Social network analytics for supervised fraud detection in insurance

María Óskarsdóttir[1], Waqas Ahmed[2], Katrien Antonio[3,4,5,6], Bart Baesens[3,7], Rémi Dendievel[2], Tom Donas[2], and Tom Reynkens[3,5]

[1] *Department of Computer Science, Reykjavik University, Iceland.*

[2] *AG Insurance, Belgium.*

[3] *Faculty of Economics and Business, KU Leuven, Belgium.*

[4] *Faculty of Economics and Business, University of Amsterdam, The Netherlands.*

[5] *LRisk, Leuven Research Center on Insurance and Financial Risk Analysis, KU Leuven, Belgium.*

[6] *LStat, Leuven Statistics Research Center, KU Leuven, Belgium.*

[7] *Dept. of Decision Analytics and Risk, University of Southampton, United Kingdom.*

Abstract

Insurance fraud occurs when policyholders file claims that are exaggerated or based on intentional damages. This contribution develops a fraud detection strategy by extracting insightful information from the social network of a claim. First, we construct a network by linking claims with all their involved parties, including the policyholders, brokers, experts, and garages. Next, we establish fraud as a social phenomenon in the network and use the BiRank algorithm with a fraud specific query vector to compute a fraud score for each claim. From the network, we extract features related to the fraud scores as well as the claims' neighborhood structure. Finally, we combine these network features with the claim-specific features and build a supervised model with fraud in motor insurance as the target variable. Although we build a model for only motor insurance, the network includes claims from all available lines of business. Our results show that models with features derived from the network perform well when detecting fraud and even outperform the models using only the classical claim-specific features. Combining network and claim-specific features further improves the performance of supervised learning models to detect fraud. The resulting model flags highly suspicions claims that need to be further investigated. Our approach provides a guided and intelligent selection of claims and contributes to a more effective fraud investigation process.

Key words: Fraud detection, Social networks, Bipartite networks, BiRank, Supervised learning, Insurance.

1 Introduction

This paper presents a big-data risk analytics approach to detect suspicious insurance claims with an analytical model that combines the information from a network of insurance claims and involved parties with the traditional claim-specific features collected by the insurer. Insurance companies bring security to society by offering protection against financial losses. By organizing and managing pools of risks they enable individuals and companies to transfer risk in exchange for a premium. Fraud, the wrongful or criminal deception intended for financial or personal gain, is a considerable concern in many industries, such as in telecommunication and healthcare. Other types and areas of fraud include corruption, money laundering, tax evasion, identity theft and credit card fraud, to name a few (Baesens et al., 2015). The insurance industry is no exception. Insurance fraud is encountered when policyholders file claims that are exaggerated, fake or based on intentional damage with the intention to receive compensation from the insurance company. In non-life insurance, the total yearly cost of fraudulent claims is estimated at over €1 billion in the UK and $30 billion in the United States, which, in the latter case, corresponds to an increased yearly premium of between $200 and $300 for an average family (FBI, 2012; Smith et al., 2014). In Australia, fraud has been identified as the most costly crime category, with insurance fraud amounting a total of AU$2.1 billion every year, equivalent to AU$73 per insurance policy (Smith et al., 2014). Furthermore, the Dutch Association of Insurers reports a saving of €79 million in 2015 and 82 million in 2018 because of fraud identification efforts (Verbond van Verzekeraars, 2016, 2019). Undeniably, it is in the interest of both insurance companies and their customers to detect fraudulent activity to keep insurance contracts affordable.

When an insurance claim is flagged because of suspicion of fraud, fraud investigators use their expertise to confirm whether the claim is fraudulent or not (Warren and Schweitzer, 2018).

The insights obtained from such investigations are then used to help detect other suspicious claims. This comprises the fraud detection cycle. The rare occurrence of fraud on the one hand and the many uninvestigated cases or claims on the other hand, are important challenges in the construction of fraud detection methods (Jensen, 1997; Caudill et al., 2005; Baesens et al., 2015). The uncommon nature of fraud creates imbalanced data sets, where fraudulent claims are at risk of being treated as insignificant in analytical models, causing a bias towards the non-fraudulent claims. In addition, organizations have typically neither the time nor the resources to investigate all (suspicious) claims and to label each claim as fraudulent or legitimate (Warren and Schweitzer, 2018). As a result, only a small fraction of all claims have a known label. A fraud detection model is an efficient way to lead the investigators in the direction of fraudulent claims. By using techniques such as business rules or analytical models, only the most suspicious claims can be given to the investigator for further inspection. This avoids spending precious time on investigating non-fraudulent claims. Insurance fraud materializes in various ways; garages and hospitals may for example overcharge for material and work. Here, we consider fraud committed by policyholders when filing claims, such as false declarations, exaggeration of damage and intentional damage.

In the insurance industry, fraud detection models are based on business rules as well as analytical techniques (EIOPA, 2019). Business rules are usually formed by intuition, past experiences and input from the experts that conduct the investigations. Both unsupervised and supervised machine learning methods have been successfully used in the literature on insurance fraud detection strategies. A first and prominent fraud detection technique is PRIDIT, which computes a claim-specific fraud suspicion score via principal component analysis of so-called RIDIT scores (Brockett et al., 2002; Ai et al., 2009). RIDIT transforms a claim's available categorical features into a set of numerical features, reflecting the relative abnormality of the observed categorical features. As such, the scores represent the likelihood of fraud and can be used as input in further statistical analysis. Ai et al. (2013) aim to estimate the measure of overall fraud rate in a population of insurance claims. They develop the PRIDIT-based fraud rate estimator (called PRIDIT-FRE) which relies on a set of features and an investigated sample of claims. Golden et al. (2020) introduce aPRIDIT, an unsupervised learning method designed for a data set with discrete ordered categorical features. aPRIDIT is asymmetric in the sense that the different levels in a categorical feature may carry different information about the target group membership. Stripling et al. (2018); Ghorbani and Farzai (2018); Nian et al. (2016); Hainaut (2019) use principles from anomaly detection for unsupervised fraud detection strategies that aim at finding unusual and uncommon behavior deviating from the norm in the dataset. Approaching insurance fraud detection as a classification problem, Viaene et al. (2002) present a benchmark study of classification techniques. This supervised learning perspective has the advantage of learning fraud patterns in the data but requires past knowledge about fraudulent cases, which is rare because of the high class imbalance. Several fraud detection approaches tackle the problem of class imbalance by rebalancing the dataset using sampling techniques, such as under- and oversampling, SMOTE or ROSE (Subudhi and Panigrahi, 2017; Van Vlasselaer et al., 2016; Sundarkumar and Ravi, 2015). In a semi-supervised approach, Botelho and Antunes (2011) use unlabeled data to reinforce the learning algorithm. Other researchers strive at optimally adapting known techniques to the fraud detection problem (Li et al., 2018), investigate the value of cost-sensitive fraud screening (Viaene et al., 2007), or enriching the data with alternative information. Kose et al. (2015) use machine learning methods to incorporate expert knowledge in the fraud detection model to make the learning of fraud patterns more interactive. Wang and Xu (2018) mine experts' text descriptions of accidents and apply deep learning to detect fraud in car insurance whereas Šubelj et al. (2011) use networks to detect groups of collaborating fraudsters.

