UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Social Network Analysis for Insurance Fraud Detection

Nuno C. Garcia

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática
(2º ciclo de estudos)

Orientador: Prof. Doutor Hugo Proença

Covilhã, Outubro de 2015

Acknowledgments

First of all, I would like to thank Professor Hugo, for his support and guidance through this year. This work is fruit of a collaboration between UBI and Deloitte Portugal. I would like to thank Deloitte Portugal, namely Nuno Carvalho, Tiago Durão and Diogo Fernando for the opportunity of developing research on an enterprise environment, and the fortune of being able to learn with the standards of excellence that guide the company. Also, a word of gratitude to all my colleagues with whom I shared this journey.

To all of my friends, for the happiness they bring to my day, giving me strength to go through adversities.

To my family, for the always important harmony and positive feelings.

To my mother, my father and my sister, for the everyday love and wise advises, for never letting me down.

To Vanessa, for the patience, encouragement and love, that made this a pleasant year, full of light and beauty.

Resumo alargado

A detecção de fraude configura um desafio interessante, que não está totalmente resolvido particularmente no que respeita a fraude em seguros automóvel. A fraude no seguro automóvel representa várias centenas de milhões de euros de prejuízo para as companhias seguradoras na Europa, e consequentemente um aumento de preço das apólices cobrado ao consumidor final. A dimensão do mercado segurador e o impacto que a fraude tem nas companhias faz com que a tarefa de detecção de fraude possa transformada em vantagem competitiva, e assim se assuma como uma prioridade no sector. A fraude que provoca danos mais volumosos é a praticada por grupos organizados, que concebem esquemas e contornam o sistema de forma a sistematicamente repetir a actividade fraudulenta.

Esta dissertação aborda o tema da detecção de fraude de uma perpectiva que não será a mais comum nos sistemas hoje em dia utilizados. Em vez de analisar dados de sinistros como números e estatísticas isoladas, tenta perceber as relações entre as entidades que participam nos sinistros e identificar estruturas suspeitas de entre um vasto conjunto de dados.

O conjunto de dados necessário à análise que propomos tem características especiais, como por exemplo ser sensível a divulgação a terceiros por conter dados pessoais e ser normalmente propriedade das companhias de seguros ou de estruturas policiais. Por estes motivos, não existem conjuntos de dados públicos que permitam o desenvolvimento de uma investigação neste sentido.

Para colmatar este facto, propomos um gerador de grafos aleatório capaz de produzir redes com padrões semelhantes àqueles que seria expectável encontrar em cenários reais. O gerador incorpora conhecimento descrito na literatura [ŠFB11] sobre características e padrões encontrados em conjuntos de dados relacionados com este tema. Além disso, especialistas de seguros da Deloitte, parceira no desenvolvimento desta dissertação, contribuiram com a sua experiência no campo para que o gerador pudesse representar fielmente a realidade.

No que respeita à detecção de fraude, este trabalho propõe uma abordagem que inclui a classificação de componentes do grafo como fraudulentos ou honestos, através do uso do conhecido classificador SVM (*Support Vector Machine*). São feitas avaliações de performance com várias variações do método proposto e de parte do método que inspirou a abordagem usada, chamado PRIDIT. Uma das conclusões mais interessantes que estas experiências parecem sugerir é que nem sempre o uso do método PRIDIT garante o aumento de performance desejado.

As contribuições deste trabalho centram-se no desenvlvimento de um gerador de grafos para o contexto de análise de fraude de seguros automóvel, e na avaliação e comparação do uso de SVM na classificação de componentes fraudulentos.

Abstract

Fraud detection configures a very interesting problem yet to solve, particularly when related to automobile insurance claims. In this research we address this challenge from a not so typical "record" perspective of data, but rather from a network point of view, where relations between entities involved in claims are explored to detect organized fraud structures.
First we propose a random data generator, able to generate graphs that resemble realistic patterns evidenced on authentic scenarios, based on insurance authorities statistics and graph features already described in the literature. We show how this graph copes with the requirements on every single step, and how it can be adjustable to different locals.
Secondly, we propose a variation of Subelj approach [ŠFB11], and apply it to the generated graphs. This approach explores the relations between entities, takes advantage of the power of social network analysis metrics and statistical methods such as RIDIT scores and Principal Component Analysis to score each connected component and Support Vector Machines to classify them either fraudulent or honest.
The main contributions of this research is a new approach to generate data regarding automobile insurance claims suitable for social network analysis, and a variation of an approach described on the literature, proving thus not only benchmark results but also new insights regarding fraud detection through graph-based algorithms.

Keywords

Social network analysis, random graph generator, fraud detection, SVM

Contents

List of Figures

List of Tables

Acronyms

UBI	Universidade da Beira Interior
EU	European Union
OLAF	European Anti-Fraud Office (*Office Européen de Lutte Antifraude*)
APS	Associação Portuguesa de Seguradores
ABI	Association of British Insurers
SVM	Support Vector Machine
PCA	Principal Component Analysis

Chapter 1

Introduction

The term *fraud* refers to "deliberately deceiving someone else with the intent of causing damage" [Cor]. Throughout history, there are several records of major fraudulent events, such as bankruptcy, credit card or healthcare fraud, among others, that cause damage in a great variety of domains.
The Fraud Triangle is a model that tries to explain why people commit fraud, consisting in three elements: motives, which for instance can be the lack of money; rationalization, that is the ability to justify the crime as acceptable given the circumstances; and opportunity, which gives the fraudster the possibility to act in secret [Cre73].
Moreover, fraud can be distinguished in soft and hard fraud. The first one refers to an exaggeration of the damages reported on one legitimate claim, or when an individual lies about details or conditions in order to lower the policy's premium or undercover important facts about the claim. In contrast, the hard fraud, instead of deriving from an occasional opportunity, refers to an intentional scheme designed by an individual or, most commonly, a group of individuals, with the objective of collecting the value related to the insurance policy.
One of the main factors that makes fraud so hard to detect is its dynamic nature. Fraudsters tend to break the law in uncountable imaginative ways, following changes in business norms and adapting to legislation in place.
To fight fraud at an european level and protect the financial interests of the European Union (EU), the EU created in 1999 the OLAF, European Anti-Fraud Office. This organism can coordinate and investigate fraud all over EU, as well as cooperate on the development and implementation of anti-fraud legislation, being a very visible sign of how engaged the EU is on this matter over this century.
Regarding the law framework in Portugal, insurance fraud is specifically contemplated on the Penal Code as a crime entitled "swindle against patrimony", punished with fine or imprisonment, depending on the amounts of money involved [Rep13].
Technology has become a very important piece on the investigation of such crimes. With the huge amount of data available to analyse, analytics software systems are the most reasonable way to start looking for suspicious activities. Recently, social network analysis have been applied to many fields with success on clarifying how people interact, bringing new insights to a broad range of problems, including to fraud detection [BCGJ11].
This dissertation addresses the issue of hard fraud on automobile insurance, which consists in any "duplicitous act performed with the intent to obtain an improper payment from an insurer" [Cor]. The present work was developed on enterprise environment, under a protocol established by UBI and Deloitte Portugal, having the co-supervsion of two Deloitte's professionals besides the professor at UBI.

1.1 Motivation and Objectives

The insurance industry plays a relevant role on national and international economy panorama. According to the Insurance Europe, the european insurance and reinsurance federation, european

insurers generate premium income of almost 1100€ billion[1], employ nearly 1 million people and invest around 7700€ billion in the economy [Ins13].
A 2013 report from the Portuguese Association of Insurers (*Associação Portuguesa de Seguradoras* APS) estimates 2012's net income in 539€ millions [dS13]. The same report states that in 2012 the total cost borne by policyholders having insurance contracts in the Portuguese market is 11.4€ billion, in contrast with the 11.9€ billions returned to the society. This fact, allied with the finance and economic situation installed, leverages fraud detection as a great tool to reduce costs and minimize this gap.
It is estimated that 10% of all claims expenditure in Europe is related with fraud, either detected or undetected. To contextualize in numbers, the Association of British Insurers (ABI) states that undiscovered fraud represents 2.2€ billion each year, despite the amount of fraud detected from 2010 compared to 2011 increased by 7%. Also, data from Insurance Federation of France related to the year of 2011 shows records of 35042 fraudulent claims, equivalent to 168€ millions which would be lost to fraudsters. In Germany, the fraud value per year goes up to 4€ billions [Ins13].
Despite the lack of official statistics on fraud in Portugal, there are some studies that point to the raising of fraudulent events in the last years. For instance, Liberty Seguros, that operates in Portugal, revealed that fraud duplicated from 2007 to 2012. Going deeper on the matter, and reporting only to the automobile insurance branch, there was an increase of 112% on fraudulent sinister comparing data from 2007 with 2011 [+].
Analysing carefully these numbers, it can be conclude that this increase can be due to the fact that insurance companies are putting more effort on discovering fraud. Yet, another explanation emerges within the community, stating that this phenomenon is also related to the fragile financial and economical situation the country is experiencing, which drives people to have risky attitudes to win a little more money [+].
Insurance companies are increasingly aware of the damage fraud brings to their business, therefore they have been particularly engaged in investigate fraud with more sophisticated techniques. Reducing fraud impact on insurance business allows companies to reduce insurance bill to the customer as well. For instance, it is estimated that in the United Kingdom, fraud is responsible for an additional 58€/year on every policy, on average. On top of all this facts, identifying suspicious activity with more efficiency allows companies not to loose time with genuine claims, improving the quality of service and client satisfaction.
It is clear that the fraud problem needs to be address in a more structural way, through education and actions that can increase the awareness to this problem, with the final objective being the change of customers' mindset, ethics and honesty, acting on the rationalization vertex of the fraud triangle. However, to act on the opportunity vertex, providing tools that allow to identify risks in a short time period is fundamental for the health of the business and can be extremely helpful on the overall fight against fraud.
Given all these statistics, showing the relevance of insurance on the economy and the impact of fraud, and due to the fact that, to a large extent, this is a still an unsolved problem and a relatively little explored field on computer science, the opportunity and pertinence to study this subject seems evident not only to the Deloitte Portugal, but also from a research perspective.

1.2 Document Organization

The rest of this document is organized in as follows:

[1] A billion is equivalent to 1000 million.

- Chapter 2 - State of the art - This chapter addresses issues related to fraud detection, outlines the most common methods proposed on the literature to handle fraud detection in a broad range of domains. It also details some aspects related to social network analysis (SNA), gives some historic perspective and describes some selected methods particularly focused on insurance fraud detection using SNA. Finally, it also gives a market overview on the available solutions that could be used for this purpose.
- Chapter 3 - Proposed system - This chapter describes in detail all the phases of the proposed method.
- Chapter 4 - Experiments and results - This chapter describes the dataset, the experiments conducted with the approaches suggested before, compares performances and discusses results.
- Chapter 5 - Conclusions - This chapter outlines the conclusions of the present work, and outlines possible future work.

Chapter 2

State of the art

This chapter begins with a section dedicated to introduce fraud detection and provide a general overview of the most important data mining techniques used for this purpose, based on the Phua *et al.* survey [PLSG10] and a more specific survey on automobile insurance fraud detection from Viaene *et al.* [VDBD02]. After this, follows a section on social network analysis, its associated techniques and applications. Finally, it presents a review of the most relevant literature on social network analysis applied specifically to fraud detection.

2.1 Fraud Detection

With the increase of business environment complexity, and the need of cost savings, the ability to identify fraud as early as possible becomes a very important tool among companies from all over industries. To check manually all risk factors and all fraud indicators for every process, is not either cost nor time efficient, justifying the application of automated or semi-automated techniques for fraud detection.

On 1996, on a historical paper, Fayyad *et al.* [FPSSU96] already pointed fraud detection (on credit card monitoring and suspicious transactions) as one of the main Knowledge Discovery in Databases (KDD) application areas, along with marketing, finance, manufacturing, telecommunications and internet agents. In this paper they define data mining as the "application of specific algorithms for extracting patterns from data", a step of KDD, a major process of "discovering useful knowledge from data". There are other steps that must be considered on KDD, such as data preparation, data cleaning, data selection, visualization and interpretation of results.

Data mining techniques have been applied to fraud detection in many industries, being the most relevant the telecommunication sector, financial services (credit card transactions, insurance claims and money laundering), retail and government issues (escape tax, terrorism). It is not the scope of this section to describe the foundations of every data mining technique used in fraud detection, therefore, for a precise and complete explanation of each method please see [B$^+$06].

Phua *et al.* [PLSG10] analysed literature related to data mining applied to fraud detection and categorised methods and techniques from the year 2000 to 2010. They categorise 51 papers on an industry level, on four main groups, internal, insurance, credit card and telecommunications fraud detection, and in some subgroups, respectively: management and employee fraud; home, crop, automobile and medical insurance; credit applications and credit transactional; and finally, telecommunication subscriptions and telecommunication superimposed. They reference 6 articles on automobile fraud insurance detection, far less than the most referenced category with 15 references, credit transactional fraud.

The survey issues the number of attributes and examples of datasets used on 40 fraud detection papers on internal, insurance, credit card and telecommunications. There are 11 insurance datasets identified, the largest with 4000 examples and the fewest attributes (about 6), and the smallest with less than 100 examples and about 70 attributes, which is the maximum for this category. Compared to these, telecommunications and credit card transactions datasets are much

bigger, many containing millions of records. Automobile insurance datasets' attributes consist mostly on binary indicators concerning accident, claimant, driver, injury, vehicle, etc. Regarding the percentage of fraud and test examples in each study, 16 of the 19 papers considered for this analysis have skewed data with less than 30% of fraud.

Almost all data sets related to internal and insurance fraud lack of temporal information. Also, almost all data sets have been de-identified. Three of the 51 papers referenced by Phua *et al.* used simulated data, vainly, because results were not realistic or not explained. It is extremely difficult to find public data sets for fraud detection, in fact, the authors claim that there is only one small data set for automobile insurance.

Phua *et al.* also addressed the problem of performance measures. The research says some authors use Receiver Operating Characteristic (ROC) analysis, and only one focused on maximise Area under the ROC (AUC) and minimize cross entropy, the same that proposed to minimize Brier score. There are also authors that propose using one metric from threshold, ordering and probability metrics to evaluate supervised algorithms, and Activity Monitoring Operating Characteristic specifically for timely credit transactional and telecommunications superimposition fraud detection. Regarding semi-supervised algorithms, some recommend entropy and a few of its variations, information gain and information cost. Hellinger and logarithmic scores have been used by two authors on unsupervised approaches, as well as some other statistic measures to detect outliers, like *t*-statistic score. Insurance fraud detection is also commonly evaluated by comparing results with the opinion of domain experts.

Then, the authors continue to discuss the four major methods' categories: supervised approaches on labelled data, hybrid approaches with labelled data, semi-supervised approaches with non-fraud data and unsupervised approaches with unlabelled data. Kou *et al.* also surveyed techniques on data mining, presenting the most relevant techniques regarding the application, specifically, for credit card, telecommunications and intrusion detection [KLSH04]. Bolton & Hand [BH02] describe the statistical and machine learning methods most commonly used in fraud detection, covering a wide range of supervised and unsupervised algorithms.

