), 2003.
- Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, and J. Christopher Westland. Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3):602–613, 2011.

- Mustafa Bilgic and Lise Getoor. Effective label acquisition for collective classification. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 43–51. ACM, 2008.
- Mustafa Bilgic and Lise Getoor. Reflect and correct: A misclassification prediction approach to active inference. *Transactions on Knowledge Discovery from Data*, 3(4):1–32, 2009.
- Mustafa Bilgic, Lilyana Mihalkova, and Lise Getoor. Active learning for networked data. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 79–86, 2010.
- Robert C. Blattberg, Byung-Do Kim, and Scott A. Neslin. *Why Database Marketing?* Springer, 2008.
- Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D.-U. Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4):175–308, 2006.
- Richard J. Bolton and David J. Hand. Unsupervised profiling methods for fraud detection. *Credit Scoring and Credit Control VII*, pages 235–255, 2001.
- Richard J. Bolton and David J. Hand. Statistical fraud detection: A review. *Statistical Science*, 17(3):235–249, 2002.
- .R Brause, T. Langsdorf, and Michael Hepp. Neural data mining for credit card fraud detection. In *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 103–106. IEEE, 1999.
- Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- Emilio Carrizosa, Belén Martín-Barragán, and Dolores Romero Morales. A nested heuristic for parameter tuning in support vector machines. *Computers & Operations Research*, 43: 328–334, 2014.
- Deepayan Chakrabarti, Spiros Papadimitriou, Dharmendra S. Modha, and Christos Faloutsos. Fully automatic cross-associations. In *Proceedings of the 10th ACM SIGKDD International Conference on*

- Knowledge Discovery and Data Mining (SIGKDD)*, pages 79–88. ACM, 2004.
- Philip K. Chan and Salvatore J. Stolfo. Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 164–168, 1998.
- Philip K. Chan, Wei Fan, Andreas L. Prodromidis, and Salvatore J. Stolfo. Distributed data mining in credit card fraud detection. *Intelligent Systems and their Applications, IEEE*, 14(6):67–74, 1999.
- Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):15, 2009.
- Duen Horng Chau, Shashank Pandit, and Christos Faloutsos. Detecting fraudulent personalities in networks of online auctioneers. In *Proceedings of the 10th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECLM/PKDD)*, pages 103–114. Springer, 2006.
- Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2011.
- Chao Chen, Andy Liaw, and Leo Breiman. Using random forest to learn imbalanced data. Statistics Tech. Report 666, University of California, Berkeley, 2004a.
- Hsinchun Chen, Wingyan Chung, Jennifer Jie Xu, Gang Wang, Yi Qin, and Michael Chau. Crime data mining: A general framework and some examples. *Computer*, 37(4):50–56, 2004b.
- Chaochang Chiu, Yungchang Ku, Ting Lie, and Yuchi Chen. Internet auction fraud detection using social network analysis and classification tree approaches. *International Journal of Electronic Commerce*, 15(3):123–147, 2011.

- Corinna Cortes, Daryl Pregibon, and Chris Volinsky. Communities of interest. In *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis (IDA)*, pages 105–114. Springer, 2001.
- Donald R. Cressey. Other people's money: A study of the social psychology of embezzlement. 1953.
- Andrea Dal Pozzolo, Olivier Caelen, Yann-Aël Le Borgne, Serge Waterschoot, and Gianluca Bontempi. Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*, 41(10):4915–4928, 2014.
- Johannes De Smedt, Jochen De Weerd, and Jan Vanthienen. Fusion miner: Process discovery for mixed-paradigm models. *Decision Support Systems*, 2015.
- Linda Delamaire, Hussein Abdou, and John Pointon. Credit card fraud and detection techniques: a review. *Banks and Bank systems*, 4(2):57–68, 2009.
- Cassius Dio. *Historiae Romanae*, volume LXXIV of 8-16. c. 170.
- Georges Dionne, Florence Giuliano, and Pierre Picard. Optimal auditing with scoring: Theory and application to insurance fraud. *Management Science*, 55(1):58–70, 2009.
- Jose R. Dorronsoro, Francisco Ginel, C. Sgnchez, and C. S. Cruz. Neural fraud detection in credit card operations. *IEEE Transactions on Neural Networks*, 8(4):827–834, 1997.
- Ekrem Duman and Ilker Elikucuk. Solving credit card fraud detection problem by the new metaheuristics migrating birds optimization. In *Advances in Computational Intelligence*, pages 62–71. Springer, 2013.
- David Easley and Jon Kleinberg. *Networks, Crowds, and Markets*. Cambridge University Press, 2010.
- ECB. Third report on fraud. *European Central Bank*, 2014.