Fraudulent activity can often be associated with organized schemes carried out by collaborating fraudsters that go to great lengths in order to hide their tracks while maximizing their gain (Van Vlasselaer et al., 2016). In criminology, it is well known that two thirds of criminals commit crimes together (Reiss Jr, 1988) in order to maximize their reward and to minimize the risk of getting caught (Andresen and Felson, 2009). In particular, in organized crime, such as fraud, social relations are of a great importance as they not only provide access to co-offenders and profitable opportunities but also invigorate the necessary trust (Van Koppen et al., 2009). Police departments are therefore examining criminal networks to a greater extent to detect organized crime groups and their activity with social network analytics (Tayebi and Glässer, 2016; Troncoso and Weber, 2020). Social network analytics is the analysis of social structures, i.e. the ties between individuals and organisations, using networks and graph theory (Newman, 2010). In fraud detection, this activity can be detected for example by measuring proximity to known fraudulent cases or exposure to fraud influence (Jamshidi and Hashemi, 2012). In a first attempt of using this approach for fraud detection in insurance, Šubelj et al. (2011) argue that groups of collaborating fraudsters can only be detected with the appropriate representation, i.e. social networks, since they are often related to each other in some way. They use collision data from police records to build networks, rely on expert knowledge to identify suspicious sub-networks using PRIDIT scores and propose an iterative assessment algorithm to detect the most suspicious entities in each sub-network.

This paper addresses risk management in the context of insurance fraud. We propose a framework to build and to evaluate an analytical fraud detection model for insurance claims. We hereby combine the socio-demographic information about the policyholder and classical information about the claim with information about the claim's social network structure. The social network connects a claim with all its involved parties, e.g., policyholders, brokers, garages and experts, which in turn are connected to all claims they are involved in. In this way, we build a bipartite network with a few million insurance claims and parties that is a fitting representation of the relationships between the various parties by means of the claims in which they are involved. As such, we are able to look beyond the classical properties of the claim, the policyholder and the policy, and study the social structures of collaborating fraudsters in insurance fraud detection tools and models. The network contains claims from several types of insurance covers, e.g., motor, fire, and liability, which provides a holistic view on the customers of the insurance company. In the actual fraud detection model, we focus on the motor line of business.

We start by establishing fraud as a social phenomenon in this network and show that fraudulent claims are connected to other fraudulent claims to a greater extent. The collaboration of fraudsters in fraud circles to hide their tracks and lower the risk of getting caught, is a fundamental assumption for a network based fraud detection approach, and is known as homophily in social sciences. Homophily refers to people's tendency to associate with others whom they perceive as being similar to themselves in some way. In a network of claims, this means that the fraudulent claims are more often connected to other fraudulent claims and that non-fraudulent claims are more likely to be connected to other non-fraudulent claims. In addition, fraudulent and non-fraudulent claims are connected to each other to a lesser degree.

Having empirical evidence of homophily, we rank the claims in the network relative to known fraudulent claims. In essence, the goal of node ranking is to assign a score to each node in accordance with a specific criterion. This became a prominent research topic as internet search engines were emerging, starting with PageRank and HITS (Kleinberg, 1999; Page et al., 1999). Applications of node ranking are omnipresent including social analysis of Twitter accounts (Kwak et al., 2010), connectivity patterns in neuroscience (Lohmann et al., 2010) as well as fraud detection in online lending, tax evasion and social security (Min et al., 2018; Botelho and Antunes, 2011; Van Vlasselaer et al., 2016). We use the BiRank algorithm proposed by He et al. (2017) as an extension of the PageRank algorithm tailor made for bipartite networks. In addition, inspired by Van Vlasselaer et al. (2016), we modify the BiRank algorithm by adapting the query vector to include the knowledge of known fraudulent claims in the network. This steers the ranking towards the fraud in the network and results in higher scores for claims that are closer and more densely connected to known fraudulent claims. In this way, all nodes in the network obtain a score which represents proximity to fraudulent claims. We use the scores to compute a set of features which characterize the position of a claim in the network, such as, the average score of all parties involved in the claim. We combine the scores and the features from the network with the traditional features of the claim, the policyholder and the policy in an analytical fraud detection model to flag highly suspicions claims that need to be investigated further. The approach provides a guided and intelligent selection of claims and thus contributes to a more effective fraud investigation process. As such, our paper carefully demonstrates how to harness high-volume, high-dimensional and multi-source insurance data through network science. We design a framework to evaluate the potential of the network derived as well traditional insurance features to detect fraudulent claims with supervised machine learning methods. Thereby, we unravel the potential of these insurance data to prevent insurance fraud via big-data analytics.

The rest of this paper is organized as follows. In the next section, we present the dataset used in this study and demonstrate how to convert it into a bipartite network. Moreover, we discuss fraudulent structures and show empirical evidence of homophily among fraudulent claims. In Section 3, we elaborate on our fraud detection model and in Section 4 we explain the setup of the subsequent experiments. In Section 5, we present the results of our proposed technique and finally, conclude our research in Section 6.