The first category Phua *et al.* address uses all labelled data available to create a model and predict whether new instances are fraudulent or legal. Neural networks are among the most popular techniques, although others have also been used. Regarding neural networks, for instance, Ghosh & Reilly (1994) [GR94] trained a three-layer feed-forward Radial Basis Function neural network on both fraud and non-fraud accounts, using a set of 20 features per transaction. They sampled the data set, consisting on transactions of six months of 1991, so to have a ratio of 30 non-fraud accounts to each fraud account, ending up with a training set with about 450000 transactions. Due to its characteristics the network only needs two trainings passes through the data set, and then produced a fraud score for each transaction.

The authors evaluated the model on an unsampled set of all transactions of October and November of 1991 (not coincident and posterior with training data period), corresponding to roughly 2000000 transactions. The first measure they present is a rank curve of the percentage of fraudulent transactions detected against the number of accounts flagged for review per day. The authors claim that if the system flag near 50 accounts per day, 40% of them would be fraudulent, which is a major improvement to the fraud detection efforts before that consisted in analysing 750 accounts to spot only one fraudulent instance. Ghosh & Reilly also measured the earliness of the detection, presenting a histogram that indicates that 50% of detected accounts are on the first two days of fraud activity, with a fraud threshold operating point of 1-fraud-per-day. Also, they measured the type of fraud detected regarding the several categories related to credit card fraud, such as lost cards, stolen cards, application, counterfeit, mail-order and non-received issue fraud, concluding

the system was detecting all fraud categories. This system was implemented on a bank, enabling the entity to achieve savings of 20% to 40%, at a reduced operating point human review.
Even though the aim of Barse *et al.* (2003) [BKJ03] was to prove the usability and applicability of synthetic data for fraud detection, it is relevant to point that they also used successfully a feed forward neural network with one hidden layer and an additional trace memory to deal with temporal dependencies in an IP based video on-demand service. Weatherford [Wea02] also discusses the use of neural networks, giving some market solutions as examples, as well as artificial immune systems.
Ezawa & Norton (1996) [EN96] built a model based on Bayesian networks in order to predict whether a telecommunications' customer account or transaction is collectible. They focus not only on the performance of the solution but also on the efficiency, due to the huge quantity of data commonly associated to telecommunications data sets, namely in this experiment from four to six million of records, each with more than 30 variables. The authors present some features inherent to the problem that motivate the choice of Bayesian networks instead of regression systems, nearest-neighbour systems and neural networks, such as data size and dimension, unequal misclassification costs, and the probabilistic nature.
They evaluate the proposed system with four different probability models, two dependent and two independent, and used the Receiver Operating Characteristic curve to analyse results. Finally, Ezawa & Norton concluded the dependent models perform better than the independent, and discussed the balance between the probability threshold of collectible / uncollectible accounts and the true positives / false positives ratio.
Viaene *et al.* [VDD04] focus on the detection of suspicious personal injury protection automobile insurance claims, using a weight of evidence reformulation of AdaBoosted naive Bayes scoring. They evaluate the model concerning to the discriminatory power, ranking ability, and calibration of probability estimates. Thus, the authors use the percentage correctly classified measure to assess discriminatory power, receiver operating characteristic (ROC) analysis and area under the ROC curve to assess ranking ability, the logarithmic score to assess the quality of probability estimates, Brier score to asses both the quality of probability estimates and the discriminatory power, and finally they use Brier score on the calibration plot to assess again the quality of the model calibration. The data set consists on 1400 closed claims each with 48 binary indicators, previously investigated bu expert domains and assess using a 10-point-scale score wherein a score greater than 3 indicates that it should "not pass" without further investigation. In this data set, the "no pass" category represents 28% of all claims. For evaluation, the authors split the data set in 2/3 for training and 1/3 for testing, and eventually conclude the weight of evidence reformulation of AdaBoosted naive Bayes scoring to be the best among the models tested.
In [MTVM02], Maes *et al.* discuss the use of neural and Bayesian networks in credit card fraud detection. They use a feed-forward multi-layer perceptron with the backpropagation of error signal algorithm and for the Bayesian network they use the STAGE algorithm, described in the paper. The authors use ROC to measure the performance of the methods. Comparing both approaches, the Bayesian network outperforms the neural network for 8% maximum with 74% true positive at a cost of 15% false positives, and regarding training times, while neural networks may take several hours the Bayesian network only takes twenty minutes, although evaluating new instances is completely the opposite, with the neural networks being much faster.
One way Kim *et al.* found to evaluate the proposed support vector machine (SVM) ensemble on [KPJ+03] was to apply it to a fraud detection problem, specifically on this case, mobile telecommunication payment. They performed binary and multi-class classification, concluding that SVM ensemble outperforms single SVM, that the particular type which implements majority voting ag-

gregation using polynomial kernel is the best for this application, with correct classification rates of above 95%, although not presenting false positives or other measures, and that multi-class is better according to this metrics than binary classification.

Fan (2004) [Fan04] proposed a cross-validation decision tree ensemble method to compare different scenarios related to the problem of combining old and new chunks of data to mine concept-drifting data streams. Although the purpose of this experiment was not to evaluate the performance of the fraud detection method, they used a credit card fraud data with 5 million of transactions to assess the model performance. The authors note that it is not wise to use old data blindly, and conclude that the cross-validation decision tree ensemble consistently outperforms all compared existing approaches that use old data blindly, particularly when the chunks of new data are small.

Wang *et al.* also addresses the problem of mining data streams with concept drifts, and proposes a general framework for mining concept-drifting data streams using weighted ensemble classifiers. In [WFYH03], the authors apply successfully their method to the same data set used by Fan (2004), and discuss the advantages of classifier ensembles over single model classifier.

In [RMN+99] Rosset *et al.* present a two-stage system based on adaptation of the C4.5 rule generator with an additional rule selection mechanism, design to cope with the unique features of rule-discovery for fraud analysis. The authors describe what makes the traditional algorithms and methods of classification and rule-discovery fail when applied to fraud, such as: the existence of, at least, two data levels (customer details and behaviour details); the requirements of good rules and how they have to be accurate at the customer level, sensitive related to the coverage of true positives cases and coverage of true positive alerts; and requirements of good rule-sets, namely how to choose the right rule-sets in order to maximize the three previous requirements. The data set used in this experiment is related to telecommunications transactions, and consisted on a few hundred "legitimate" customers and a few hundred bad debt customers. They run the standard C4.5 rule generator and the proposed "bi-level compliant" version of the C4.5 engine, and verified that the latter returned much more interesting patterns than the first, even though they did not disclose the rules because of confidentiality issues. The authors used 4 measures of performance: set size, accuracy, fraud coverage and maximum correlation.

Bonchi *et al.* (1999) [BGMP99] expose a case study on fiscal fraud detection that illustrates how classification-based techniques can help to plan audit strategies, *i.e. a posteriori* fraud detection. In this experiment, the autors also used a variant of C4.5 algorithm, in this case, the C5.0, and evaluate the model with 2 domain-independent metrics (confusion matrix and misclassification rate) and 4 domain-dependent related to audit costs, money recovery, profitability and relevance.

Wheeler & Aitken (2000) [WA00] addressed the problem of reducing the number of cases flagged for investigation by existing systems, in the credit approval process, using a Case-Based Reasoning approach. The data set consists in pairs of records, one related to the application to be investigated and the other to the evidence that justified the decision. A weight matrix and nearest neighbour algorithm were used on the retrieval component, and the diagnosis was composed by a set of algorithms such as probabilistic curve, best match, negative selection, density selection and default goal. The authors argue that this multi-algorithmic approach may outperform the use of isolated algorithms.

Belhadji *et al.* (2000)[BDT00] developed a model for the detection of insurance fraud related to property damages in the automobile sector, based on the systematic use of fraud indicators. The authors choose the attributes that best indicate the existence of fraud on three steps: first with the help of domain experts of several insurers, then calculate the conditional probabilities for each indicator in order to reduce the number of indicators to include in the model, and finally with

Probit regressions. The authors then calculate the probability of fraud for each file in the data set, discuss the threshold to apply and the costs of further investigation on suspicious cases.
The combination of some of the previous supervised techniques referred above provide more sophisticated approaches to this problem. For instances, Chan *et al* (1999) [CFPS99] proposes a credit card fraud detection system, with great focus on scalability and efficiency. They apply mining techniques to generate classifiers in parallel and then combine the base models (Bayes, C4.5, CART and RIPPER) generating a metaclassifier. Their approach have the limitation of being necessary to run preliminary experiments to determine the desired training distribution.
Phua *et al.* (2004) [PAL04] combined backpropagation neural networks, naive Bayes and C4.5 algorithms as base classifiers. The authors used a stacking-bagging approach, *i.e.*, they used a single meta-classifier to choose the best base classifiers, and then combined the predictions of base classifiers. This model was evaluated on a public automobile insurance fraud detection data set, and the authors claim it outperforms the best bagged algorithm and the best classifier, as so the common technique used in the industry which was, at the time, backpropagation neural networks.
Ormerod *et al.* (2003) [OMB+03] propose a tool that uses a Bayesian network of fraud indicators, which dynamically adapts its weights according to a rule generator, considering how predictive each indicator is of specific types of fraud. The authors do not present any comparison or evaluation experiments. Kim & Kim [KK02] focus on the bias of the training set, due to the nature of fraud. They analyse the fraud density and work it along with a backpropagation neural network to calculate a weighted suspicion score on credit card transactions.
Also, some authors suggest the combination of supervised with unsupervised algorithms, mostly on telecommunications fraud detection [PLSG10]. In 1997, Fawcett & Provost [FP97] proposed a fraud detection system that would learn rules from labelled fraudulent behaviour and use the indicators to monitor daily usage and find anomalies for each customer. A Linear Threshold Unit learns the output of these monitors (set of indicators), and generates high confidence alarms. The authors present comparisons with another fraud detection strategies and several detectors, claim to outperform state-of-the-art at the time, and present other advantages such as the adaptability and flexibility compared to others.
Among other usage of unsupervised algorithms is the segmentation of insurance data into clusters for supervised methods to analyse. Williams & Huang (1997) [WH97] focus on finding hot spots (clusters) in very large real world databases, adopting a multi-strategy approach comprised of three steps: cluster detection using k-means, description of clusters using C4.5 rule generator, and rule evaluation using statistics, visualization tools and ultimately domain knowledge. This method was applied to a healthcare data set in order to identify hot spots of fraudulent payments related to the government healthcare program. The same author followed a similar methodology with some tweaks, for instances, using a genetic algorithm instead of C4.5 rule generator [Wil99]. Likewise, Brockett *et al* [BXD98] use Self Organising Maps for clustering and then backpropagation neural networks to identify fraudulent claims.
Fraud detection can be addressed as an anomaly detection problem, if we can extract features from all legitimate records, claims or other data and identify when a pattern does not fit the normal behaviour expected. In [CBK09], Chandola *et al.* give a comprehensive view on anomaly detection techniques, analysing both more general methods and more specific application-dependent methods, namely related to fraud detection. They cite some examples on credit card fraud detection, most of them using neural networks and some using rule-based systems and clustering algorithms. Statistical profiling methods and parametric statistical modelling have been used for mobile phone, insider trading and medical healthcare fraud detection, as well as rule-base sys-

tems and neural networks. Regarding the insurance domain, Chandola *et al.* say this problem is often addressed as a activity monitoring problem as in [FP99], but approaches using neural networks to identify anomalies can also be found.

Moreau *et al.* [MLV+99] studied fraud detection on mobile communications and proposed BRUTUS, which is a hybrid detection tool based on rule generation methods and neural networks that enable the profiling of both network subscribers and traffic. They compared rule-based and supervised neural networks systems with unsupervised neural networks, and, as expected, the supervised approaches reached better results than the unsupervised, although the best results came from a hybrid model. In this sense, Taniguchi *et al* [THHT98] also reported better results using supervised neural and Bayesian networks than unsupervised Gaussian mixture models, and also, that the combination of supervised and unsupervised techniques could produce better results.

However, fraud detection systems applied in real world face some particular challenges that makes it a problem so hard to solve and so interesting. Specially, developing methods to detect fraud which use labelled data assumes that this data is going to be available and thus would not compromise the performance of the system. In reality, labelled data is extremely hard to find, expensive to obtain as it demands a lot of manual work of domain experts, and even more difficult to have in the amount necessary to effectively train some models. Phua *et al.* [PLSG10] suggest that the future of fraud detection research can benefit from unsupervised approaches already proposed in related fields such in anti-terrorism, law enforcement, intrusion and spam detection, etc. These reasons justify the need for unsupervised methods using unlabelled data, which give an extra challenging nature to the already difficult fraud detection task, but also can provide the flexibility and adaptability this task demands.

Bolton & Hand focus on detecting behavioural fraud through the analysis of credit card transactions over time [BH+01]. They present two unsupervised methods to detect behaviour changes. The first tool is Peer Group Analysis, a method based on the ability to recognize when an account starts to behave differently from other accounts which usually used to behave alike. The method summarize each account behaviour pattern, chooses the peer group accounts of each account and monitors the behaviour of each peer group to spot if an account from a peer group is beginning to stand out, based on a *t*-statistic analysis to calculate the distance to the peer group centroid. The author also propose a method called Break Point Analysis, which focus on spending behaviour. This tool focus on intra-account behaviour in order to detect rapid spending, based on the comparison of recent transactions with previous transactions, within a time-window of 24 hours.

Brockett *et al.* introduce a mathematical technique for fraud classification that does not require training data [BDG+02]. The authors used principal component analysis of RIDIT scores for ranking automobile insurance claims, providing measures of both the ordered categorical attributes of a claim and the claim file overall score. Hollmén & Tresp present a call-based fraud detection using a hierarchical regime-switching model [HT99]. This real-time method models a hierarchical structure, at the lowest level representing the behaviour of individual calls, next level the switching behaviour from normal to fraudulent, and at the highest the transition to being "victimized" by a fraudster. Hidden Markov Models have been used to model time series at different time resolutions. The authors used ROC curves to measure the peformance of the model, and claim to have detection probabilities of 0.92 for a fixed false alarm probability.

More related to the scope of this dissertation, Phua *et al.* pointed link analysis and graph mining as hot research topics little applied to fraud detection, but already applied for anti-terrorism, for instance. In this issue, [CPV02] addressed telecommunications fraud detection, and propose a data structure able to handle dynamism through time, with appearing and disappearing edges

and nodes. Although there are some market solutions, discussed later on this chapter, using visualization tools to help domain experts spot fraud rings, there are little work on this topic among the research community. Back in 1997, Cox *et al.* [CEWB97] proposes a suite of visual interfaces with the intent to combine human pattern recognition capabilities with computational capacity for telephone fraud detection.