- Tina Eliassi-Rad and Keith Henderson. Ranking information in networks. In *Proceedings of the 4th International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction (SBP)*, pages 268–275. Springer, 2011.
- Pablo A. Estévez, Claudio M. Held, and Claudio A. Perez. Subscription fraud prevention in telecommunications using fuzzy rules and neural networks. *Expert Systems with Applications*, 31(2):337–344, 2006.
- Andrew Fast, Lisa Friedland, Marc Maier, Brian Taylor, David Jensen, Henry G. Goldberg, and John Komoroske. Relational data pre-processing techniques for improved securities fraud detection. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 941–949. ACM, 2007.
- Tom Fawcett and Foster Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997.
- Štefan Furlan and Marko Bajec. Holistic approach to fraud management in health insurance. *Journal of Information and Organizational Sciences*, 32(2):99–114, 2008.
- Brian Gallagher, Hanghang Tong, Tina Eliassi-Rad, and Christos Faloutsos. Using ghost edges for classification in sparsely labeled networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 256–264. ACM, 2008.
- John Galloway and Simeon J. Simoff. Network data mining: Discovering patterns of interaction between attributes. In *Proceedings of the 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD)*, pages 410–414. Springer, 2006.
- Zengan Gao and Mao Ye. A framework for data mining-based anti-money laundering research. *Journal of Money Laundering Control*, 10(2):170–179, 2007.

- Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):721–741, 1984.
- Sushmito Ghosh and Douglas L. Reilly. Credit card fraud detection with a neural-network. In *Proceedings of the 27th Hawaii International Conference on System Sciences (HICSS)*, volume 3, pages 621–630. IEEE, 1994.
- David F. Gleich. Pagerank beyond the web. *arXiv preprint arXiv:1407.5107*, 2014.
- Henry G. Goldberg and Ted E. Senator. Restructuring databases for knowledge discovery by consolidation and link formation. In *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining (KDD)*, volume 95, pages 136–141, 1995.
- Mangesh Gupte and Tina Eliassi-Rad. Measuring tie strength in implicit social networks. In *Proceedings of the 3rd Annual ACM Web Science Conference (WebSci)*, pages 109–118. ACM, 2012.
- Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *Proceedings of the 30th International Conference on Very Large Databases (VLDB)*, pages 576–587. VLDB Endowment, 2004.
- Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*, volume 1. Springer New York, 2001.
- Keith Henderson, Brian Gallagher, Lei Li, Leman Akoglu, Tina Eliassi-Rad, Hanghang Tong, and Christos Faloutsos. It’s who you know: Graph mining using recursive structural features. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 663–671. ACM, 2011.
- Constantinos S. Hilas and John N. Sahalos. User profiling for fraud detection in telecommunication networks. In *Proceedings of the 5th International Conference on Technology and Automation (ICTA)*, pages 382–387, 2005.

- Shawndra Hill, Foster Provost, and Chris Volinsky. Learning and inference in massive social networks. In *Proceedings of the 5th International Workshop on Mining and Learning with Graphs (MLG)*, 2007.
- David Jensen. Prospective assessment of AI technologies for fraud detection: A case study. In *Proceedings of the 1997 AAAI Workshop on AI Approaches to Fraud Detection and Risk Management*, pages 34–38. AAAI, 1997.
- Sanjeev Jha, Montserrat Guillen, and J. Christopher Westland. Employing transaction aggregation strategy to detect credit card fraud. *Expert Systems with Applications*, 39(16):12650 – 12657, 2012.
- Danai Koutra, Tai-You Ke, U Kang, Duen Horng Polo Chau, Hsing-Kuo Kenneth Pao, and Christos Faloutsos. Unifying guilt-by-association approaches: Theorems and fast algorithms. In *Proceedings of the 2011 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 245–260. Springer, 2011.
- M. Krivko. A hybrid model for plastic card fraud detection systems. *Expert Systems with Applications*, 37(8):6070 – 6076, 2010.
- Ankit Kuwadekar and Jennifer Neville. Relational active learning for joint collective classification models. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 385–392, 2011.
- Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pages 591–600. ACM, 2010.
- Daniel Z. Levin and Rob Cross. The strength of weak ties you can trust: The mediating role of trust in effective knowledge transfer. *Management Science*, 50(11):1477–1490, 2004.
- Qing Lu and Lise Getoor. Link-based classification. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, volume 3, pages 496–503, 2003.