2 Insurance fraud dataset

2.1 Description of the data

We analyze a dataset with a couple of million claims filed by an insurance company's policyholders over a period of a few years. Claims are formal requests sent to an insurance company for coverage or compensation of a covered loss. The dataset includes information on the insured event, such as the claim amount, as well as information on the policyholder and their claims history. Our goal is to build a fraud detection model for the motor insurance cover. Therefore, Table 1 details the available features for the motor insurance claims and their meaning. We refer to these features as *intrinsic* features. They are specific to the line of business (LoB) for which the fraud detection model is designed.

In addition to the intrinsic features, each claim has a list of the people and companies involved in the claim. These are policyholders, brokers, experts and garages. Policyholders are people or companies that are involved in the incident in one of the following ways. A policyholder may be covered by the insurer and request compensation using their own policy, or the policy of another client of the insurance company. Alternatively, they may be involved in a claim (e.g., as a victim) while not being covered by the insurance company. Insurance brokers are individuals or companies that sell, solicit and negotiate insurance on behalf of policyholders. Experts evaluate the loss filed via an insurance claim and garages repair damaged cars. In this paper the term *party* refers to all these entities that are not claims. We use claims and involved parties to build a network, connecting claims and parties across all available LoBs.

To establish whether a claim is fraudulent or not, it must undergo an investigation by fraud inspectors at the insurance company. These inspectors label the claims as fraudulent or non-fraudulent after an inquiry into the claim. However, due to time constraints and lack of resources, only a small fraction of all claims undergo such an investigation. This means that most claims are settled without an investigation. Hence, a number of fraudulent claims will remain undetected. We refer to claims that have been investigated as known, having a known label or as being labeled, whereas claims that were never investigated are referred to as unlabelled, unknown, or, having an unknown label. Per year in our dataset, less than a percent of all claims go through a formal fraud investigation with less than half resulting in a known fraud label.

Table 1: Description of the available intrinsic features for the motor insurance claims.

Type	Feature	Description
Target	`fraud`	Is the claim fraudulent: yes, no or unknown.
Policyholder characteristics	`age`	Age of policyholder when incident occurred.
	`responsibilityCode`	Policyholder's responsibility in the incident: at fault, shared responsibility or full right.
	`numContracts`	Number of contracts the policyholder has or has had with the insurer.
	`claimAge`	Number of months from beginning of contract to the date the incident occurred.
	`nClaims1`	Number of claims across all LoBs in the last year before the incident occurred.
	`nClaims5`	Number of claims across all LoBs in the last 5 years before incident occurred.
	`lastClaim`	Number of months since the last claim occurrence.
	`amount1`	Claimed amount across all LoBs in the last year before current claim occurrence.
	`amount5`	Claimed amount across all LoBs in the last 5 years before current claim occurrence.
	`refused1`	Number of times compensation was refused in the last year before current claim occurrence.
	`refused5`	Number of times compensation was refused in the last 5 years before current claim occurrence.
	`atfault1`	Number of times the policyholder had responsibility code 'at fault' in the last year before current claim occurrence.
	`atfault5`	Number of times the policyholder had responsibility code 'at fault' in the last 5 years before current claim occurrence.
	`samesits1`	Number of times the policyholder had the same responsibility code as the current claim in the last year before current claim occurrence.
	`samesits5`	Number of times the policyholder had the same responsibility code as the current claim in the last 5 years before current claim occurrence.
Claim characteristics	`claimAge`	Number of months from beginning of contract to the date the incident occurred.
	`people`	Number of people involved in the claim.
	`company`	Number of companies involved in the claim.
	`police`	Police was called when the incident happened: yes or no.
	`daysReport`	Number of days from the occurrence of incident to the filing of the claim.
	`amount`	The claimed amount for closed claims or the expected claimed amount if the claim is still open.

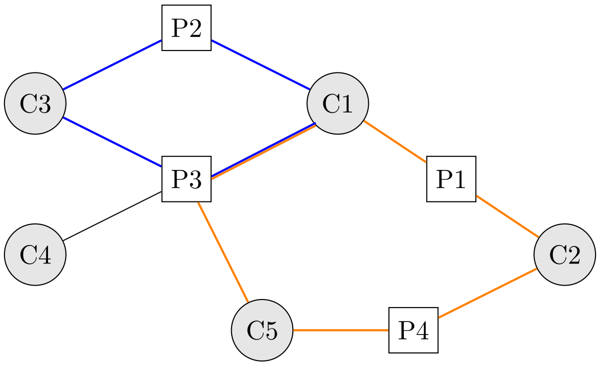


Figure 1: A sample network of claims and parties. The circles are claims and the rectangles are parties. The nodes connected with blue edges form a 4-cycle and the nodes connected with orange edges form a 6-cycle.

2.2 A bipartite network of insurance claims and involved parties

We use the claims and the involved parties to build a bipartite network, including both open and closed claims reported within a certain period. When a claim is filed, we link to it all the involved parties. The involved parties in a motor insurance claim could be, e.g., the policyholder who owns the car, the broker who solicits the claim and the garage where the car is repaired. Furthermore, we link each party to all the claims in which this party is involved. We refer to the network as a social network. Figure 1 shows a sample network of five claims and four parties. Claim C1 is linked to its involved parties, P1, P2 and P3. Each party is then also linked to the claims that they are involved in. Party P1 is linked to claim C2, party P2 is linked to claim C3, and party P3 is linked to claims C3, C4 and C5. This network is not restricted to motor insurance claims only, but includes claims from all lines of business in the non-life sector of the insurance company, i.e., motor, fire, liability and work accidents. As such, the social network represents a holistic view on the riskiness of clients.