Fraud detection methods can be inspired by similar fields, namely terrorism, financial crime and intrusion and spam detection [PLSG10]. Bayesian networks have been applied to detect simulated anthrax attacks, and other techniques like sliding linear regression or hidden Markov models have been used to detect other epidemics. The United States government use some tools to detect suspicious activities such us money laundering, violative trading and insider trading activities, based on some techniques already mentioned, including Bayesian inference engines, link analysis for visualization purposes, case based reasoning, nearest neighbour retrieval, decision trees, association rules, text mining, statistical regression and fuzzy matching. Zhang *et al.* [ZSY03] proposed a technique to uncover money laundering by analysing documents and suggest links between them to generate community models, using a correlation measure. Also Donoho [Don04] compares C4.5 decision tree algorithm, backwards stepwise logistic regression and neural networks with a manually built expert system for early detection of insider trading in option markets. C4.5 algorithm outperformed the rest of the methods, and the expert system produced the worst results. Regarding spam and intrusion detection, recent research focus on the branch of anomaly detection methods, semi-supervised and unsupervised techniques, thus some algorithms could have some application on the fraud detection problem.

2.2 Social Network Analysis

This section is going to approach the state of the art on SNA, first by giving an historic perspective, followed by the description of some relevant applications on the industry, then providing some definitions and then surveying some of the most common methods and techniques.

2.2.1 Historical perspective

Social Network Analysis (SNA) has captured the interest of many researchers from a broad spectrum of scientific disciplines as anthropology, economics, geography, biology, marketing, and several more, but its foundation came from both psychology and sociology. In the 30's, Jacob Moreno developed sociometry to investigate friendship relations, founded a journal entitled *Sociometry* and invented the sociogram, which is a diagram of points and lines representing relations between persons. In the 50's, the department of sociology Manchester University started to investigate the cohesiveness property of society, inspiring later on the 60's and 70's a Harvard group to develop a mathematical formulation of many social sciences concepts. [SC11]

One of the most famous (and early) SNA studies is the Travers and Milgram experiment on the late 60's, associated with the expression "six degrees of separation", in which the researchers wanted to analyse the social structure of USA population and extrapolate the results to a world scale [TM69]. The procedure to do social networks experiments by these times was incredibly different from what it is nowadays, not only respecting the application domains but also to the size of data, the nature of data itself, how it is collected, how it is processed, and so on.

Furthermore, the mathematical foundations of social network analysis can be traced throughout the history of graph theory in discrete mathematics, beginning in 1736 with the work of Leonard

Euler. Later on with the analysis of all kinds of networks (computer, biological, financial, medical, transportation, and others) [BCGJ11], these methods have been applied in so many distinct contexts that we can group the different experiments on the major label of *network science*.
One of the most relevant books on SNA until today was published on the last decade of the 20th century [Was94], resuming in a comprehensive way the methods and applications developed to the time. Also, the major version UCINET IV was released on 1992, consisting on a suite of software programs for the analysis of social network data, roughly at the same decade Pajek started to be implemented, another well-known software package for SNA.
More recently, SNA is having a great impact due to the powerful insights on how society is structured and its behaviours, using data from online social networks (Facebook, Twitter, LinkedIn, *etc.*), mostly for marketing purposes, but not only. These platforms provide a massive source of data enabling not only to address old problems in novel ways, but also to pose new ones, regarding the type of data they collect and to the nature of relations they are able to describe.
Bonchi *et al.* [BCGJ11] say the research on social network analysis was conducted not from a business application viewpoint, but with a more theoretical approach, trying to answer common problems related to the structure of networks (its properties), its evolution along time, or how information propagates inside a network, but without relating these problems with a specific application.

2.2.2 Applications

Social network mining research has been approached from a very generic point of view, sometimes making it hard to claim their space on the market applications, perhaps with the exception of telecommunications sector. However, there are many potential applications for SNA, besides fraud detection and the sociological problems, which are surveyed in this section. In this issue [BCGJ11], Bonchi *et al.* outline some applications for SNA categorized in operating processes and management and support processes, some of them referred on the next paragraphs.
Regarding fraud detection, and besides the solutions presented on the section 2.3 and the references related to unsupervised methods above, there are a few relevant application studies, for instances, in [VD06] the authors investigate camouflaged fraud in complex domains with high number of known relationships. Neville *et al.* [NŞJ+05] focus on exploit several sources of information, considering relationships among multiple entities on their statistical relational learning algorithms. More on this topic to be addressed on the section 2.2.4.
SNA is mostly used for marketing and advertising purposes across industries, and particularly on online environment. For this goal, calculating reputation and trust, and identifying communities of interest becomes major issues, with an infinity of contexts where these techniques can be applied. For instances, calculating individuals reputation on a recommendation system or on a similar environment can be extremely useful to power the usability of a e-commerce website like eBay, to manage products revisions and score reviewers [MA02], helping the community to regulate itself. Finding experts on some matter on Q&A forums or within the corporation human resources has been also studied [YSK03].
On marketing as well, SNA is being applied to build suggestion friends systems by implicit link analysis and link prediction [SBC+10], to trend spotting by analysing customers data [NPS08] and understand what will be their needs, to identify interesting communities for customer loyalty reward programs [HPV06], to evaluate how communities evolve through time, and to understand how information, knowledge spread along the network [HLL+07] [GGLNT04], to give a few examples. Lately, there has been a pike of interest in influence and its propagation along the network

in order to understand viral marketing [Wor08], comprising tasks such as selecting the key users in a network to start a campaign, track how influential users affect the opinion of its followers and how it affects the networks, etc.

Also, SNA can change the way corporations work, on the inside by inspiring new strategies regarding how they manage their knowledge (expert finding) [DYB+07] [PE06], how they assemble teams, how the employees work collaboratively and communicate to each other, and on the outside, by leveraging communication to its partners and to its customers on a more effective and direct way. Also, the way corporations launch their marketing campaigns, by building systems to monitor customers and keep track of products' reactions, identify customers communities, understand how their customers network evolve over time, and other business intelligence golden insights. Another interesting application of SNA is the analysis of how epidemics spread and to understand the behaviour of such health phenomena [PSV01].

This vast range of applications arise privacy issues in what respects to the acquisition, preparation, manipulation and storage of sensitive data. A new paradigm of privacy-preserving data mining was proposed to deal with this question [AS00]. Particularly in SNA, not only the information attached to the nodes and edges of the network is sensitive, but also in some situations, the structure of the network can be significant, and therefore needs to be hidden [BDK07]. In this sense, there have been many suggested approaches. Anonymization of graphs and social networks is discussed in [BCGJ11], which categorizes methods of identity obfuscation in three categories. The first has to do with deterministic edges deletion or addition, providing a k-anonymity. The second is not deterministic, and is based on random additions, deletions and switching edges operations, and the third, instead of changing the appearance of the network, group the nodes into supernodes and protect the data in the sense that it changes the resolution of the network.

2.2.3 Mathematical foundations and general concepts

Social network analysis, as seen before, emerged from a variety of science fields, what makes it hard to define conceptually, however it can be described as a set of methods for the analysis of social networks, "based on the assumption of the importance of relationships among interacting units" [Was94] [Sco88]. A social network can be informally defined as a set of social entities and the relations among them, being this feature of relational attribute the fundamental aspect that defines a social network. A social relation is a dyadic attribute, meaning that it is an attribute related to a pair of entities, *i.e.*: kinship (brother of, mother of), social roles (married to, boss of), affective, actions (attacks, communicates), distance, etc.

Formally, a network is commonly represented as a graph, which is an object very well studied on mathematics and is at the foundations of what is called graph theory, that inspired SNA on several of its techniques. A graph is usually defined as a set of points (vertices or nodes), and a set of edges or links defined by a pair of vertices. This definition does not define a direction for edges, thus representing an undirected graph, whereas a directed graph impose that edges are defined by ordered pairs of vertices. Multigraphs are a spare relaxation of these definitions, allowing loops and multiple edges. Also, it is regularly used labels to annotate some information on both vertices and edges, originating labelled graphs. It is often assigned to edges a weight, representing the strength of the relation between those entities. In that case, the graph is called a weighted graph. Another relevant concept is the one of connected component, which refers to a subgraph where any two points have a path to each other. For an extended approach to network theory, please see [New03].

2.2.3.1 Link Mining

Link mining stands at the intersection of data mining and network analysis, involving tasks like object ranking, community detection, collective classification, link prediction and subgraph discovery. In fact, Getoor & Diehl [GD05] suggest a taxonomy on link mining that highlights 8 tasks, some of them covered in this section:

- Object-Related Tasks

  Link-Based Object Ranking (LBR)

  Link-Based Object Classification (LBC)

  Object Clustering (Group Detection)

  Object Identification (Entity Resolution)

- Link-Related Tasks

  Link Prediction

- Graph-Related Tasks

  Subgraph Discovery

  Graph Classification

  Generative Models for Graphs

Regarding SNA, LBR represents the task of measure the importance of an individual in the network, which can be addressed through the use of centrality measures.

- Degree Centrality [Fre79] - Local measure calculated by counting the number of links of an actor to actors directly adjacent to it. High degree centrality values indicates this entity is strongly connected to others and possibly is an influential actor. Degree centrality C_D of an actor v_i can be calculated as:

  $$C_D(v_i) = \sum_{j=1}^{n} a(v_i, v_j)$$

  where n is the total number of actors in the network and

  $$a(v_i, v_j) = \begin{cases} 1 & \text{if } v_i \text{ and } v_j \text{ are connected by an edge,} \\ 0 & \text{otherwise.} \end{cases}$$

  A generalization of this measure is the k-path centrality, that counts the number of paths of length k starting from a given node.

- Closeness Centrality [Fre79] - Measures how close an actor is to all other actors in the network. This means that small values indicates that the actor is well located in the network, close to everybody. Closeness centrality C_C of an actor v_i can be calculated as:

  $$C_C(v_i) = \frac{n-1}{\sum_{j=1}^{n} d(v_i, v_j)}$$

  where n is the total number of actors in the network and $d(v_i, v_j)$ is the geodesic distance between the two actors. The geodesic distance is the number of edges between two nodes in a shortest path.

- Betweenness Centrality [Fre79] - This measure indicates the number of times an actor stands in the way of a path to another actor in the network, or it can also be seen as the probability an actor is included in the communication between two other. This means that actor with high centrality usually work as bridges to promote and simplify the flow of information within the distinct parts of the network. Betweenness centrality C_B of an actor v_i can be calculated as:

$$C_B(v_i) = \sum_{j,k}^{n} \frac{g_{jik}}{g_{jk}} \quad , \quad i \neq j \neq k$$

  where n is the total number of actors in the network, g_{jk} is the number of geodesic paths from actor j to k and g_{jik} is the number of those geodesic paths that pass through actor i.

- Eigenvector Centrality [Bon72] - This measurement indicates how central the node is regarding the whole structure of the network, in other words, how a node is connected to other well-connected nodes in the network. It is computed by taking the principal eigenvector of the adjacency matrix.

There are several variations based on this approaches, proposed for dynamic graphs, for measuring centrality relative to other objects, and also several different algorithms to calculate those [ST11].
The are also some measures based on edges, instead of nodes:

- Tie Strength [Gra73] - This measure reflects the embeddedness of an edge, this means, if two incident nodes on a given edge have a high overlap of neighbourhoods. For instances, if two actors A and B connected through a given edge have many neighbour nodes n_A, n_B in common, they're said to have a strong tie. Formally, using the Jaccard coefficient:

$$S(A, B) = \frac{|n_A \cap n_B|}{|n_A \cup n_B|}$$

- Edge Betweenness - Similar to the betweenness centrality presented before, this measure has a particular use in graph partitioning, where edges with high betweenness are iteratively removed until some point, creating disconnected components. It represents the number of pairs of nodes for which a given edge belongs to the geodesic path.

The second object-related task indicated is LBC, which in the context of this work, can be useful to classify a node as fraudulent or not. Using LBC, there is a primary assumption that labels of related objects (which would be classified the same category) tend to be correlated, a fact that these algorithms for collective classification need to take advantage of. There are several approaches to this problem, with multiple applications not only on SNA but also on fields like computer vision and natural language processing.
A curious outcome of a couple of experiments in LBC is that using class labels of related objects improves classification performance, but including its features may have a negative effect [CDR+98] [OML00].
Moving to the next task, group detection aims to identify clusters of nodes that resemble the same features. There are several approaches to this task described in the literature, and for a deep review of this topic it is recommended to see Fortunato [For10]. Methods are mainly divided in agglomerative or divisive approaches and deterministic or stochastic. The edge betweenness [Fre79] measure, already referred before, was used in [GN02] by Girvan and Newman to rank edges and build a top-down approach of the classic hierarchical algorithm for community detection. Instead of starting with all nodes disjointed and iteratively construct communities in respect

to some similarity measure, they started with all nodes belonging to one community and then started to remove edges, thus naturally creating communities, ending up with a hierarchical subgraph partitioning algorithm. Girvan and Newman further proposed the modularity measure to evaluate what is the best hierarchical subgraph. Another well-known family of community detection algorithms, based on graph theory, is graph-partitioning algorithms, focused on the concepts of minimum cut and maximum flow [FTZ04] and spectral partitioning which involves using standard clustering algorithms on point in Euclidean space [Llo82].
Object identification as known as entity resolution, consists on identifying duplicates of instances on the graph, this is, if there are multiple nodes referring to the same entity. Link prediction focus on discovering links that are not explicitly defined on the graph, thus uncovering hidden connections. On the graph-related tasks, subgraph discovery focus on discovering frequent substructures in a set of graphs, graph classification is the task of classify an entire graph as instance of a given concept, and generative models for graphs focus on the generation of new graphs based on a set of features previously defined or assessed from another graph. All these latter tasks could be interestingly applied to fraud detection. Although it is not the scope of this thesis, some aspects are further discussed on section 5.1.
Depending on the context of link mining, and specially applied to fraud detection, it is very common to find data sets with skewed data, that is, with many records of honest transactions (or whatever it represents), and little few cases of fraudulent transactions. This problem can be addressed at the time of pre-processing data, with either under-sampling the majority class or over-sampling the minority class, or using different weights on the training set distribution in order to balance in some way the data set, through boosting algorithms [KN06] [Alm09].