- Sofus A Macskassy. Using graph-based metrics with empirical risk minimization to speed up active learning on networked data. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 597–606. ACM, 2009.
- Sofus A. Macskassy and Foster Provost. Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning Research*, 8:935–983, 2007.
- Sam Maes, Karl Tuyls, Bram Vanschoenwinkel, and Bernard Manderick. Credit card fraud detection using bayesian and neural networks. In *Proceedings of the 1st International NAISO Congress on Neuro Fuzzy Technologies*, pages 16–19, 2002.
- Mary McGlohon, Stephen Bay, Markus G. Anderle, David M. Steier, and Christos Faloutsos. Snare: A link analytic system for graph labeling and risk detection. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 1265–1274. ACM, 2009.
- Miller McPherson, Lynn Smith-Lovin, and James M. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.
- Helen Tadesse Moges, Véronique Van Vlasselaer, Wilfried Lemahieu, and Bart Baesens. Determining the use of data quality metadata (dqm) for decision making purposes and its impact on decision outcomes – an exploratory study. *Decision Support Systems*, under review.
- Jay Robert Nash. *The Great Pictorial History of World Crime*, volume 1. Scarecrow Press, 2004.
- Jennifer Neville, Özgür Şimşek, David Jensen, John Komoroske, Kelly Palmer, and Henry Goldberg. Using relational knowledge discovery to prevent securities fraud. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (SIGKDD)*, pages 449–458. ACM, 2005.

- Mark Newman. *Networks: an introduction*. Oxford University Press, 2010.
- Tore Opsahl and Pietro Panzarasa. Clustering in weighted networks. *Social Networks*, 31(2):155–163, 2009.
- Lawrence Page. Method for node ranking in a linked database, 2001. US Patent 6,285,999.
- Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.
- Shashank Pandit, Duen Horng Chau, Samuel Wang, and Christos 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 (WWW)*, pages 201–210. ACM, 2007.
- Juyong Park and Albert-László Barabási. Distribution of node characteristics in complex networks. *Proceedings of the National Academy of Sciences*, 104(46):17916–17920, 2007.
- Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artificial intelligence*, 29(3):241–288, 1986.
- Clifton Phua, Daminda Alahakoon, and Vincent Lee. Minority report in fraud detection: classification of skewed data. *ACM SIGKDD Explorations Newsletter*, 6(1):50–59, 2004.
- B. Aditya Prakash, Hanghang Tong, Nicholas Valler, Michalis Faloutsos, and Christos Faloutsos. Virus propagation on time-varying networks: Theory and immunization algorithms. In *Proceedings of the 2010 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 99–114. Springer, 2010.
- Foster Provost. Machine learning from imbalanced data sets. In *Proceedings of the 2000 AAAI Workshop on Imbalanced Data Sets*, pages 1–3, 2000.

- Foster Provost, Brian Dalessandro, Rod Hook, Xiaohan Zhang, and Alan Murray. Audience selection for on-line brand advertising: Privacy-friendly social network targeting. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 707–716. ACM, 2009.
- Jon T.S. Quah and M. Sriganesh. Real-time credit card fraud detection using computational intelligence. *Expert Systems with Applications*, 35(4):1721–1732, 2008.
- Matthew J. Rattigan, Marc Maier, and David Jensen. Exploiting network structure for active inference in collective classification. In *Proceedings of the 7th International Conference on Data Mining (ICDM)*, pages 429–434. IEEE, 2007.
- Ryan Rossi and Jennifer Neville. Time-evolving relational classification and ensemble methods. In *Proceedings of the 16th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD)*, pages 1–13. Springer, 2012.
- Daniel Sánchez, M.A. Vila, L. Cerda, and José-María Serrano. Association rules applied to credit card fraud detection. *Expert Systems with Applications*, 36(2):3630–3640, 2009.
- Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93, 2008.
- Alex Seret, Thomas Verbraken, Sébastien Versailles, and Bart Baesens. A new som-based method for profile generation: Theory and an application in direct marketing. *European Journal of Operational Research*, 220(1):199–209, 2012.
- Burr Settles. Active learning literature survey. Computer sciences technical report 1648, University of Wisconsin-Madison, 2009.
- Umang Sharan and Jennifer Neville. Exploiting time-varying relationships in statistical relational models. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis (WebKDD)*, pages 9–15. ACM, 2007.