The network is a bipartite network because it is composed of two types of nodes, i.e., claims (C) and parties (P), with no edges between nodes of the same type. Formally, we let $G = (C \cup P, E)$ be a bipartite network of nodes $C \cup P$ and edges E. Each node belongs to exactly one of the sets C and P and each edge in E connects one node in C to one node in P (Newman, 2010). The nodes in the set C correspond to insurance claims and the nodes in P are the various parties involved in the claims, that is, the policyholders, brokers, experts and garages. We let n_C be the number of nodes in C and denote individual nodes, i.e. claims, with c_i where $i \in \{1, \ldots, n_C\}$. Similarly, n_P is the number of nodes in P and individual nodes, i.e. parties, are given by $p_j, j \in \{1, \ldots, n_P\}$. The network's edges carry weights to indicate the strength of the connection between the two nodes they connect. All the edges are represented collectively by a weight matrix $\boldsymbol{W} = (w_{ij})$, where $i \in \{1, \ldots, n_C\}, j \in \{1, \ldots, n_P\}$, with n_C rows and n_P columns, corresponding to the nodes in C and P respectively. Thus, if node c_i is connected to node p_j the value of w_{ij} is positive and zero otherwise. When the network is unweighted, all non-zero values in $\boldsymbol{W}$ are equal to one. The network is undirected as no direction is given to the edges, that is $w_{ij} = w_{ji}, \forall i, j \in n_C \cup n_P$.

The k-th order neighborhood of a node c_i, denoted with $\mathcal{N}_{c_i}^k$, is the set of all nodes that are connected to c_i, via a path of exactly k edges. Thus, the first order neighborhood of a claim c_i is the set of all parties that are involved in the claim, or

$$\mathcal{N}_{c_i}^1 = \{p_j | w_{ij} \neq 0\},$$

whereas the second order neighborhood of a claim is the set of all claims that are involved in the parties in $\mathcal{N}_{c_i}^1$, or

$$\mathcal{N}_{c_i}^2 = \{c_k | p_j \in \mathcal{N}_{c_k}^1 \wedge w_{kj} \neq 0\} \setminus c_i.$$

The degree of a node, denoted with d_i for claim c_i or d_j for party p_j, is the number of nodes in the node's first order neighborhood for an unweighted network and the sum of weights on the edges between the node and the nodes in its first order neighborhood for a weighted network. We use the diagonal matrix $\boldsymbol{D_C}$ to denote the degree of all claims, i.e., $(\boldsymbol{D_C})_{ii} = d_i$, and similarly, the diagonal matrix $\boldsymbol{D_P}$ for the degree of all parties. The matrices $\boldsymbol{D_C}$ and $\boldsymbol{D_P}$ are square matrices of orders n_C and n_P, respectively.

Table 2 shows summary statistics for the nodes and edges in the network. Set C contains only one type of nodes, namely the claims. The average and median of the number of parties involved in a claim is 3.79 and 3, respectively. There are four types of party nodes, namely, policyholders, brokers, experts and garages. With a relative frequency of 96.36%, policyholders are the most common parties and they are connected to two claims on average. Broker nodes have a relative frequency of only 0.39% and yet some of them are very important in the network as is shown by their high edge relative frequency. Out of all the edges, 27.82% are connected to a broker and the broker degree ranges from one to almost twenty thousand. Experts show a similar trend in

Table 2: Summary statistics of nodes and edges in the sets C and P. The column *Nodes* shows the relative frequency of a specific node type within the set of parties. Similarly, the column *Edges* shows the relative frequency of edges connected to the respective node type in the set of all edges. As an edge always connects one claim and one party, the edge can be attributed to just the party, as is done in this table.

Set	Type	Nodes	Edges	Mean degree	Median degree	Min degree	Max degree
C	Claim	-	-	3.79	3	1	42
P	Policyholder	0.9636	0.4908	2.06	1	1	20274
	Broker	0.0039	0.2782	261.00	42	1	19830
	Expert	0.0037	0.1513	148.00	1	1	125951
	Garage	0.0288	0.0797	10.10	1	1	10436

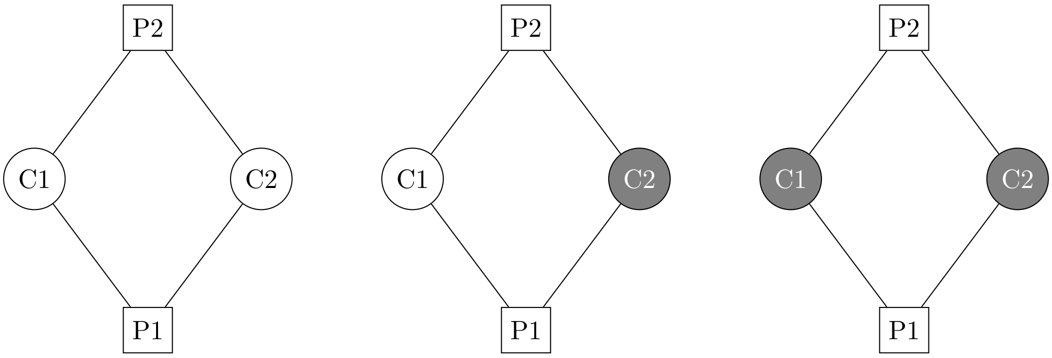


Figure 2: 4-cycles with zero (left), one (middle) or two (right) fraudulent claims. Fraudulent claims are colored dark gray and non-fraudulent claims are white.

terms of node relative frequency, but their edge relative frequency is lower. Finally, although there is a higher number of garages, their connectivity is less than that of brokers and experts. This can be explained by the fact that garages are almost only relevant for motor claims, while our network also incorporates other LoBs.

2.3 Fraudulent structures

A cycle in a network is a set of nodes connected via edges, where all nodes and edges are distinct and the last node is connected to the first node (Newman, 2010). In Figure 1, C1, P2, C3 and P3 form a four node cycle (blue edges), also called a 4-cycle. In addition, since claims C2 and C5 are both linked to party P4 there is a cycle of six nodes, made up of C1, P1, C2, P4, C5 and P3 (orange edges). We refer to these cycles as 6-cycles. These cyclic structures are examples of incidents involving multiple policyholders who file claims together.

The most simple structure in a bipartite network is a 4-cycle, as seen in Figure 2 where white

Table 3: Relative frequency of 4-cycles with zero, one and two fraudulent claims among 4-cycles with known labels.