2.2.4 Related Work

2.2.4.1 Subelj, L. *et al.* method

In [ŠFB11] Subelj *et al.* proposed an expert system for detection of groups of automobile insurance fraudster, the so called non-opportunistic fraud. This approach uses networks to represent data, justified with the fact that collaborating fraudsters are commonly related to each other, and since networks are essentially relations between entities, it is appropriate to detect groups of fraudsters, besides the cle ar visualization necessary for the following stages of investigation.
The authors also propose a novel algorithm to find fraudulent entities within this networks, considering intrinsic attributes and relations between entities. The algorithm allows the incorporation of domain knowledge, which the authors say is useful to adapt it to new types of fraud. Collision networks, which consist in individuals and vehicles, are used to assign to each entity a suspicion score.
The system frameworks consists in four modules: the networks construction from the data set; the highlight of suspicious connected components on the networks of first module, considering structural properties; the third module consists on assigning a suspicion score to each entity of each suspicious component, using the novel proposed algorithm; and finally, in the fourth module the system provides a clear visualization of this identified suspicious entities of the network to the domain experts.
In the first module, the authors present some guidelines for constructing networks from relational domain data, and discuss that there are several different types to represent a collision network, particularly ten possible ways to connect three entities (collision, participant and vehicle).
They further discuss, considering the guidelines described, the drivers networks (networks where drivers involved in the same collision are connected), participant networks (where participants

are connected to corresponding drivers), Connect Passengers Through Accidents (COPTA) networks (where participants are connected to collisions vertices, a new type of vertex), and the vehicles networks (where collisions are represented by edges between vehicles, which also connects to all participants), as seen in figure 2.1. Finally, they state that they use different types of

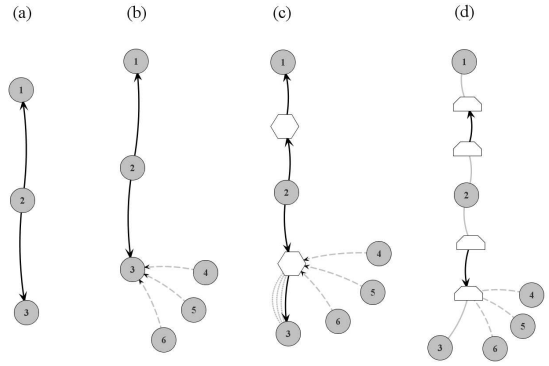


Figure 2.1: Four types of networks representing same two collisions - (a) drivers network, (b) participants network, (c) COPTA network and (d) vehicles network. Rounded vertices correspond to participants, hexagons correspond to collisions and irregular cornered vertices correspond to vehicles. Solid directed edges represent involvement in some collision, solid undirected edges represent drivers (only for the vehicles network) and dashed edges represent passengers. Guilt in the collision is formulated with edge's direction. (Figure used with author's authorization)

networks in the different modules, using participant networks in the second module (suspicious components detection) and COPTA networks in the third module (entity suspicious detection). Furthermore, Subelj *et al.* suggest that edge betweenness [GN02] could be used to simplify large networks and end with smaller and simpler connected components to input the second module, without loosing relevant information.

The second module aims to detect fraudulent components within the networks of participants, and eliminate all others. Domain expert knowledge suggest that these fraudulent components share some structural properties, such as dimension, density, centrality measures, existence of cycles and so on. The authors define a collection of such binary indicators, that will be used to produce an overall answer to whether a component is suspicious. For the structural independent indicators, domain experts set simple thresholds, while for the dependent the process is not so trivial. Subelj *et al.* propose to build random networks with rewiring algorithms representing honest behaviour, assess its values, and then decide if the real networks indicators correspond with honest or fraudulent activity. Then they discuss the use of principal component analysis of RIDITs [BL77] to sort of weight each indicator reflecting its importance and score each component as suspicious or not.

The third module is related to the detection of suspicious entities within the already detected suspicious components. The Iterative Assessment Algorithm (IAA) assigns a suspicion score to each participant, considering not only both intrinsic and relational attributes of the entity evaluated, but also of the related entities, which results in using indirectly all network on the assessment of one entity. IAA uses some assessment model AM to iteratively score an entity, which arguments are the set of scores for the neighbour entities of the entity considered, the intrinsic attributes of related entities and itself, and the relational attributes of the entity. The algorithm starts with a fixed arbitrary value for the initial scores and then iterates a number of times or until some convergence parameter.

The authors propose and evaluate three models, AM_{raw}, AM_{bas}, and AM^{mean}. The first being

the simplest, is just the sum of suspicion scores of the related entities; the second model already introduces intrinsic and relational attributes of entities as factors; the third model sort of averages the previous two using vertex degree and Laplace smoothing, after some normalizations. After knowing each participant suspicion score, other entities such as collisions and vehicles can be assessed using one of the previous models. In figure 2.2 it can be seen what would be the final module of the system, a visualization framework to facilitate experts to analyse the case.

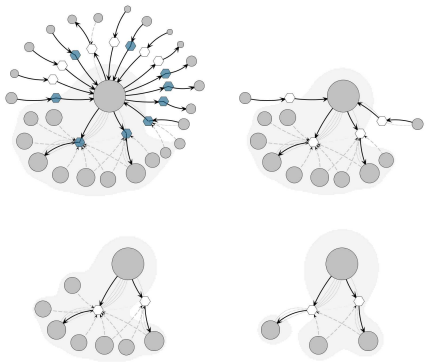


Figure 2.2: Four COPTA networks showing same group of collisions. Size of the participants' vertices correspond to their suspicion score; only participants with score above some threshold, and connecting collisions, are shown on each network. The contour was drawn based on the harmonic mean distance to every vertex, weighted by the suspicion scores. (Blue) filled collisions' vertices in the first network correspond to collisions that happened at night. (Figure used with author's authorization)

The novel algorithm is evaluated on real world data, automatically retrieved from police records, resulting on a data set of 211 participants and 91 collisions. The best performance is achieved with the AM_{bas}^{mean}, resulting in an average AUC (Area Under Curve) of 0.9228. The method AM_{raw}^{mean} also exceed in performance other well known centrality measures as betweenness, closeness, distance and eigenvector centrality. The authors also compared the algorithm with Naive Bayes, suport vector machines, random forest and k-nn (best performance up to AUC$\approx$0.86).
Subelj *et al.* finally concludes that the results presented before suggest that appropriate data representation is fundamental, and that their method obtained strong results with high recall (important for fraud detection purposes). Furthermore they state that domain knowledge can be incorporated in the system and that it improves performance. They also note that running the IAA for too many iterations over-fits the model.

2.2.4.2 Chen, H. *et al.* method

Chen *et al.* [CCX+04] propose a general framework for crime data mining, grounded by the experience of a jointly project developed with police departments of USA, the Coplink project. They begin to classify crimes in different categories, and then identify the main data mining techniques used for fraud detection and its major applications on crime detection: entity extraction to automatically identify person, addresses, and other information from police reports; clustering techniques to identify individuals who committed identical crimes; link analysis to identify identical transactions and uncover money laundering schemes; association rule mining to reveal patterns on intrusion attacks for instances for later protection systems; outlier detection to spot intrusion attacks online; classification to decide whether an email is spam, or to predict crimes; string comparator techniques to match similar police records; and social network analysis to reveal gangs and other criminal substructures, also with visualization abilities.

They based their study on a dataset consisting on 1.3 million police records of suspect and criminal. The authors claim that some types of techniques are more suitable for a specific type of crime than others. They place the 4 major classes by increasing order of analysis capability, and relate them with crimes categories: entity extraction, association, prediction and pattern visualization techniques. Also, they organize types of crimes in increasing public harm: traffic violations, sex crime, theft, fraud, arson, gang/drug offences, violent crime and cybercrime. In this sense, they place social network analysis as a facilitator technique for crime association and pattern visualization.
To give an example, Chen *et al.* describe three cases on a Coplink case study, named-entity extraction, deceptive-identification and criminal-network analysis. For the scope of this thesis, we will stick with the latter one. Their task was to identify subgroups and key elements on criminal networks built upon 272 incident summaries involving 164 committed crimes. A concept-space approach helped to extract criminal relations, weighted by a co-occurrence perspective, that is, how frequently a given pair of criminals were cited in the same incident record. They used hierarchical clustering techniques to identify connected components and then block-modelling approach to analyse relations among them. To identify key actors on those subgraphs, centrality measures such as degree, betweenness and closeness were applied.
From this analysis, there were identified 16 target gang members and cross-validate the results with experts from the police department, who confirmed the results. Not only the criminal gangs corresponded to the subgroups identified, but also most of the key members were in fact leaders of the known gangs. They further discuss the drawback of their framework of generating static networks, the possibility to have a symbiotic interaction of techniques to increase performance and the challenges of the field.

2.2.4.3 Chiu *et al.* method

Chiu *et al.* [CKLC11] focus on internet auction fraud and present a hybrid system using both social network analysis and other data mining techniques.
They begin to build an Internet auction transaction network, based on the relation *Seller -> Buyer*, giving a hierarchical perspective to the auction transaction. A seed account is selected from the blacklist provided by the Yahoo! auction website. Having this, the algorithm searches for all account buyers of the seed, and then drill down another level in the same way, ending up with a hierarchical view of this 3-level transaction (the seed, the seed's buyers, and the buyers of the seed's buyers).
A previous experiment studied regular patterns of honest transactions, and suggested that their behaviour is like "one to many" and "many to one", that is, one seller sells to many buyers, and one buyer buys to many sellers. This means that behaviours that resemble repetitive closed loops transactions might indicate fraudulent activities. Actually, the 2-core indicator applied to these kinds of networks have been successfully to filter suspicious activities on Internet auction transactions [Sun09]. Having this done, the authors can spot the subgraphs of the entire network that might represent auction fraud.
Several SNA metrics are applied to the networks to assess its characteristics, before being pruned by the 2-core technique. Such metrics, like degree, nbetweenness (normalized betweenness), k-core (*e.g.* 2-core means that each node has at least 2 edges to other nodes), k-plex (*e.g.* k-plex_k=2_size=5 means that there are 5 nodes and each one is related to at least 5-2=3 other nodes) and n-cliques (*e.g.* n-cliques_k=1_size=3 means that there are 3 nodes at the distance 1 of each other) are used to assess networks' indicators and create a vector of binary values for each

record, which represent whether the network has or not a given indicator. These vector is going to be in input for the three data mining classifiers (CART, C5.0 and neural networks), to decide whether a transaction record is fraudulent or not.
Results showed that the accuracy rates of neural networks were inferior compared to the other two classifiers. Furthermore, the authors presented some interesting insights on the relevance of the indicators for classification using CART and C5.0, for instances, that the n-cliques is not significant to increase classification accuracy, suspicious transaction can be identified directly using 2-core technique and that the fraudulent accounts can be classified from the suspicious transactions using nbetweenness, 6-core, k-plex_k=2_size=5. Chiu *et al.* further discuss the knowledge outcomes regarding types of Internet auction accounts and regarding classification models.

2.3 Market Solutions

This section presents an overview of the different solutions, from a few of the main players, related to fraud detection. The ability to flag a claim as likely fraudulent enables agencies to maximize efficiency as it prevents investigators to spend too much time on honest claims and helps prioritize higher value cases. These frameworks provide a comprehensive handling on claims and consequently possible fraud cases, first with advanced data acquisition and management methods, sophisticated analytical techniques for fraud detection, and lastly with useful report generation tools and case administration for further investigation.

2.3.1 IBM i2

IBM i2 [IBM] is a framework from IBM to help fight crime in many domains and contexts, whether it is related to financial crimes, national security or monitoring activity across national borders. This framework features several different products, each focused on helping to investigate a specific type of crime, for instance:

- i2 National Security and Defense Intelligence supports government agencies gather, merge and making sense of all the information necessary to coordinate all type of security operations and to help identify attacks to public safety. It features a social network analysis module, which can be useful to identify key players and have a clear visualization of group interactions, and can be combined with event, geospatial and temporal elements.

- i2 COPLINK is a software especially made for police that enables a great flexibility of interactions with other law enforcement organizations by sharing data in a secure way, offers a vast search tool featuring person, location, vehicles and other objects, and facial recognition search. Additionally, besides sharing the integration of information and intelligence features of the previous tool, COPLINK is available on mobile devices, allowing officers to stay update everywhere and anytime.

- i2 Integrated Law Enforcement integrates analytics features, lead generation and communication technologies to deliver insightful reports on crime investigation. It gives a comprehensive overview of of policing and partners' information, to analyse and develop more efficient strategies based on intelligent insights, to better predict, prevent and uncover criminal activities.

- i2 Analyst's Notebook is an extensive environment for visual analysis to discover patterns, networks and trends in complex structured and unstructured data from several sources. The

insights originated from these powerful tools are easily converted visual briefing charts and integrated with other intelligence products. This particular solution has an extremely flexible data acquisition method, efficiently hands telephone call records data, financial transactions, IP logs, mobile forensics data and so on, through a wizard-style visual importer, a modest drag-and-drop tool, and via IBM i2 iBridge and IBM i2 Information Exchange for Analysis Exchange. These two last tools smoothly integrate some relational databases, such as Microsoft SQL Server and Oracle 11g, and facilitates concurrent searches on multiple sources. On top of these, there's the software development kit i2 Analyst's Notebook SDK, which exponentially increases the flexibility of this solution, allowing it to adapt to the organizations' specific requirements. It enables independent applications to control i2 Analyst's Notebook capabilities, like input data using and visualize results using other interfaces.
The essential feature of this solution is the Association Charting / Link Analysis. This component offers a truly immersive intelligent environment with plenty of formatting options, integrates a timeline perspective, provides item semantic types to approximate data to real world meaning, filters and histograms for a quick data overview, temporal analysis and heat matrix view for details on time events, geospatial analysis with the integration of Google Earth Explorer and ArcGIS Server, entity matching to cope with the ability of integrate data from multiple sources and treat the duplication of same entities properly, easy charts generation and, the most relevant feature in this scope, the social network analysis component. This module provides deep insights on how groups interact and behave through mapping and measuring complex networks of entities, allied to temporal analytics previously referred. The main centrality measures of social network analysis are included (degree, closeness, betweenness and eigenvector) along with conditional and highlight formatting, as well as K-Cores which is a clustering technique and the possibility of assign weights to edges in straightforward ways.

2.3.2 SAS Fraud Framework

SAS offers a framework exclusively dedicated to fraud detection, with particular solutions focused on the main industries, such as SAS Fraud Framework for Insurance, for Health Care and for Government, with the possibility to adapt a solution to a specific industry needs. Besides this main framework solutions, SAS offers SAS Fraud Management, SAS Fraud Network Analysis and several other solutions more related to national security, crime investigation and law enforcement [SAS].
Concerning to the insurance industry framework, SAS solution processes data through analytics models along with rules engines to spot anomalies and raise alerts, with some near real-time scoring features. Some of the techniques this fraud analytics engine uses consist in text mining, exception reporting, automated business rules, predictive modelling, database searches, anomaly detection, and other data mining methods. It also features a social network analysis component, for visualization purposes only, *i.e.* it outputs the network of entities as it is, to help fraud investigators visualize links and patterns otherwise hard to identify on plain data. It can draw networks with data from multiple sources and avoid duplicated entities that might occur in the process. Plus, this tool can score the associated networks of a customer, based on risk scores or red flags previously calculated, and lets the analyst to drill down to details such as historic policy and claims activity and other information related to linked entities.
From this solid analysis, it can generate some documentation and integrate these reports on the case investigation process within the solution. This integration of tools within the framework

makes it easy to collect all information, produce insights and share them within the team, and later manage the case for further investigation.