- Manali Sharma and Mustafa Bilgic. Most-surely vs. least-surely uncertain. In *Proceedings of the 13th IEEE International Conference on Data Mining (ICDM)*, pages 667–676. IEEE, 2013.
- Manali Sharma, Di Zhuang, and Mustafa Bilgic. Active learning with rationales for text classification. In *North American Chapter of the Association for Computational Linguistics – Human Language Technologies*, 2015. URL <http://www.cs.iit.edu/~ml/pdfs/sharma-naaclhlt15.pdf>.
- Aihua Shen, Rencheng Tong, and Yaochen Deng. Application of classification models on credit card fraud detection. In *Proceedings of the 2007 International Conference on Service Systems and Service Management (ICSSSM)*, pages 1–4. IEEE, 2007.
- Tommie W. Singleton and Aaron J. Singleton. *Fraud auditing and forensic accounting*, volume 11. John Wiley & Sons, 2010.
- Geoff Smith, Mark Button, Les Johnston, and Kwabena Frimpong. *Studying fraud as white collar crime*. Palgrave Macmillan, 2010.
- Abhinav Srivastava, Amlan Kundu, Shamik Sural, and Arun K. Majumdar. Credit card fraud detection using hidden markov model. *IEEE Transactions on Dependable and Secure Computing*, 5(1):37–48, 2008.
- Rodney T. Stamler, Hans J. Marschdorf, and Mario Possamai. *Fraud Prevention and Detection: Warning Signs and the Red Flag System*. CRC Press, 2014.
- Salvatore Stolfo, W. Fan, Wenke Lee, Andreas Prodromidis, and P. Chan. Credit card fraud detection using meta-learning: Issues and initial results. In *Proceedings of the 1997 AAAI Workshop on Fraud Detection and Risk Management*, 1997.
- Lovro Šubelj, Štefan Furlan, and Marko Bajec. An expert system for detecting automobile insurance fraud using social network analysis. *Expert Systems with Applications*, 38(1):1039–1052, 2011.
- Mubeena Syeda, Yan-Qing Zhang, and Yi Pan. Parallel granular neural networks for fast credit card fraud detection. In *Proceedings of*

- the 2002 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, volume 1, pages 572–577. IEEE, 2002.
- Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Proceedings of the 6th International Conference on Data Mining (ICDM)*, pages 613–622. IEEE, 2006.
- Hanghang Tong, Spiros Papadimitriou, S. Yu Philip, and Christos Faloutsos. Proximity tracking on time-evolving bipartite graphs. In *Proceedings of the 2008 SIAM International Conference on Data Mining (SDM)*, pages 704–715, 2008.
- Véronique Van Vlasselaer, Thomas Verbraken, and Bart Baesens. Mining data on twitter. Master’s thesis, KU Leuven, 2012.
- Véronique Van Vlasselaer, Jan Meskens, Dries Van Dromme, and Bart Baesens. Using social network knowledge for detecting spider constructions in social security fraud. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Network Analysis and Mining (ASONAM)*, pages 813–820. IEEE, 2013.
- Véronique Van Vlasselaer, Cristián Bravo, Olivier Caelen, Tina Eliassi-Rad, Leman Akoglu, Monique Snoeck, and Bart Baesens. Apate: A novel approach for automated credit card transaction fraud detection using network-based extensions. *Decision Support Systems*, 75:38–48, 2015.
- Véronique Van Vlasselaer, Tina Eliassi-Rad, Leman Akoglu, Monique Snoeck, and Bart Baesens. Gotcha! network-based fraud detection for social security fraud. *Management Science*, under review.
- Sepepe vanden Broucke, Jochen De Weerd, Jan Vanthienen, and Bart Baesens. Determining process model precision and generalization with weighted artificial negative events. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1877–1889, 2014.
- Wouter Verbeke. *Profit-driven Data Mining in Massive Customer Networks: New insights and algorithms*. PhD thesis, KU Leuven, Belgium, 2012.

- Wouter Verbeke, David Martens, Christophe Mues, and Bart Baesens. Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications*, 38(3):2354–2364, 2011.
- Jyun-Cheng Wang and Chui-Chen Chiu. Recommending trusted online auction sellers using social network analysis. *Expert Systems with Applications*, 34(3):1666–1679, 2008.
- David J. Weston, David J. Hand, Niall M. Adams, Christopher Whitrow, and Piotr Juszczak. Plastic card fraud detection using peer group analysis. *Advances in Data Analysis and Classification*, 2(1):45–62, 2008.
- Richard Wheeler and Stuart Aitken. Multiple algorithms for fraud detection. *Knowledge-Based Systems*, 13(2):93–99, 2000.
- Christopher Whitrow, David J. Hand, Piotr Juszczak, D. Weston, and Niall M. Adams. Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery*, 18(1):30–55, 2009.
- Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *preprint 1304.5634*, 2013.
- Zhu Yanchun, Zhang Wei, and Yu Changhai. Detection of feedback reputation fraud in taobao using social network theory. In *Proceedings of the 2011 International Joint Conference on Service Sciences (IJCSS)*, pages 188–192. IEEE, 2011.
- Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239, 2003.
- Vladimir Zaslavsky and Anna Strizhak. Credit card fraud detection using self-organizing maps. *Information and Security*, 18:48, 2006.