		Party nodes Two people	 One person & one company	 Total
Number of fraudulent claim nodes	Zero	16.33%	36.56%	35.66%
	One	12.24%	30.29%	29.47%
	Both	71.43%	33.15%	34.87%

and dark gray circles represent non-fraudulent and fraudulent claims, respectively. In these structures zero, one or both claims can be fraudulent. Table 3 shows the relative frequency of each type of 4-cycle among all 4-cycles with known labels in our network. For 4-cycles where both party nodes are people (policyholder, broker or expert), over 70% have two fraudulent claims, 16.33% of 4-cycles have two legitimate claims and 12.24% have one claim with each label. Including 4-cycles where one of the parties can be a company, 34.87% and 35.66% of all 4-cycles have two fraudulent and two non-fraudulent nodes, respectively. In contrast, with a relative frequency of 29.47%, the ratio of 4-cycles with one claim of each label is lower. As such, in the 4-cycles with all labelled claims, fraudulent as well as non-fraudulent claims group together to some extent.

6-cycles can have zero, one, two or three fraudulent claims, as seen in Figure 3. We extract from the network all 6-cycles where all the claims have a known label and look at the distribution of fraudulent and non-fraudulent labels in these 6-cycles. When a labelled claim is part of such a 6-cycle, fraudulent claims appear on average in 1.6 6-cycles whereas non-fraudulent claims are on average in 1.4 6-cycles. Among all 6-cycles with at least one fraudulent claim, the relative frequency of 6-cycles with one, two and three fraudulent claims is 42.13%, 46.88% and 10.69%, respectively. Similarly, among all 6-cycles with at least one non-fraudulent claim, the relative frequency of 6-cycles with zero, one and two fraudulent claims is equal to 20.86%, 61.64% and 17.36%, respectively. As such, both fraudulent and non-fraudulent claims tend to group together in 6-cycles.

Table 4 shows the relative frequency of fraudulent and non-fraudulent claims in the second and fourth order neighborhoods of claims with a known label. Hereto, we calculate for each labelled claim the ratio of fraudulent and non-fraudulent claims relative to the size of the respective neighborhood, that is, including the unknown claims as well. In the second order neighborhood

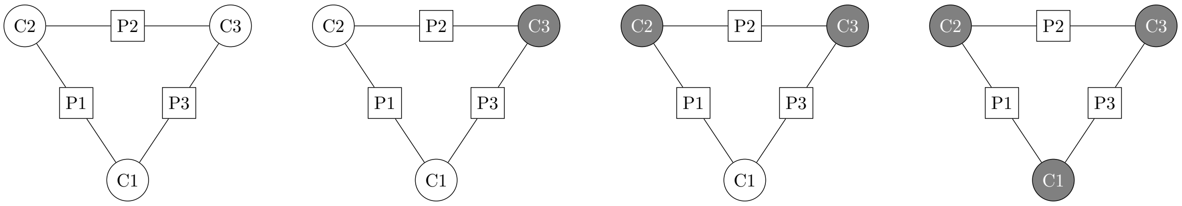


Figure 3: 6-cycles with zero, one, two and three fraudulent claims (from left to right). Fraudulent claims are colored dark gray and non-fradulent claims are white.

Table 4: Relative frequency of fraudulent and non-fraudulent claims among all claims in the second and fourth order neighborhoods of labelled claims.

		Relative frequency	
Neighborhood	Label	Fraudulent claims	Non-fraudulent claims
Second order	Non-fraudulent	0.154%	0.394%
	Fraudulent	0.269%	0.223%
Fourth order	Non-fraudulent	0.0884%	0.196%
	Fraudulent	0.0894%	0.135%

of non-fraudulent claims the ratio of non-fraudulent claims is 0.394%, on average, and the ratio of fraudulent claims is 0.154%, on average. In contrast, for fraudulent claims, the ratio of non-fraudulent claims is 0.223%, an average, and the ratio of fraudulent claims is 0.269%, on average, in the second order neighborhood. More fraudulent claims than non-fraudulent claims appear in the neighborhood of fraudulent claims and vice versa for the neighborhood of non-fraudulent claims. In the fourth order neighborhood, this also holds for non-fraudulent claims, but not for the fraudulent ones.

We find some empirical evidence of structural similarities in the network among fraudulent claims on the one hand and among non-fraudulent claims on the other hand. Claims with the same label tend to be more connected to each other. In addition, the claims with opposite labels are less connected to each other. This establishes some evidence for homophily, especially among the fraudulent claims (Park and Barabási, 2007).

However, finding new (yet undetected) fraudulent claims through exploration of the network alone is a complex task as Figure 4 illustrates. This excerpt from the real network shows ten parties and seven claims of which four are fraudulent (dark gray), two are non-fraudulent (white) and one is unknown (light gray). The expert on the left side is involved in 26 claims where six are known fraudulent and eleven are known non-fraudulent. The garage on the right has 501 claims where six are labelled as fraudulent and four are labelled as non-fraudulent. Let