2.3.3 Deloitte Analytics

Deloitte also offers some solutions when it comes to fraud [Del]. uDetect is a solution tailored for unemployment insurance fraud, not only identifying fraudulent behaviours, that can result in benefit year earnings overpayments, but predicting them too. It evaluates claims automatically by scoring the unemployment insurance application, secondly by updating each risk score considering the weekly patterns of submissions, and lastly by identifying, based on predictive models, unusual complex applications that could result in more severe losses. There is also an analytics-driven approach to solve the energy theft problem, which is a challenge because utilities really struggles to make sense of the huge amount of data available. Utilities Fraud Finder uses SAP Hana to work with these big data volumes and enable companies to analyse for instance 20 years of worth data.
Enterprise Fraud and Misuse Management is a more general solution, most commonly used in vertical industries such as banking, brokerage, insurance, retail and government, highly adaptable to a company's requirements. This product gives the ability to search for fraud in real-time on transactional activity and identify fraud rings, by implementing data science techniques such as predictive analytics, social network and geospatial analysis, combining structured with unstructured data like emails, social media, etc. This solution was recently adapted to fraud related to loyalty programs, a new exploitation point for fraudsters that can result in financial and reputational threat for organizations. Besides the Fraud Risk Assessment Deloitte offers, and on a more technological perspective, this solution can feature web behaviour detection and link analysis, all crafted in sophisticated dashboards for clear visualization. With the aim to detect and predict occupancy fraud risk, a predictive model was built that used some factors like daily reimbursements reports, daily property room level data, occupancy rate breakdowns, reservation data, hotel attributes and average daily rates.

2.3.4 Others

There are some other players besides the ones presented earlier, with solutions that cover mostly the same field or with similar applications:

- NetReveal platform [Sys] is a product from Bae System which uses different techniques depending on the type of fraud the organization is dealing with. In the insurance and tax fraud context, this solution uses network models, but it can use text analysis if it deals with corporate fraud, unsupervised learning if it is about detecting unauthorized trading, rule libraries for compliance solutions, etc. Network analytics links entities across data and applies scoring models to detect fraud rings or money laundering scenarios.

- Centrifuge Analytics [Cen], from Centrifuge, provides a combination of charts, timelines, tables, relationship maps and geospatial views in order to detect fraud and risk in financial services, retail, pharmaceuticals and government sectors. This application is accessible from the Internet, and can be embedded on current solutions used by the organization. It has built-in support for Hadoop, Splunk and Digital Reasoning Synthesys platforms, which expands the range of application to a new level.

- Fico provides a vast number of products regarding fraud, eventually resulting on a comprehensive platform to fraud, security and compliance management, the Fico Falcon Platform [Fic]. Namely, FICO Insurance Fraud Manager includes social link analysis techniques and predictive models such as neural networks, can detect organized fraud rings and instantly decide which claims to play automatically. To cope with the link analysis feature, FICO provides Identity Resolution Engine to evaluate the true entity of a person.

- Palantir [Pal] is a company that develops software used in a broad range of applications, from uncovering human traffic rings, finding exploited children and solving complex financial crimes. Palantir Anti-Fraud translates transactions, network traffic, weblogs and other dense, low-signal, disparate data into a coherent object model, so analysis is done at a more meaningful entity level. When a fraudulent event is identified, the software runs clustering algorithms to spot cases that conforms to the latter fraud pattern. Palantir Insurance Analytics focus on the healthcare industry, uncovering fraud perpetrated by healthcare providers, pharmacists, and patients, having special attention to the confidentiality of sensitive data. But perhaps the most surprising application of Palantir is to counter-terrorism. Recently, Palantir intelligent software was used by the USA Marines for forensic analysis of roadside bombs and predicting insurgent attacks in Afghanistan, was involved in the localization of Mexican drug cartel members and Osama Bin Laden, tracked hackers who installed spyware on the computer of the Dalai Lama, and helps CIA in several other missions.

There are also some open-source tools available for SNA:

- JUNG [jun] - Java Universal Network/Graph Framework, is a library written in Java, for the modelling, analysis, and visualization of graphs or networks, including algorithms specially for SNA.

- SNAP [LS14] - Standard Network Analysis Platform (SNAP) is a library written in C++ (and available in Python through an interface), for network analysis and graph mining, able to manipulate huge networks of millions of nodes and billions of edges.

- Pajek [BM98] - Developed in Slovenia, it is a well-known noncommercial program for analysis and visualization of large networks.

- R [T$^+$12]- R is a free open-source programming environment and a programming language, designed for statistical computing. It can be extended through packages, and particularly for SNA purpose it can be useful to use igraph (generic network analysis package), network (manipulates and displays networks), statnet (statistical modelling of networks package) and sna (social network analysis package).

The graph generator this work proposes (described on the next chapter) is built upon the python library of SNAP. The rest of the analysis is carried out with python libraries as well. Nevertheless, IBM I2 was tested and could provide some quick and rough out-of-box insights related to network measurements.

Chapter 3

Proposed System

The following chapter describes the system proposed, which is to be considered a part of a major monitoring solution regarding insurance analytics. It is given a holistic view of the end-to-end fraud detection module system, as so as a detailed description of each step that composes it.

In first place, it is described the random graph generator that generate the dataset used to train and evaluate the classifier. Then, we describe the fraud detection module, that includes the stage of data under-sampling, community detection, feature extraction and subsequent transformations (such as by the PRIDIT technique), and the classification task. Figure 3.1 illustrates the workflow of both the graph generator and the fraud detection module.

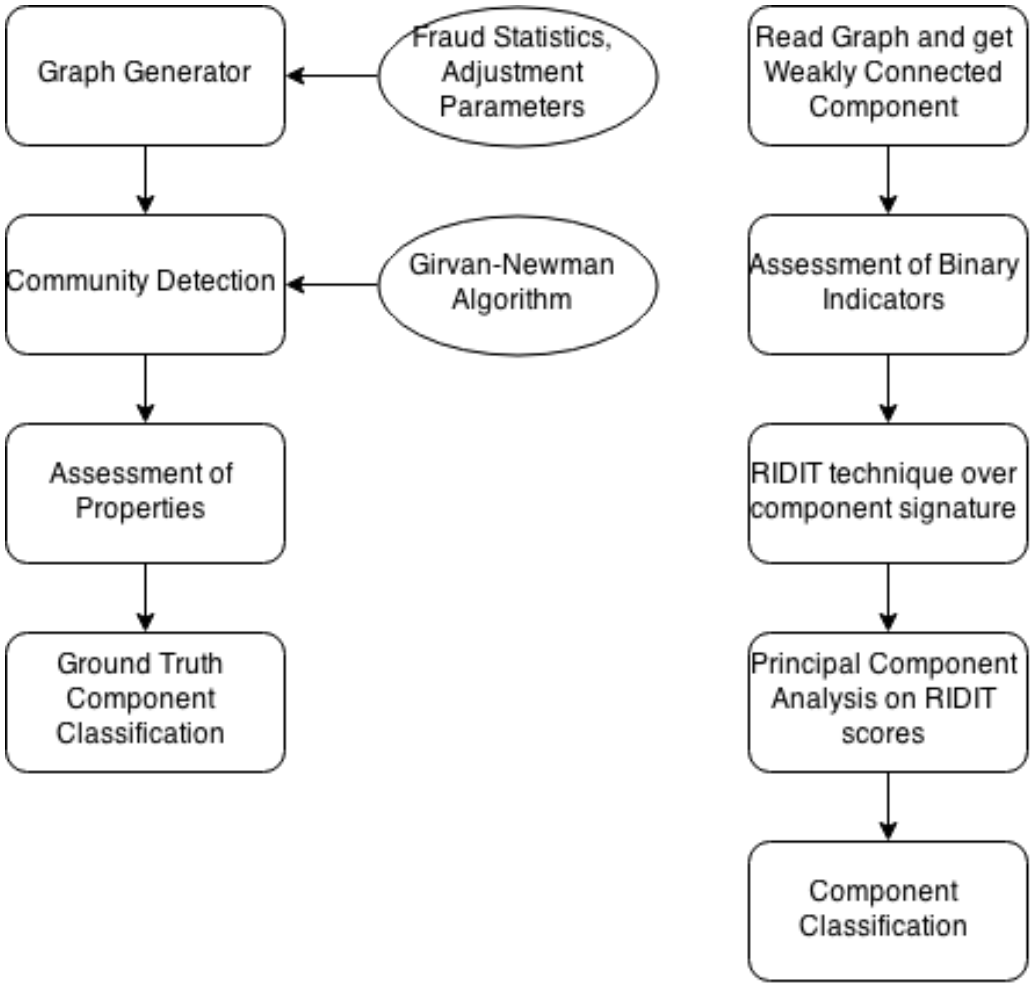


Figure 3.1: Schema for the proposed system.

3.1 Automobile Insurance Claim Graph Generator

As seen in the last chapter, fraud in automobile insurance has been already addressed in the literature, mostly from a "record" perspective, *i.e.* by analysing data regarding each entity as an isolated island, rather from a network perspective where it can be explored relations between entities.
Due to its characteristics and nature, which involves the manipulation of sensitive information that is either property of insurers or police departments, there are no public datasets available suitable for this kind of analysis, to the best of our knowledge.
Although the lack of public datasets, the domains experts working on insurance fraud investigation have a natural empirical knowledge regarding what a typical fraud pattern looks like. Moreover, there are some comprehensive reports on automobile insurance which give statistics on the impact of fraud on countries like England or Italy, among other practical insights discussed later [+] [oE10] [Ins13] [dS13].
Plus, Subelj *et al.* have explored a dataset with real data from local police [ŠFB11] and described in a very useful way some of its features, particularly on data translated to graph representation, regarding the appearance of the graph, and the characteristics of its components.
Having collected all this knowledge about the problem provides us a solid baseline to tackle the interesting challenge of generating data able to mirror real patterns, in this case, to mirror fraud structures associated with automobile collisions. In this section we present a random graph generator able to meet such requirement, resulting in an expected representation of a real scenario.
Resuming, the goal of this generator is to generate graphs that match the following key criteria:

- the patterns of fraud and non-fraud should be random, but resemble expected known behaviours;
- the ratio between fraud and non-fraud entities should be realistic;
- the graph should have a meaningful interpretation, apart from respecting the other requirements defined previously, *i.e.* representing a well-known scenario, such as the Lisbon automobile park reality.

Each graph is generated considering a set of 5 initially fixed constants, which act as parameters to the algorithm and help to meet the latter requirements. Some of these have strictly the purpose of regulate some aspects of the graph, while others are the ones based on real statistics and data features, providing the aimed consistency with the reality.
To cope with the first criteria, we have founded our approach on the expertise provided by Deloitte insurance experts, and the observations made by Subelj *et al.* that the fraudulent structures of a graph would have such features:

- frequently much larger and denser compared to non-fraudulent structures;
- ratio of the number of collisions to the number of different drivers is generally close to 1, when in completely independent collisions it is 2;
- existence of vertices with degree values and centrality measures above normal;
- components have small diameter;
- and existence of short cycles;

There are two parameters that influence the behaviour of the generator in order to produce such results, the probability of a node to be named as fraudulent, and the probability of rewiring an edge. The use of these two values will be explained later on this section.
The third parameter is based on a statistic of a 2010 document from the European Insurance and Reinsurance Federation. According to it, the number of detected fraudulent claims in England on 2008 represented 0.92% of all motor claims, while in Italy it is estimated that an average of 2.81% of claims involve fraud, a value that varies from around 1% on the North to 8% on the south [oE10]. Despite of the lack of information related to Portugal in this matter, it is reasonable to expect values close to the reality of Italy, given the similarity of both countries.
Finally, to satisfy the third requirement, our approach consider both the number of vehicles in the universe set and the ratio of number of collisions to the number of vehicles in that universe. For instances, the data of Portuguese Insurers Institute related to the year of 2013 shows that for the region of Lisbon there are 1.385.131 vehicles and a percentage of 11,7% of accidents. These two values are respectively the fourth and the fifth of the initial set of parameters.
Resuming, the initial parameters are the following:

1. probability of nodes to be a fraudulent entity;
2. probability of rewiring an edge;
3. probability of a collision to be a fraudulent event;
4. number of automobile in the universe to study;
5. ratio of claims to number of automobile in the universe.

The generated graphs aim to represent what is called the drivers network, where nodes reproduce a drivers network, or in a more relaxed perspective, a participants network, where drivers or participants (thus also vehicles) are represented by nodes , and edges represent collisions between those nodes.
As slightly introduced on the previous paragraphs, there is a notion of fraudulent and honest behaviour associated to each entity, applied both to nodes and edges. This notion allow the generator to model interactions based on the fundamental assumptions:

- an honest claim/collision is most probably a result of an accident between honest entities/drivers, almost impossible to involve a fraudulent node, and impossible (for this purpose) to happen only with fraudulent entities;
- on the other hand, a fraudulent claim/collision is most probably a result of an accident involving only fraudulent entities, much less possible (but not negligible) to involve an honest node, and impossible to have the participation of only honest nodes.

Our approach starts by randomly sort each node n to be either set to fraudulent F with probability p_1 (the first parameter), or honest H with probability 1-p_1 , according to an uniform distribution $\mathcal{U}$:

$$\text{f\_node(n)} = \begin{cases} F & \text{if } \mathcal{U}(n) < p_1 \\ H & \text{otherwise.} \end{cases}$$

This parameter will have impact on the number and size of fraudulent components, and should be adjusted considering the other parameters' values. The nodes are stored in two lists, one for fraudulent and another for honest nodes, regarding their nature.
As the number of vehicles and the ratio of collisions to number of vehicles are previously defined, the number of collisions (in the graph represented by edges) is inherently also determined.

Similarly to what was done about the nodes, also each edge e is randomly sorted to be either named fraudulent F with probability p_3 (the third parameter), or honest H with probability 1-p_3 , according to a uniform distribution $\mathcal{U}$:

$$\text{f\_edge(e)} = \begin{cases} F & \text{if } \mathcal{U}(n) < p_3 \\ H & \text{otherwise.} \end{cases}$$

In the case of being an honest edge, the algorithm picks one node from the nodes' lists *fraud_list* or *honest_list* using the *random()* (function to choose a random item from a list), and according to a uniform distribution $\mathcal{U}$, in such way:

$$\text{pick(n)} = \begin{cases} random(fraud\_list[]) & \text{if } \mathcal{U}(n) < p \\ random(honest\_list[]) & \text{otherwise} \end{cases}$$

The probability p is set to 0.5% in this case, but it is sensitive to change if necessary for graph adjustments, as long as the value keeps to make sense, *i.e.*, keeps very low. The second node is always chosen randomly from the *honest_list*, according to the fundamental assumptions stated earlier.

On the same page, a similar procedure is applied in case of a fraudulent edge, with a minor, yet very important twist. The algorithm picks one node from the nodes' lists in such way:

$$\text{pick(n)} = \begin{cases} random(honest\_list[]) & \text{if } \mathcal{U}(n) < p \\ random(fraud\_list[]) & \text{otherwise} \end{cases}$$

with probability p set to 2%. Note that it is more probable to have an honest node on a fraudulent collision than a fraudulent node on an honest collision. This fact tries to meet the idea that a fraudulent individual might want to collide in purpose to a random car presumably honest under certain conditions.

Now comes the tricky part where the second parameter enumerated earlier, the rewiring probability, is crucial to meet the morphological requirements that Subelj described as characteristic of fraudulent components.