Publication list

Articles in internationally reviewed scientific journals

- Véronique Van Vlasselaer, Cristián Bravo, Olivier Caelen, Tina Eliassi-Rad, Leman Akoglu, Monique Snoeck, and Bart Baesens. Apate: A novel approach for automated credit card transaction fraud detection using network-based extensions. *Decision Support Systems*, 75:38–48, 2015.
- Carlos A.R. Pinheiro, Véronique Van Vlasselaer, Bart Baesens, Alexandre G. Evsukoff, Moacyr A.H.B. Silva, and Nelson F.F. Ebecken. A models comparison to estimate commuting trips based on mobile phone data. *Software Engineering in Intelligent Systems*, 35–44, 2015.

Articles submitted for publication in internationally reviewed scientific journals

- Véronique Van Vlasselaer, Tina Eliassi-Rad, Leman Akoglu, Monique Snoeck, and Bart Baesens. Gotcha! network-based fraud detection for social security fraud. *Management Science*, under review.
- Helen Tadesse Moges, Véronique Van Vlasselaer, Wilfried Lemahieu, and Bart Baesens. Determining the use of data quality metadata (dqm) for decision making purposes and its impact on decision outcomes – an exploratory study. *Decision Support Systems*, under review.

Academic books with internationally recognized scientific publisher

- Bart Baesens, Véronique Van Vlasselaer, and Wouter Verbeke. *Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection*. John Wiley & Sons, 2015.
- Thomas Verbraken, Véronique Van Vlasselaer, Wouter Verbeke, David Martens, and Bart Baesens. *Advanced Rule Base Learning: Active Learning, Rule Extraction, and Incorporating Domain Knowledge*. In: Koen W. De Bock, Scott A. Neslin, Kristof Coussement (Eds.), *Advanced database marketing: innovative methodologies & applications for managing customer relationships*. Gower Publishing, 2013.

Papers at international conferences and symposia, published in full in proceedings

- Véronique Van Vlasselaer, Tina Eliassi-Rad, Leman Akoglu, Monique Snoeck, and Bart Baesens. Afraid: fraud detection via active inference in time-evolving social networks. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Network Analysis and Mining (ASONAM)*, IEEE, 2015.
- Véronique Van Vlasselaer, Leman Akoglu, Tina Eliassi-Rad, Monique Snoeck, and Bart Baesens. Guilt-by-constellation: fraud detection by suspicious clique memberships. In *Proceedings of 48th Annual Hawaii International Conference on System Sciences (HICSS)*, 2015.
- Véronique Van Vlasselaer, Jan Meskens, Dries Van Dromme, and Bart Baesens. Using social network knowledge for detecting spider constructions in social security fraud. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Network Analysis and Mining (ASONAM)*, pages 813–820. IEEE, 2013.

Meeting abstracts, presented at international scientific conferences and symposia, published or not published in proceedings or journals

- Véronique Van Vlasselaer, Leman Akoglu, Tina Eliassi-Rad, Monique Snoeck, and Bart Baesens. Finding cliques in large fraudulent networks: theory and insights. *Conference of the International Federation of Operational Research Societies (IFORS)*, 2014.
- Véronique Van Vlasselaer, Jan Meskens, Dries Van Dromme, and Bart Baesens. Social network analysis for detecting spider constructions in social security fraud: new insights and challenges. *European Conference on Operational Research (EURO)*, 2013.

Meeting abstracts, presented at international professionally oriented conferences and symposia, published or not published in proceedings or journals

- Véronique Van Vlasselaer, Leman Akoglu, Tina Eliassi-Rad, Monique Snoeck, and Bart Baesens. Gotch'all! Advanced network analysis for detecting groups of fraud. *Predictive Analytics World (PAW)*, 2014.
- Bart Baesens and Véronique Van Vlasselaer. Social network analytics for fraud detection: insights and challenges. *SAS Analytics*, 2013.
- Véronique Van Vlasselaer and Bart Baesens. Improving fraud detection techniques using social network analytics for the Belgian government. *Predictive Analytics World (PAW)*, 2013.

Doctoral dissertations from the faculty of business and economics

A full list of the doctoral dissertations from the Faculty of Business and Economics can be found at:

www.kuleuven.ac.be/doctoraatsverdediging/archief.htm.