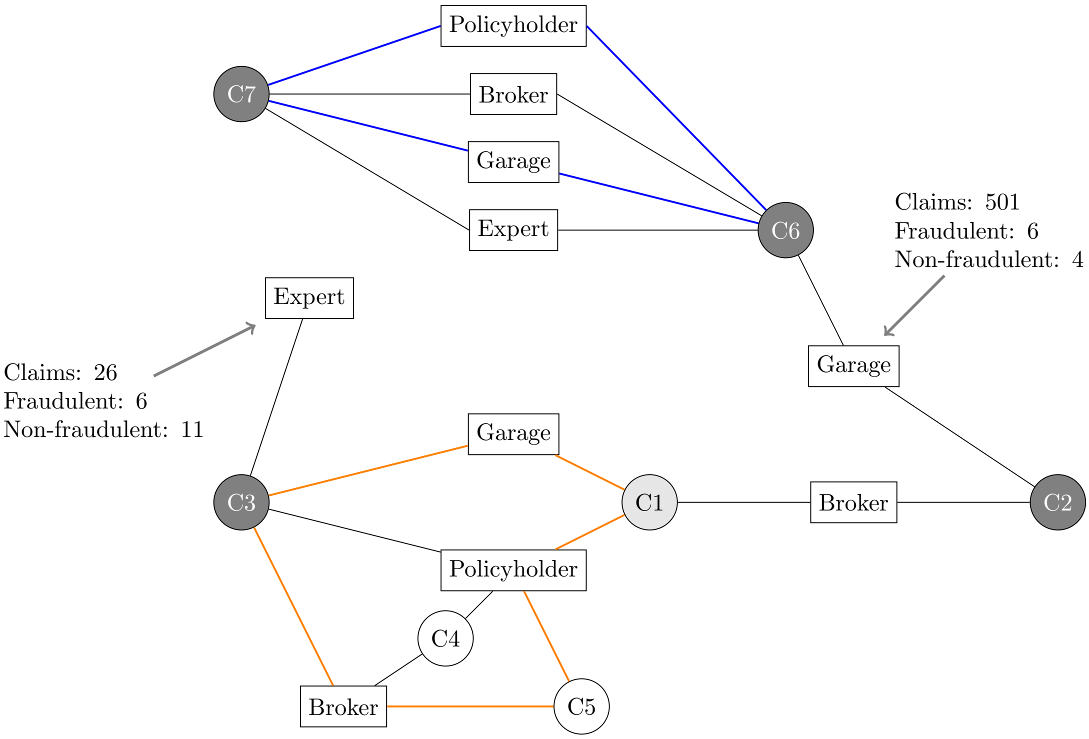


Figure 4: Excerpt from the real network, illustrating the network's complexity. In the network, the dark gray, light gray and white circles represent fraudulent, unknown and non-fraudulent claims, respectively. The nodes connected with blue edges form a 4-cycle and the nodes connected with orange edges form a 6-cycle.

C1 be a new claim that the claim handlers need to screen to decide whether it is suspicious or not. From the intrinsic features in Table 1 they can easily see that the policyholder has an older fraudulent claim. However, detecting the influence of other nodes in the network on the claim's fraud propensity is not obvious, and requires taking the network structure into account. For example, in this excerpt, C1 is only three edges away from the expert on the left who has a number of fraudulent claims. Moreover, the broker on the right is part of the first order neighborhood of C1 while being connected to a proven fraudulent claim involving a garage with a dubious reputation. Claim C1 forms a 4-cycle with the fraudulent claim C3 and a 6-cycle with fraudulent claim C3 and the non-fraudulent claim C5. Similarly, claim C2 is connected to a garage involved in 501 claims. It would not be feasible for the claim handlers to look at all those claims to establish connections to known fraudulent claims and to unravel structures such as the 4-cycle that exists between claims C6 and C7.

This example illustrates the complexity of manually exploring the network to find fraudulent claims and motivates the need for a more automated approach.

3 Supervised learning with intrinsic and network derived features

We develop an analytical model for flagging suspicious claims inspired by our empirical findings of homophily among the fraudulent claims in the network. In Section 3.1, we use a ranking algorithm based on the personalized PageRank of Page et al. (1999) to score the nodes in the network with respect to their exposure to known fraudulent claims. We then extract features from the network in Section 3.2 and combine them with the intrinsic features listed in Table 1 in a predictive model to identify the most suspicious claims in Section 3.3.

3.1 Computing fraud scores

The key principle of PageRank is to score —and as such to rank— a node in a network based on both the number of connecting nodes as well as the rank of the nodes that provide these connections. The algorithm can be personalized to bring out nodes that are important from the perspective of a specific set of source nodes, such as the known fraudulent claims in our problem setting. The resulting scores are then biased or personalized towards the source nodes, reflecting a preference for these nodes. He et al. (2017) formalized the (personalized) ranking of nodes in a bipartite network. We apply their BiRank algorithm to our network of claims and parties.

3.1.1 BiRank algorithm

The BiRank algorithm takes as inputs a network and a *query vector* and calculates a score or ranking for each node in the network. The query vector encodes some prior knowledge, such as fraud. In what follows, we use c_i and p_j to denote the score of node c_i and p_j, respectively. The collection of scores across all claim nodes is stored in $\mathbf{c}$, while $\mathbf{p}$ is the vector with the scores of the nodes of type party.

BiRank iteratively computes a node's score as a sum of the scores from the connected nodes. Let w_{ij} be a non-negative number expressing the weight on the edge between nodes c_i and p_j, where w_{ij} is 0 if the nodes are unconnected. Then

$$c_i = \sum_{j=1}^{n_P} w_{ij} p_j \quad \text{and} \quad p_j = \sum_{i=1}^{n_C} w_{ij} c_i$$

for claims and parties, respectively. To ensure convergence and stability, He et al. (2017) propose a symmetric normalization that smooths the edge weight by the degree of the two connected nodes simultaneously. This gives the normalized scores

$$c_i = \sum_{j=1}^{n_P} \frac{w_{ij}}{\sqrt{d_i}\sqrt{d_j}} p_j \quad \text{and} \quad p_j = \sum_{i=1}^{n_C} \frac{w_{ij}}{\sqrt{d_i}\sqrt{d_j}} c_i \tag{1}$$

where d_i (respectively d_j) is the degree of node c_i (respectively p_j), as introduced in Section 2.2. The normalization dampens the contribution of high degree nodes and gives better quality results (He et al., 2017).

Query vectors incorporate prior information or belief in the node ranking, and steer it towards the fraudulent claims, by factoring them directly into the ranking process. Let $\mathbf{c^0}$ and $\mathbf{p^0}$ be query vectors of length n_C, respectively n_P, for the claims and parties. To incorporate the query vectors into the ranking we expand (1) as

$$c_i = \alpha \sum_{j=1}^{n_P} \frac{w_{ij}}{\sqrt{d_i}\sqrt{d_j}} p_j + (1-\alpha) c_i^0 \quad \text{and} \quad p_j = \beta \sum_{i=1}^{n_C} \frac{w_{ij}}{\sqrt{d_i}\sqrt{d_j}} c_i + (1-\beta) p_j^0 \tag{2}$$

where $\alpha \in [0, 1]$ and $\beta \in [0, 1]$ are parameters that control the trade-off between the importance of the network structure and the query vector. Writing the BiRank equation (2) in matrix form gives

$$\mathbf{c} = \alpha \boldsymbol{S} \mathbf{p} + (1-\alpha)\mathbf{c}^0 \quad \text{and} \quad \mathbf{p} = \beta \boldsymbol{S}^T \mathbf{c} + (1-\beta)\mathbf{p}^0, \tag{3}$$

where $\boldsymbol{S} = \boldsymbol{D}_C^{-\frac{1}{2}} \boldsymbol{W} \boldsymbol{D}_P^{-\frac{1}{2}}$ is the symmetrically normalized weight matrix. When using the BiRank algorithm to compute fraud scores, the ranking vectors $\mathbf{c}$ and $\mathbf{p}$ are first randomly initialized and then computed iteratively until convergence to a unique stationary point, which is theoretically guaranteed as outlined in (He et al., 2017).