At this point, we have a fraudulent edge and two nodes picked from the nodes' lists. If both nodes happen to be fraudulent, which is the majority of cases, then, an edge rewiring will happen with probability $p_{rewiring}$, in such way that both nodes are already in the same component. In other words, with probability $p_{rewiring}$ the chosen nodes belong both to the same component, thus creating a cycle in this component when the edge is added. It is possible to exist laces, which can be interpreted as accidents involving just the vehicle concerned.

To evaluate the algorithm presented on this chapter related to the detection of suspicious structures, it is necessary to annotate data in the best possible way. The ground truth set used on this experiments consist on the annotation of connected components of each graph with either fraudulent or honest label. This annotation derives almost directly from the process of generating data, as the algorithm have the need to, at a given point, decide whether an entity is fraudulent or honest. Hence, each component is labelled fraudulent or honest according to whether it has more fraud or honest entities, respectively.

Graphs are stored as edge lists in plain text files, as well as the ground truth set, that consist on listing the vertices corresponding to each component and the related components' label.

3.2 Fraud Detection

The fraud detection algorithm used to identify suspicious entities is largely inspired by Subelj *et al.* [ŠFB11]. Despite it has already been referred in section 2.2.4.1, we now dig into the details and mathematics of the method, the particularities and adjustments this work implements and its technicalities of the implementation.
It have been previously described in the past section how graphs are generated.These graphs are used as input to the following method, both in the training and testing phases.

3.2.1 Community Detection

The main goal is to identify the suspicious components out of the entire network, where each component would represent a criminal scheme. That said, it becomes natural to identify in first place those components of the graph. This task is commonly referred as community detection, and there are extensive work on this subject [For10].
After visual analysis of the graphs' structures, the great majority of the components that we are interested in may be detected by simply identifying the weak components of the graph.
However, there are components of some graphs that have a much more complex structure, and are much larger than fraudsters networks are supposed to be. Although, such structures may be naturally divided in sub-communities, without loosing information or meaning of the results of posterior analysis. To address such cases we propose the Girvan-Newman community detection algorithm [GN02]. An example of the initial structure and the result of the Girvan-Newman method is represented on figure 3.2.

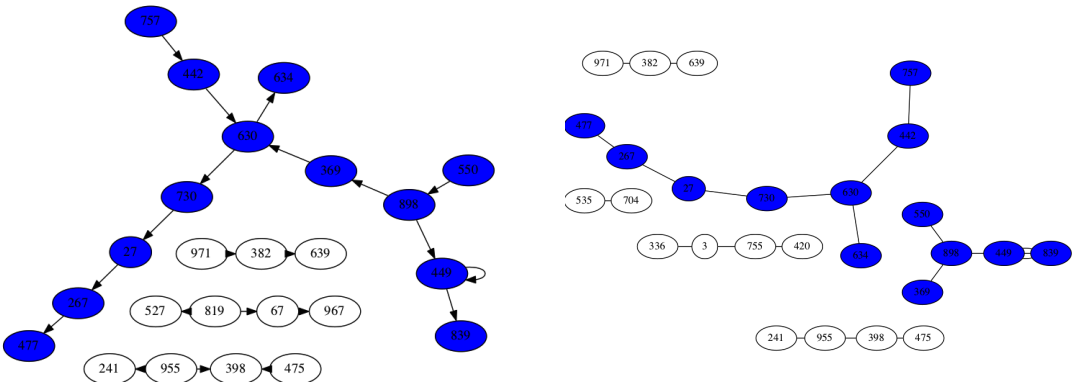


Figure 3.2: The blue component on the left represent the initial component and on the right the two blue components resulting of the Girvan-Newman algorithm

This method is based on the edge betweenness centrality (section 2.2.3.1), that measures how much an edge is in between of two communities. The value is calculated for each edge by counting the number of shortest paths between any pair of vertices that pass through that edge, and weight the count conveniently so the total sums up to 1. The method consists in the following steps:

1. Calculate edge betweenness value for all edges on the graph.
2. Remove the edge with the highest value.
3. Recalculate betweenness values for all edges affected by the latter step.
4. Iterate from step 2 and stop when all edges have been removed.

After this, we reconstruct the components by re-adding the edges connecting vertices that still belong to the same component.

3.2.2 Feature Extraction

The next phase of the algorithm is feature extraction. In this phase we assess features that can describe the components and give a hint on their fraudulent or honest nature. The selection of the feature set is of extreme importance because it has great impact on the method's performance, even if the rest of the method is well suited to the problem. The kind of features chosen to represent an object should be in harmony with the approach the method proposes. For example in this context of fraud detection using network characteristics it makes sense to choose features that explore relationships within the actors, more than its intrinsic characteristics.
The features used are the following:

- binary value that indicates whether the component corresponds to a "simple" collision, *i.e.* if the component is composed only by 2 nodes;
- ratio between the number of collisions to the number of drivers;
- the maximum node degree of the component and its degree centrality;
- binary indicator on the existence of a cycle in the component;
- component diameter;
- number of nodes (entities) in the component;
- number of edges (collisions) in the component;
- number of triads triad is a subgraph consisting on three nodes and the possible edges between them.

These features compose the feature vector used on the next stages. Note that features expressed by real values have to be binarized for the PRIDIT method, described below.

3.2.3 PRIDIT method

The Principal component analysis of RIDIT scores (PRIDIT) was first introduced by Brockett *et al.* [BDG+02]. It uses Relative to an Identified Distribution (RIDIT) scores, a scoring method introduced by Bross (1958) [Bro58].
In general, the RIDIT method transforms a set of categorical values into a set of probabilities of occurrence of those values. In this context, RIDIT method is used to address the question of how many components have a specific feature set to 1 or 0, considering the binary feature vector described in the previous section. The RIDIT scores R_i are calculated as

$$R_i(c) = \begin{cases} p_i^0 & \text{if } F_i(c) = 1 \\ -p_i^1 & \text{if } F_i(c) = 0 \end{cases}$$

where F_i(c) represents a binary feature of the component c, p_i^1 is the probability of the feature i is set to 1 and p_i^0 = 1 - p_i^1, considering the entire dataset.
For example, let the feature number 2 be equal to 1 on 95 of the 100 components, meaning that p_i^1 = 0.95, and thus, R_i(c) = p_i^0 = 1 - p_i^1 = 0.05. Such a low RIDIT score suggests that this feature does

not give us much information alone on whether the component is fraudulent, because almost all of the components on the dataset have it set to 1. On the other hand, considering the same feature on other component where F_i(c) = 0, RIDIT score would be R_i(c) = $-p_i^1$ = -0.95, suggesting that this feature is probably really useful to discriminate non-fraudulent components.
At this point we have a vector *j* of RIDIT scores for each component *i* of the graph, that is a matrix $R_{i.j}$. The next step is to weight these indicators according to their discriminative strength, using the principal component analysis od RIDIT scores.
Our goal is to have a vector of weights W, with as many elements as number of features. The result of matrix product S=RW represents a linear combination of RIDIT scores, and is the final weighted vector of component fraud scores. The idea is to assess features' values agreement with the entire set as

$$W = R^T S / ||R^T S||,$$

where the result is 0 when the set scores and a particular score are ortogonal, and touches minimum / maximum values when in total agreement. This way, features' scores which agree with the entire ensemble will get higher weights.
But the method does not stand here, and propose to refine the calculation by starting with some vector W^0 = [1,1,...,1] and iteratively calculate more accurate weights, until it converges to a fixed W^∞. Instead, W^∞ can be calculated as the first principal component of the matrix R^TR. To conclude, a component score may be written as

$$S(c) = R(c)W^\infty,$$

where R(c) is the vector of RIDIT scores for that component.

3.2.4 Classification using Support Vector Machine

Support Vector Machines (SVMs) were first introduced in 1992 by Boser *et al.* [BGV92], and nowadays are among the most well known supervised learning algorithms, standing doubtless the most relevant alternative to the traditional neural networks. Its popularity arises probably due to the apparent simplicity of usage with satisfactory results in most of the classification tasks.
It is relevant to outline the concept, mechanism and theory on what SVMs are founded, even though not going too deep into the details. SVM is a supervised learning technique, which means that given a training dataset composed by attributes and labels, the algorithm builds a model to predict the target label of the test dataset samples, based on its attributes.
Considering instances as points in some x-dimensional space and binary classification, the goal of SVM is to find the best hyperplane that separates the two classes. For example, instances having two features can be mapped into 2D space, and thus the SVM would find the straight line (in the most simple case, linear classification) which better separates both classes. When mapped into this 2D space, test instances would belong to one of the sides of the hyperplane, and labeled in conformity.
However, datasets often are non-separable, specially in low dimensions. To be able to perform non-linear classification, SVM use kernels to map inputs to high-dimensional spaces. Some of the most simple and used kernels are the linear, polynomial, radial basis function (RBF) and sigmoid.
Fraud detection task, along with some other problems, has some particular characteristics that makes it a not so trivial job for SVM. Datasets used for fraud detection have usually skewed classes distribution, with very few positive samples compared to the negative samples, which affects the performance of SVM [TZCK09]. Also, often the cost of misclassification is not equal

for false positives and false negatives, as we want to have the highest recall possible with a reasonable precision. One of the solutions to handle this problem is by undersampling the class that dominates the data set, in order to equalize the number of both classes' instances.
This process is implemented in a naive but effective way. It starts by identifying components according to its complexity, *i.e.*, the components with only two nodes (let this be type), the ones in which that verify the condition $number\_of\_nodes = number\_of\_edges - 1 \;\&\&\; max\_degree = 2$, and the remainders more complex components. These more complex components are never cut from the original graph, instead the other two types are the ones that are undersampled. Until the number of honest components is equal to the number of fraudulent ones, one randomly chosen component from each type is deleted.
The SVM classification process follows the guide provided by Hsu *et al* [HCL$^+$03]. The first step refers to data preprocessing. All the features used in this experiments are not categorical, thus are well suited to the method. Another preprocessing task the process should apply is data scaling. SVM may be particularly sensitive to data scale. We take up author's suggestion of linearly scaling to range [-1, +1].
The next point refers to the model selection, in particular the kernel and its parameters. Hsu *et al.* state that RBF kernel is a reasonable first choice, as it extends the linear kernel and can handle non-linear data separation. The authors also advise on the use of this kernel when the number of features is very large, which is not the case in this experiment.
Regarding the parameters search, the goal is to choose the best pair (C, γ) that maximizes some performance score, thus allowing the model to generalize with more accuracy and better predict new instances. The parameter C sets the penalty for misclassification. The higher this parameter, the more complex and rigid is the decision surface, because the classifier will try to classify well every single point. The parameter γ sets the importance of the training samples, in inverse logic. That is, for high values, other instances have to be more close to the given point, that for low values of γ.
For this task, authors recommend grid search along with cross-validation. Grid search will train the model with several pairs (C, γ), and choose for the best pair the one with higher cross-validation score. Note that grid search and the final evaluation of the whole SVM are not performed in the same set. The grid search is performed on a so called "development set", and uses cross-validation in its internal evaluation of the parameters pairs set. After choosing the best pair, the final evaluation of the method is done on a "test set", that was previously never used for the grid search.

Chapter 4

Experiments and Discussion

This chapter describes the datased used on the experiments carried out on this work, reports results and presents performance evaluations, and discusses the meaning of these results.

4.1 Dataset Description

The dataset used on the experiments reported on this chapter was produced using the generator described on section 3.1, with the following parameters:

- probability of a node being a fraudulent entity = 5%;
- probability of a collision being a fraudulent event = 20%;
- number of automobiles in the set = 10.000;
- probability of rewiring an edge = 20%;
- ratio of claims to number of automobile in the universe = 12%.

We are interested in analyze and detect suspicious components instead of individual entities, and due to this fact, the dataset was built by collecting components from one hundred graphs equally generated with the parameters outlined above. The proportion of components per class is far from evenly distributed, what is expectable given the nature of graphs produced and the reality they represent. The whole dataset is composed of 86177 components, with the initial distribution per class of around 9.5% fraud and 90.5% honest.
The unbalanced nature of the dataset is handled by undersampling honest components, process that is described on section 3.2.4. The motivation for the undersampling process and the consequences of such unbalanced classes are explored later on this chapter.
It is expectable that performance varies with the morphology of the graph (as the method mainly assesses morphology features), and therefore with the parameters outlined above. Such parameters were chosen due to the fact that the appearance of graphs seemed to be quite similar to what a real network would look like.
The figure 4.1 represents the kind of structures analyzed on these experiments. Blue nodes represent fraudulent nodes, while white nodes represent honest nodes.
No data correction was performed on the dataset, thus, due to the random character of the generator some components might not represent accurately the expected standard of their class. Because of that, some components may appear to have the exact same structure as others, while being from different classes. This fact will most likely result on non-separable data for SVM, hence decreasing performance.

4.2 Experiments

Metrics used on this section to report performance evaluation are among the most commonly used to evaluate classification systems:

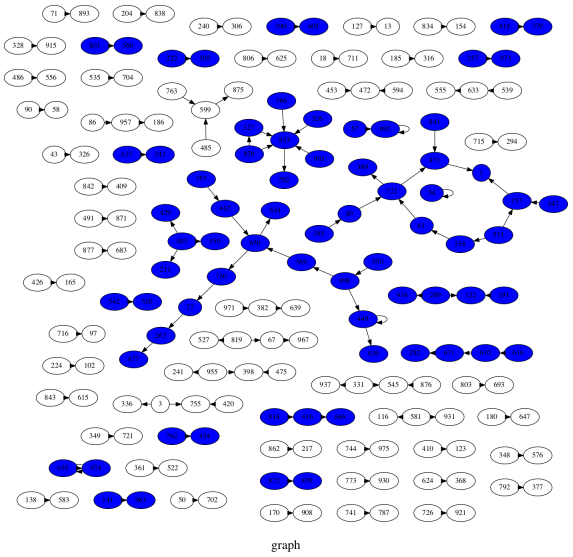


Figure 4.1: Holistic view of a graph similar to what is used for the experiments.

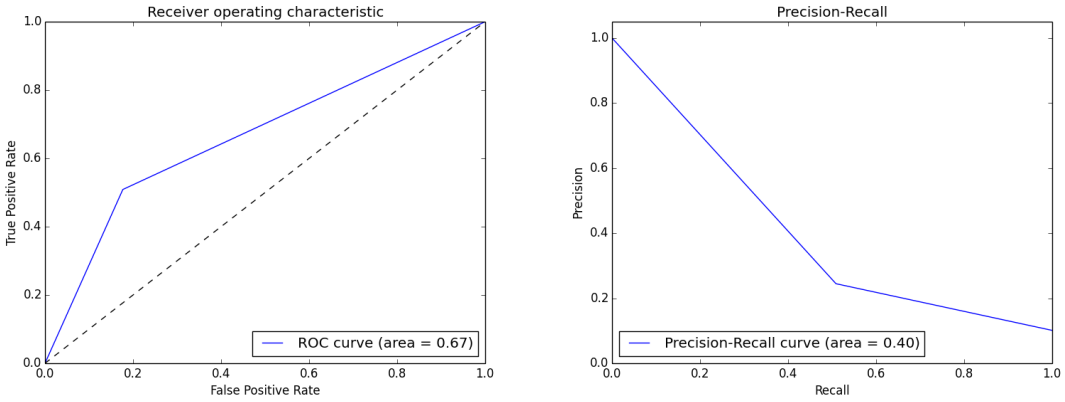
- Accuracy score - measures the proportion of correct predicted instances;
- Precision score - measures the percentage of correct positive predicted instances;
- Recall score - measures the percentage of positive instances reported;
- F1 score - summarizes precision and recall in a single value, according to the expression $\frac{2*p*r}{p+r}$;
- Area under ROC curve - ROC curve is given by plotting the values of false positive rate (ordinate axis) against true positive rate (abscissa axis) at several thresholds. The perfect system would have AUROC = 1;
- Area under Precision-Recall curve - The Precision-Recall (PR) curve results of plotting Recall on the abscissas axis and Precision on the ordinates axis. The perfect system would have AUC = 1.

Although the goal of all classifiers is to optimize both the precision and recall, in the context of fraud detection it is much more important to have high recall than high precision values; *i.e.*, it is crucial that the system reports every fraudulent component, with a reasonable low cost error if it reports some false positives. However, a compromise between the two must not be disregarded. ROC curves alone can drive to misleading conclusions when evaluating a method performance. PR curves are specially useful when handling unbalanced data sets results. An example of such fact is discussed later on this section. Therefore, to answer the question of how meaningful the classifier performance is for fraud detection, it should be more relevant to consider PR than ROC curves. Even though, both are reported on the next experiments.
These experiments aim to demonstrate that it is possible to automatize, at some level, the procedure of detecting suspicious groups of fraudsters or claims. For this purpose, the performance of PRIDIT method combined with SVM classification is compared with the SVM classifier that uses raw feature values as input.
The grid search for the best pair of SVM parameters evaluates pairs of values from a set of exponentially growing sequences, as suggested by [HCL+03]. For parameter C we try for C = $[10^{-3}, 10^{-2}, ..., 10^{3}]$ and for γ we try $\gamma = [10^{-4}, 10^{-3}, ..., 10^{1}]$. Parameter C is used for both Linear and radial basis function (RBF) kernel, while γ is only used for the second.

Grid search also allows to optimize regarding different target evaluation scores. We report results using parameter optimization for the target scores ['accuracy', 'average precision', 'f1', 'precision', 'recall'].

Figure 4.2 shows the performance of SVM classifier using the whole dataset and real values as indicators. These figures refer to the performance of SVM with class weights defined accordingly with the proportion of class instances and parameters optimized with grid search that ended up being the same for every score, resulting on C = 1 and γ= 0.

Low performances revealed by the AUC values on both graphics seem to validate what is already known about the behavior of SVM when handling unbalanced data. Even applying one of the classical solutions to address unbalance datasets, that is setting different class weights, did not result on satisfactory results. These results motivates the use of the balanced dataset version produced by the undersampling process, for the rest of the experiments.



(a) (b)

Figure 4.2: Results obtained with SVM, grid search optimization for accuracy score ,using the whole data set and real values features.

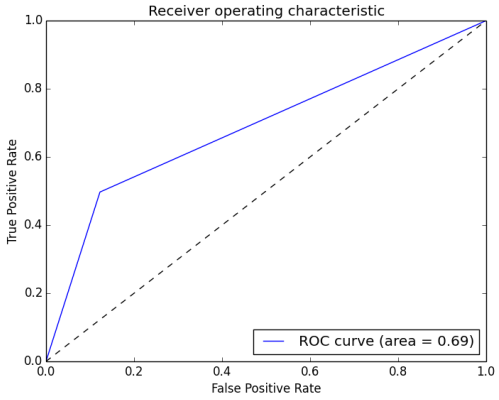
Figure 4.3 shows the performance of SVM using the PRIDIT scores. This experiment was conducted already using the undersampled dataset. It is also relevant to note that the figures are concerning to the grid search optimization for accuracy score. A variant is introduced by using 1 or 2 principal component values to calculate the final PRIDIT scores.

This experiments seems to suggest that using two principal components result in better performance than one. Using two principal component values increases the percentage of data variance explained, resulting on the slight performance improvement. The greater the variability of data, the greater the increment of data variance percentage brought by the use of the second score, resulting on better performance overall.

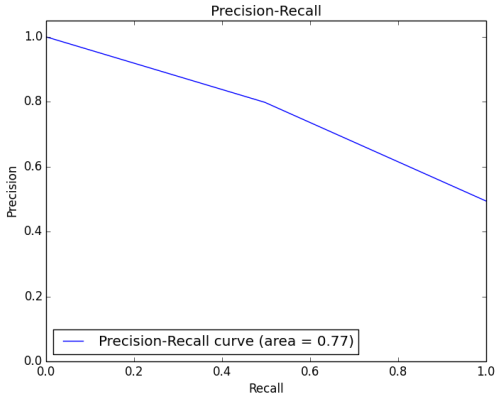
We would like to answer the question of whether PRIDIT method does guarantee a raise of performance, under these circumstances. Figure 4.4 shows the performance of SVM, this time using binary feature values, as the ones used for the PRIDIT method. As before, these figures are concerning to the grid search optimization.

It is interesting to observe that these results exceed the performance of the latter experiment report with PRIDIT method. This may suggest that PRIDIT, despite being a sophisticated method to transform data into another form, might strangulate and remove discriminant capacity to the values.

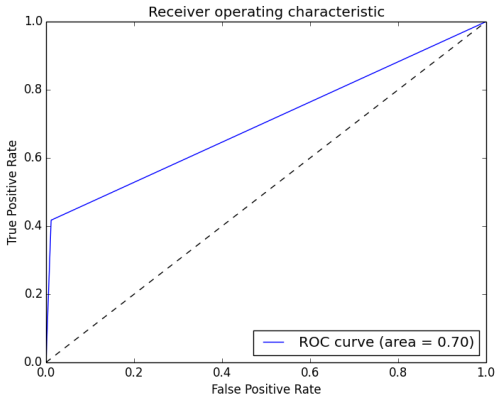
Motivated by the suggestion the latter experiment reveals, regarding the data discriminant power and the nature of data itself, the next experiment test a related hypothesis. This time, the SVM is



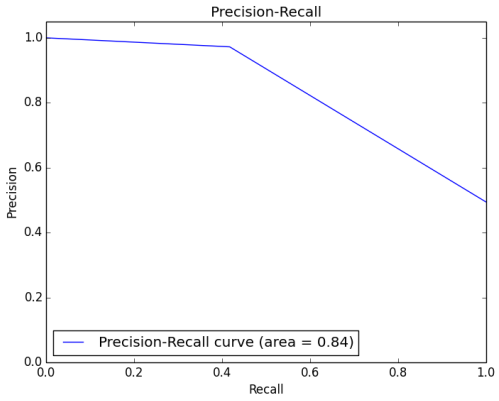
ROC curve - 1 PCA score



Precision-Recall curve - 1 PCA score



ROC curve - 2 PCA score



Precision-Recall curve - 2 PCA score

Figure 4.3: SVM performance using PRIDIT method.

trained with real features values instead of binary indicators. The figure 4.5 shows the results of this test after grid search optimization.

The AUC values regarding this test align with the intuition that real values provide more discriminant power than binary indicators and that the SVM model handle better this kind of data. Moreover, when compared to the rest of the ROC and PR curves reported so far, this system performs the best among all.

Table 4.1 summarizes the different metrics regarding the different experiments. The experiment "whole data" refers to the first test reported, using the whole dataset without undersampling. The rows "PRIDIT" refers to the one using PRIDIT data. The forth row refers to the experiment using binary feature values, without using PRIDIT, and the fifth row refers to the one using real features values. Despite "real values" recall and accuracy are slightly lower than expected, f1 score and AUC PR show that this method is the more robust.

Using two PRIDIT values does not increase that much performance when comparing to using one value. However, using just one value will produce a "hard margin" effect on the SVM hyperplane, due to the fact that C increases ten times. Given this, the model that uses two values is more adaptable and have more generalization capacity than the other.

It is relevant to note that tests on graphs after Girvan-Newman community detection algorithm did not show any perormance improvements. It does not question the utility of the algorithm

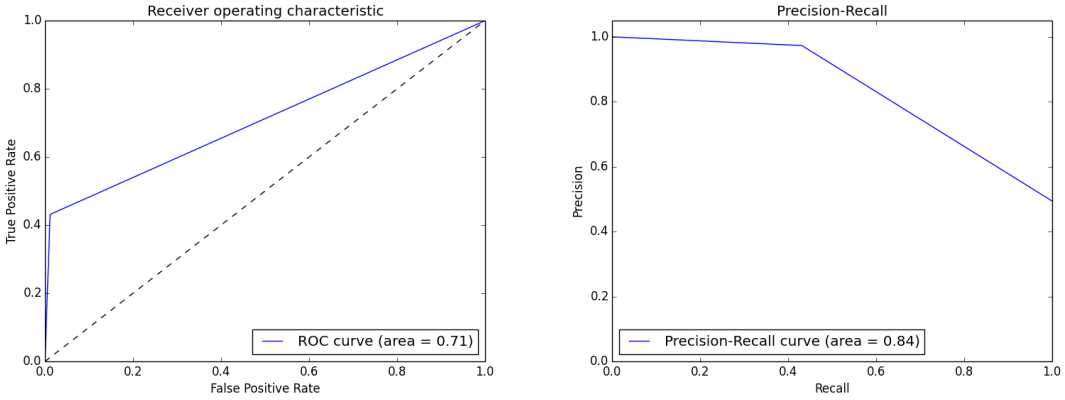


Figure 4.4: SVM performance using binary indicators.

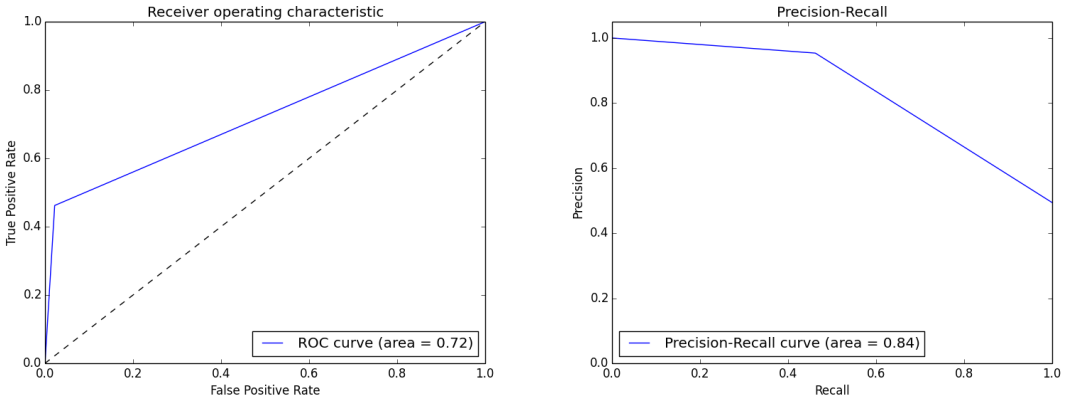


Figure 4.5: SVM performance using real values features.

though. We suggest that this set of components built upon the set of parameters showed is already sufficiently segmented for this analysis.

evaluation metrics / experiments	accuracy	precision	recall	f1	auc precision-recall	auc ROC
whole dataset	0.79	0.24	0.51	0.33	0.40	0.67
1 PRIDIT value	0.69	0.84	0.44	0.59	0.77	0.69
2 PRIDIT values	0.69	0.85	0.45	0.59	0.84	0.70
binary values	0.71	0.97	0.43	0.59	0.84	0.71
real values	0.72	0.95	0.45	0.62	0.84	0.72

Table 4.1: Summary of performance metrics for the various experiments

Chapter 5

Conclusion

The main goal of this work was the development of a graph generator able to resemble the characteristics expected from real insurance claim data. In addition, using this graph generator, we test the hypothesis of suspicious structures detection, that could indicate fraud activities.
The ability of automatically detect suspicious groups of entities represents a major improvement on the efficiency of fraud investigation departments, thereby skipping the screening phase and focusing on the investigation of true profitable cases.
To address this challenge it is necessary to have data related to insurance claims. Moreover, to test a network-based approach, data has to be convertible to graph representation. The lack of data sets suitable for this purpose originates the need of development a tool to bridge this gap.
The graph generator we propose is based on knowledge presented on the literature and on the experience of Deloitte's insurance experts. It uses statistic data from several insurance fraud reports from national and internacional credited organizations, in order to establish a common ground to the patterns it generates. It also incorporates the characteristics Subelj *et al.* identifies as commonly observed on fraud structures.
The approach followed in this work for the fraud detection task is largely inspired on the work of Subelj *et al.*. However, this work focus only on the detection of groups of fraudsters, instead of entities. In addition, this work tests the use of SVMs to classify structures with success at some level.
The experiments conducted by comparing performance results of the different methods and its variations allow to draw some interesting conclusions. The first fact to report is that data sets regarding fraud detection, which have skewed class distributions, have to be preprocessed in some way in order to be used for some experiment. In this work, we compared the results of the same method on unbalanced and balanced data sets, and confirm the significant increase on the performance the second one.
Experiments using the PRIDIT method in conjunction with SVM classifier showed that it could be useful to use more than one PCA value, as suggested by Subelj *et al.*. Even though the increase of performance on this experiment is not very significant by using two PCA values, it might indicate that it should be more expressive on data with high variability.
One of the main results of the experiments of this work is regarding the comparison between using data after PRIDIT method and raw data as input for SVM. What is interesting to see is that using either binary indicator or real values features have better performance than using PRIDIT method. This suggests a deeper analysis into the benefit of using PRIDIT method, and what are the circumstances under which it is advisable to use.

5.1 Future Work

We recognize that several improvements can be implemented on the graph generator. These enhancements should aim to increase the variability of graphs, in order to result in more meaningful feature vectors at the next phase of the method. On the same page, we are interested on the implementation of new features, capable of describe the extra variability we are looking for. Regarding

the behavior of the generator, a deeper study on how the parameters change the appearance of the graph would help to improve its quality.
The classification task should also be target of improvements. PRIDIT method did not prove to provide the best transformation to the binary indicators for the SVM. Other methods could be tested to identify the effects of the different input data on the SVM classification.