3.1.2 Fraud specialized query vector

In our insurance context, the goal is to rank the nodes relative to fraud using the information about known fraudulent claims to guide the process. The strategy is inspired by Van Vlasselaer et al. (2016) who used traditional personalized PageRank (instead of BiRank) to propagate fraud through the network. Most of the claim nodes in the network have an unknown label, but we use the known fraudulent claims as prior information in the query vector. However, only claims can be fraudulent, not parties. Therefore, we identify prior information for the nodes in C via $\mathbf{c}^0$, while $\mathbf{p}^0 \equiv \mathbf{0}$. We adjust (3) accordingly and set $\beta = 1$, since only the network structure matters in the absence of the query vector. This gives

$$\mathbf{c} = \alpha S\mathbf{p} + (1-\alpha)\mathbf{c}^0 \quad \text{and} \quad \mathbf{p} = S^T\mathbf{c}.$$

Algorithm 1 summarizes the update rules for the iterative BiRank algorithm. These rules are repeated until either the relative change in scores between two iterations is below a certain threshold or a maximum number of iterations is reached.

Algorithm 1: BiRank algorithm for computing fraud scores in a network of insurance claims and parties. Adapted from Algorithm 1 in (He et al., 2017). We omit the query vector $\mathbf{p}^0$ and set $\beta = 1$.

```
Input: Weight matrix $\boldsymbol{W}$, query vector $\mathbf{c}^0$ and hyperparameter $\alpha$;
Output: Ranking vectors $\mathbf{c}$ and $\mathbf{p}$;
Symmetrically normalize $\boldsymbol{W}$: $\boldsymbol{S} = \boldsymbol{D}_P^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}_C^{-\frac{1}{2}}$;
Randomly initialize $\mathbf{c}$ and $\mathbf{p}$;
while stopping criteria is not met do
    $\mathbf{c} \leftarrow \alpha\boldsymbol{S}\mathbf{p} + (1-\alpha)\mathbf{c}^0$;
    $\mathbf{p} \leftarrow \boldsymbol{S}^T\mathbf{c}$ ;
return c and p;
```

Figure 5 revisits the sample network in Figure 1 and shows the network before and after applying the BiRank algorithm with a query vector for fraud. Claim C4 is known to be fraudulent and claim C2 is known to be non-fraudulent. The other claims have an unknown label. The query vector specifies claim C1 as fraudulent, i.e., $c_4^0 = 1$, while the other claims do not contribute any information in the query vector, i.e., $c_1^0 = c_2^0 = c_3^0 = c_5^0 = 0$. We set $\alpha = 0.85$ in Algorithm 1. After running the BiRank algorithm on this unweighted network, all nodes obtain a fraud score, indicated by the numbers in blue.

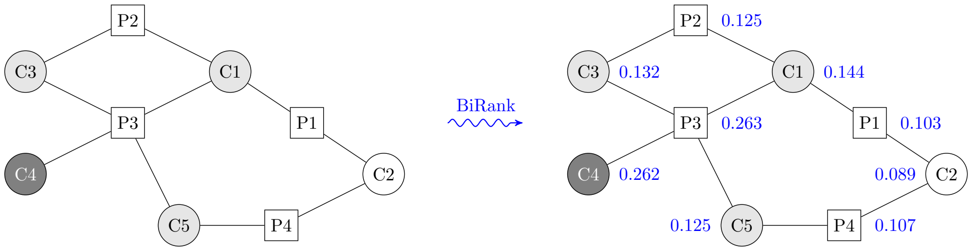


Figure 5: A sample network before and after applying the BiRank algorithm to rank the nodes with respect to fraud. Claim C4 is a known fraud and the source of fraud influence. Node C2 is known non-fraud. The network is unweighted and $\alpha = 0.85$. The fraud scores are shown in blue.

Table 5: Features extracted from the network structure, the labels and the fraud scores.

Name	Order	Description
`scores0`	0	The node's fraud score
`n1.q1`	1	The first quartile of the empirical distribution of the fraud scores in the node's first order neighborhood
`n1.med`	1	The median of the empirical distribution of the fraud scores in the node's first order neighborhood
`n1.max`	1	The maximum of the empirical distribution of the fraud scores in the node's first order neighborhood
`n2.q1`	2	The first quartile of the empirical distribution of the fraud scores in the node's second order neighborhood
`n2.med`	2	The median of the empirical distribution of the fraud scores in the node's second order neighborhood
`n2.max`	2	The maximum of the empirical distribution of the fraud scores in the node's second order neighborhood

3.2 Network featurization

Based on the network structure, the labels and the computed fraud scores, we can extract several features from the network. This process is known as network featurization. We distinguish score features and neighborhood features as motivated in Section 3.2.1 and 3.2.2, respectively. We refer to the set of extracted features as *network features*.

3.2.1 Features based on fraud score

Inspired by Van Vlasselaer et al. (2016) we compute network features from the fraud scores in the zeroth, first and second order neighborhoods of each node. Table 5 shows the list of extracted *score features*. The zeroth order neighborhood of a node is the node itself and the feature we extract is the node's fraud score. For the first and second order neighborhood features, we look at the distribution of the fraud scores in the respective neighborhoods of the claim and compute three features in each case. These are the scores' first quartile, the median and the maximum,

Table 6: Score feature values for claim C1 in the sample network in Figure 5.

Name	Order	Value
`scores0`	0	0.1440
`n1.q1`	1	0.1140
`n1.med`	1	0.1250
`n1.max`	1	0.2630
`n2.q1`	2	0.1160
`n2.med`	2	0.1285
`n2.max`	2	0.2620

Table 7: Features extracted from the neighborhood structure and the labels.