Bibliography

[+] Seguros +. Fraude nos seguros aumenta [online]. Available from: `http://www.segurosmais.com/seguros/fraude-nos-seguros-aumenta/`. 2, 26

[Alm09] Miguel Pironet San-Bento Almeida. Classification for fraud detection with social network analysis. *Dissertação de Mestrado, IST*, 2009. 16

[AS00] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. *ACM Sigmod Record*, 29(2):439–450, 2000. 13

[B+06] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006. 5

[BCGJ11] Francesco Bonchi, Carlos Castillo, Aristides Gionis, and Alejandro Jaimes. Social network analysis and mining for business applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):22, 2011. 1, 12, 13

[BDG+02] Patrick L Brockett, Richard A Derrig, Linda L Golden, Arnold Levine, and Mark Alpert. Fraud classification using principal component analysis of ridits. *Journal of Risk and Insurance*, 69(3):341–371, 2002. 10, 30

[BDK07] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th international conference on World Wide Web*, pages 181–190. ACM, 2007. 13

[BDT00] El Bachir Belhadji, Georges Dionne, and Faouzi Tarkhani. A model for the detection of insurance fraud. *Geneva Papers on Risk and Insurance. Issues and Practice*, pages 517–538, 2000. 8

[BGMP99] Francesco Bonchi, Fosca Giannotti, Gianni Mainetto, and Dino Pedreschi. A classification-based methodology for planning audit strategies in fraud detection. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 175–184. ACM, 1999. 8

[BGV92] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992. 31

[BH+01] Richard J Bolton, David J Hand, et al. Unsupervised profiling methods for fraud detection. *Credit Scoring and Credit Control VII*, pages 235–255, 2001. 10

[BH02] Richard J Bolton and David J Hand. Statistical fraud detection: A review. *Statistical Science*, pages 235–249, 2002. 6

[BKJ03] Emilie Lundin Barse, Hakan Kvarnstrom, and Erland Jonsson. Synthesizing test data for fraud detection systems. In *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*, pages 384–394. IEEE, 2003. 7

[BL77] Patrick L Brockett and Arnold Levine. On a characterization of ridits. *The Annals of Statistics*, pages 1245–1248, 1977. 17

[BM98] Vladimir Batagelj and Andrej Mrvar. Pajek-program for large network analysis. *Connections*, 21(2):47–57, 1998. 23

[Bon72] Phillip Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2(1):113–120, 1972. 15

[Bro58] Irwin DJ Bross. How to use ridit analysis. *Biometrics*, pages 18–38, 1958. 30

[BXD98] Patrick L Brockett, Xiaohua Xia, and Richard A Derrig. Using kohonen's self-organizing feature map to uncover automobile bodily injury claims fraud. *Journal of Risk and Insurance*, pages 245–274, 1998. 9

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009. 9

[CCX+04] 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, 2004. 18

[CDR+98] Soumen Chakrabarti, Byron Dom, Prabhakar Raghavan, Sridhar Rajagopalan, David Gibson, and Jon Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. *Computer Networks and ISDN Systems*, 30(1):65–74, 1998. 15

[Cen] Centrifuge. Centrifuge analytics datasheet [online]. Available from: `http://centrifugesystems.com/wp-content/uploads/2014/11/Updated-Centrifuge-Analytics-DataSheet.pdf` [cited 27 November 2014]. 22

[CEWB97] Kenneth C Cox, Stephen G Eick, Graham J Wills, and Ronald J Brachman. Brief application description; visual data mining: Recognizing telephone calling fraud. *Data Mining and Knowledge Discovery*, 1(2):225–231, 1997. 11

[CFPS99] 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. 9

[CKLC11] 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. 19

[Cor] Cornell University Law School. Legal information institute [online]. Available from: `http://www.law.cornell.edu/wex`. 1

[CPV02] Corinna Cortes, Daryl Pregibon, and Chris Volinsky. Communities of interest. *Intelligent Data Analysis*, 6(3):211–219, 2002. 10

[Cre73] Donald R. Cressey. *Other People's Money*. p.30. Montclair: Patterson Smith, 1973. 1

[Del] Deloitte. Deloitte analytics answers - fraud [online]. Available from: `http://www2.deloitte.com/us/en/pages/deloitte-analytics/solutions/analytics-answers-fraud.html#` [cited 27 November 2014]. 22

[Don04] Steve Donoho. Early detection of insider trading in option markets. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2004. 11

[dS13] Associação Portuguesa de Seguradores. *Insurance Market Overview*. October 2013. Available from: `https://www.apseguradores.pt/CMS_BO/DownloadResource.aspx?ResourceId=2126`. 2, 26

[DYB+07] Jeffrey Davitz, Jiye Yu, Sugato Basu, David Gutelius, and Alexandra Harris. ilink: search and routing in social networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 931–940. ACM, 2007. 13

[EN96] Kazuo J Ezawa and Steven W Norton. Constructing bayesian networks to predict uncollectible telecommunications accounts. *IEEE Intelligent Systems*, 11(5):45–51, 1996. 7

[Fan04] Wei Fan. Systematic data selection to mine concept-drifting data streams. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 128–137. ACM, 2004. 8

[Fic] Fico. Fico falcon fraud manager [online]. Available from: `http://www.fico.com/en/products/fico-falcon-fraud-manager/` [cited 27 November 2014]. 23

[For10] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010. 15, 29

[FP97] Tom Fawcett and Foster Provost. Adaptive fraud detection. *Data mining and knowledge discovery*, 1(3):291–316, 1997. 9

[FP99] Tom Fawcett and Foster Provost. Activity monitoring: Noticing interesting changes in behavior. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 53–62. ACM, 1999. 10

[FPSSU96] Usama M Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy. Advances in knowledge discovery and data mining. *the MIT Press*, 1996. 5

[Fre79] Linton C Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1979. 14, 15

[FTZ04] Gary William Flake, Kostas Tsioutsiouliklis, and Leonid Zhukov. Methods for mining web communities: Bibliometric, spectral, and flow. In *Web Dynamics*, pages 45–68. Springer, 2004. 16

[GD05] Lise Getoor and Christopher P Diehl. Link mining: a survey. *ACM SIGKDD Explorations Newsletter*, 7(2):3–12, 2005. 14

[GGLNT04] Daniel Gruhl, Ramanathan Guha, David Liben-Nowell, and Andrew Tomkins. Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web*, pages 491–501. ACM, 2004. 12

[GN02] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002. 15, 17, 29

[GR94] Sushmito Ghosh and Douglas L Reilly. Credit card fraud detection with a neural-network. In *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on*, volume 3, pages 621–630. IEEE, 1994. 6

[Gra73] Mark S Granovetter. The strength of weak ties. *American journal of sociology*, pages 1360–1380, 1973. 15

[HCL+03] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification, 2003. 32, 34

[HLL+07] Mary Helander, Rick Lawrence, Yan Liu, Claudia Perlich, Chandan Reddy, and Saharon Rosset. Looking for great ideas: analyzing the innovation jam. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 66–73. ACM, 2007. 12

[HPV06] Shawndra Hill, Foster Provost, and Chris Volinsky. Network-based marketing: Identifying likely adopters via consumer networks. *Statistical Science*, pages 256–276, 2006. 12

[HT99] Jaakko Hollmén and Volker Tresp. Call-based fraud detection in mobile communication networks using a hierarchical regime-switching model. *Advances in Neural Information Processing Systems*, pages 889–895, 1999. 10

[IBM] IBM. Ibm i2 framework [online]. Available from: `http://www-01.ibm.com/software/info/i2software/` [cited 27 November 2014]. 20

[Ins13] Europe Insurance. The impact of insurance fraud [online]. 2013. Available from: `http://www.insuranceeurope.eu/uploads/Modules/Publications/fraud-booklet.pdf` [cited 27 November 2014]. 2, 26

[jun] Java universal network/graph framework [online]. Available from: `http://jung.sourceforge.net/`. 23

[KK02] Min-Jung Kim and Taek-Soo Kim. A neural classifier with fraud density map for effective credit card fraud detection. In *Intelligent Data Engineering and Automated Learning?IDEAL 2002*, pages 378–383. Springer, 2002. 9

[KLSH04] Yufeng Kou, Chang-Tien Lu, Sirirat Sirwongwattana, and Yo-Ping Huang. Survey of fraud detection techniques. In *Networking, sensing and control, 2004 IEEE international conference on*, volume 2, pages 749–754. IEEE, 2004. 6

[KN06] Anuj Kumar and Vishnuprasad Nagadevara. Development of hybrid classification methodology for mining skewed data sets-a case study of indian customs data. In *Computer Systems and Applications, 2006. IEEE International Conference on.*, pages 584–591. IEEE, 2006. 16

[KPJ+03] Hyun-Chul Kim, Shaoning Pang, Hong-Mo Je, Daijin Kim, and Sung Yang Bang. Constructing support vector machine ensemble. *Pattern recognition*, 36(12):2757–2767, 2003. 7

[Llo82] Stuart Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982. 16

[LS14] Jure Leskovec and Rok Sosič. SNAP: A general purpose network analysis and graph mining library in C++. `http://snap.stanford.edu/snap`, June 2014. 23

[MA02] Mikhail I Melnik and James Alm. Does a seller?s ecommerce reputation matter? evidence from ebay auctions. *The journal of industrial economics*, 50(3):337–349, 2002. 12

[MLV+99] Yves Moreau, Ellen Lerouge, Hernan Verrelst, Joos Vandewalle, Christof Störmann, and Peter Burge. Brutus: A hybrid system for fraud detection in mobile communications. *7th European Symposium on Artificial Neural Networks, Bruges, Belgium*, 1999. 10

[MTVM02] 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*, 2002. 7

[New03] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003. 13

[NPS08] Hill Nguyen, Nish Parikh, and Neel Sundaresan. A software system for buzz-based recommendations. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1093–1096. ACM, 2008. 12

[NŞJ+05] 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 eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 449–458. ACM, 2005. 12

[oE10] Insurers of Europe. Cea statistics n°38 - the european motor insurance marketthe impact of insurance fraud [online]. 2010. Available from: `http://www.insuranceeurope.eu/uploads/Modules/Publications/cea-motor_web.pdf` [cited 27 March 2015]. 26, 27

[OMB+03] Thomas Ormerod, Nicola Morley, Linden Ball, Charles Langley, and Clive Spenser. Using ethnography to design a mass detection tool (mdt) for the early discovery of insurance fraud. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems*, pages 650–651. ACM, 2003. 9

[OML00] Hyo-Jung Oh, Sung Hyon Myaeng, and Mann-Ho Lee. A practical hypertext catergorization method using links and incrementally available class information. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 264–271. ACM, 2000. 15

[Pal] Palantir. Palantir [online]. Available from: `http://www.palantir.com/` [cited 27 November 2014]. 23

[PAL04] Clifton Phua, Damminda Alahakoon, and Vincent Lee. Minority report in fraud detection: classification of skewed data. *ACM SIGKDD Explorations Newsletter*, 6(1):50–59, 2004. 9

[PE06] Nicholas J Pioch and John O Everett. Polestar: collaborative knowledge management and sensemaking tools for intelligence analysts. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 513–521. ACM, 2006. 13

[PLSG10] Clifton Phua, Vincent Lee, Kate Smith, and Ross Gayler. A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv 1009.6119*, 2010. 5, 9, 10, 11

[PSV01] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic spreading in scale-free networks. *Physical review letters*, 86(14):3200, 2001. 13

[Rep13] Guarda Nacional Republicana. Fraude em seguros: Colaboração entre a gnr e a aps [online]. 2013. Available from: `http://www.apseguradores.org/apsbreve/apsbreve91/documentos/Artigo%20Fraude%20contra%20Seguros.pdf` [cited 27 November 2014]. 1

[RMN+99] Saharon Rosset, Uzi Murad, Einat Neumann, Yizhak Idan, and Gadi Pinkas. Discovery of fraud rules for telecommunications?challenges and solutions. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 409–413. ACM, 1999. 8

[SAS] SAS. Sas fraud framework [online]. Available from: `http://www.sas.com/en_us/software/fraud-security-intelligence.html#view-all-products` [cited 27 November 2014]. 21

[SBC+10] Rossano Schifanella, Alain Barrat, Ciro Cattuto, Benjamin Markines, and Filippo Menczer. Folks in folksonomies: social link prediction from shared metadata. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 271–280. ACM, 2010. 12

[SC11] John Scott and Peter J Carrington. *The SAGE handbook of social network analysis*. SAGE publications, 2011. 11

[Sco88] John Scott. Social network analysis. *Sociology*, 22(1):109–127, 1988. 13

[ŠFB11] 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. v, vii, 16, 26, 29

[ST11] Jimeng Sun and Jie Tang. A survey of models and algorithms for social influence analysis. In *Social Network Data Analytics*, pages 177–214. Springer, 2011. 15

[Sun09] Yung-Ta Sung. Developing a user-oriented internet auction fraud detection system by social network analysis and fraud rules matching. *Master's thesis, National Central University, Taiwan*, 2009. 19

[Sys] Bae System. Netreveal analytics platform [online]. Available from: `http://www.baesystems.com/product/BAES_166112/netreveal-analytics-platform` [cited 27 November 2014]. 22

[T+12] R Core Team et al. R: A language and environment for statistical computing. 2012. 23

[THHT98] Michiaki Taniguchi, Michael Haft, Jaakko Hollmén, and Volker Tresp. Fraud detection in communication networks using neural and probabilistic methods. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 2, pages 1241–1244. IEEE, 1998. 10

[TM69] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, pages 425–443, 1969. 11

[TZCK09] Yuchun Tang, Yan-Qing Zhang, Nitesh V Chawla, and Sven Krasser. Svms modeling for highly imbalanced classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(1):281–288, 2009. 31

[VD06] Sankar Virdhagriswaran and Gordon Dakin. Camouflaged fraud detection in domains with complex relationships. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 941–947. ACM, 2006. 12

[VDBD02] Stijn Viaene, Richard A Derrig, Bart Baesens, and Guido Dedene. A comparison of state-of-the-art classification techniques for expert automobile insurance claim fraud detection. *Journal of Risk and Insurance*, 69(3):373–421, 2002. 5

[VDD04] Stijn Viaene, Richard A Derrig, and Guido Dedene. A case study of applying boosting naive bayes to claim fraud diagnosis. *Knowledge and Data Engineering, IEEE Transactions on*, 16(5):612–620, 2004. 7

[WA00] Richard Wheeler and Stuart Aitken. Multiple algorithms for fraud detection. *Knowledge-Based Systems*, 13(2):93–99, 2000. 8

[Was94] Stanley Wasserman. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994. 12, 13

[Wea02] Margaret Weatherford. Mining for fraud. *Intelligent Systems, IEEE*, 17(4):4–6, 2002. 7

[WFYH03] Haixun Wang, Wei Fan, Philip S Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 226–235. ACM, 2003. 8

[WH97] Graham J Williams and Zhexue Huang. Mining the knowledge mine. In *Advanced Topics in Artificial Intelligence*, pages 340–348. Springer, 1997. 9

[Wil99] Graham J Williams. Evolutionary hot spots data mining. In *Methodologies for Knowledge Discovery and Data Mining*, pages 184–193. Springer, 1999. 9

[Wor08] Jennifer Wortman. Viral marketing and the diffusion of trends on social networks. 2008. 13

[YSK03] Dawit Yimam-Seid and Alfred Kobsa. Expert-finding systems for organizations: Problem and domain analysis and the demoir approach. *Journal of Organizational Computing and Electronic Commerce*, 13(1):1–24, 2003. 12

[ZSY03] Zhongfei Mark Zhang, John J Salerno, and Philip S Yu. Applying data mining in investigating money laundering crimes. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 747–752. ACM, 2003. 11