Name	Order	Description
`n1.size`	1	The number of nodes in the node's first order neighborhood
`n2.size`	2	The number of nodes in the node's second order neighborhood
`n2.ratioFraud`	2	The number of known fraudulent claims in the node's second order neighborhood divided by `n2.size`
`n2.ratioNonFraud`	2	The number of known non-fraudulent claims in the node's second order neighborhood divided by `n2.size`
`n2.binFraud`	2	A binary value indicating whether there is a known fraudulent claim in the node's second order neighborhood

capturing the variation in the exposure to fraud. For example, a high maximum indicates close proximity to at least one fraudulent node whereas a high first quartile indicates proximity to several fraudulent nodes. The first order neighborhood typically consists of only a few nodes whereas the second order neighborhood is much larger and thus provides more variation in scores and the resulting features. For example, returning to the sample network in Figure 5, claim C1 has the score feature values shown in Table 6.

3.2.2 Features based on neighborhood

Table 7 lists the extracted *neighborhood features*. First, we look at the number of nodes in a node's neighborhood. The size of the first order neighborhood is the number of parties that are involved in the claim while the size of the second order neighborhood is the number of claims in which these involved parties are engaged in. Second, following the framework of Lu and Getoor (2003) we compute three so-called link-based features using the labels of claims in a node's second order neighborhood. These are the ratio of known fraudulent as well as non-fraudulent claims, versus the size of the neighborhood. These represent the homophilic nature of fraud. The third feature indicates whether one of the involved parties is linked to a fraudulent claim, i.e., whether there is a fraudulent claim in its second order neighborhood. This feature is one of the business rules used by the insurance company to flag suspicious claims for further

Table 8: Neighborhood feature values for claim C1 in the sample network in Figure 5.

Name	Order	Value
`n1.size`	1	3
`n2.size`	2	4
`n2.ratioFraud`	2	0.25
`n2.ratioNonFraud`	2	0.25
`n2.binFraud`	2	1

investigation. The values of the extracted neighborhood features for claim C1 in Figure 5 are listed in Table 8.

3.3 Fraud analytical model

We build a supervised learning model to detect suspicious claims. The model can use features from three groups; the intrinsic features described in Section 2.1, the score features and the neighborhood features described in Section 3.2. The features in each group are denoted with $x^{\mathtt{intr}}$, $x^{\mathtt{score}}$ and $x^{\mathtt{nbh}}$, respectively.

Three different labels are present in the data, namely fraud, non-fraud and unknown. We adopt a one-vs.-all classification strategy by transforming the label to a binary target feature (Bishop, 2006). We do this in two ways. First, we build a model that distinguishes claims with a known label from those with an unknown label. Hereto, we create the target feature $y^{\mathtt{known}}$ by setting

$$y_i^{\mathtt{known}} = \begin{cases} 1 & \Leftrightarrow l_i \in \{\text{fraud, non-fraud}\} \\ 0 & \Leftrightarrow l_i \in \{\text{unknown}\}, \end{cases}$$

with l_i the label of claim c_i. This results in the dataset $\mathcal{D}^{\mathtt{known}}$ with features and target

$$x = \{x^{\mathtt{intr}}, x^{\mathtt{score}}, x^{\mathtt{nnbh}}\}, \qquad y = y^{\mathtt{known}}.$$

The claims where the target feature equals 1, are the claims that underwent a fraud investigation. These claims were considered suspicious by the claim handlers, possibly because they fulfilled some of the currently implemented business rules to highlight fraud. Analysis of this dataset will give insights into how claims are selected for further investigation at the insurance company. The findings of a supervised learning model for $y^{\mathtt{known}}$ may also inspire the creation of new, additional business rules.

Second, we build a model that distinguishes claims with a known *fraud* label from the rest of the claims. We create the target feature $y^{\texttt{fraud}}$ by setting

$$y_i^{\texttt{fraud}} = \begin{cases} 1 & \Leftrightarrow l_i \in \{\text{fraud}\} \\ 0 & \Leftrightarrow l_i \in \{\text{non-fraud, unknown}\}. \end{cases}$$

This results in the dataset $\mathcal{D}^{\texttt{fraud}}$ with features and target

$$x = \{x^{\texttt{intr}}, x^{\texttt{score}}, x^{\texttt{nnbh}}\}, \qquad y = y^{\texttt{fraud}}.$$

This dataset will help in distinguishing fraudulent claims from the rest of the claims. We use it to investigate which features are discriminative for fraud.

We analyze the data sets $\mathcal{D}^{\texttt{known}}$ and $\mathcal{D}^{\texttt{fraud}}$ with two supervised learning techniques, namely logistic regression and random forests.

Logistic regression is a statistical technique used to model the probability of a certain event or class, in our case whether a given claim is fraudulent. Given a training set, logistic regression expresses the probability of success, $P(fraud)$, as follows:

$$P(fraud) = \frac{1}{1 + e^{-(b_0 + \sum b_j x_j)}}$$

where x_j are the features in the data set and b_0, b_j are the parameters to be estimated by the model. We focus on logistic regression for model building because of its popularity in the insurance industry and its interpretability. This is in line with our goal to understand which features are predictive for the occurrence of the target variable. These discriminative characteristics can then be transferred into business rules that claim handlers can easily understand and deploy when deciding whether an incoming claim should be flagged for investigation.

In contrast, random forests are powerful ensemble models that are capable of discovering complex patterns in the data (Breiman, 2001). Ensemble methods estimate multiple models instead of using just one. Random forests create a bag (called ‘forest’) of decision trees during training that together predict the outcome. To avoid overfitting, random forests feature additional elements of randomness. Firstly, each decision tree is trained on a bootstrap copy of the training data set. Secondly, when performing splits of nodes in the decision trees, only a random subset of features is considered. In this way, diverse and independent base models are created. To make a prediction, the random forest outputs the class that the majority of the trees predicted. This results in reduced generalization error of the prediction of the final model. Random forests are black box models, because it is not obvious how a prediction is made. However, by investigating which features tend to give informative splits in the decision trees, it is possible to estimate their importance with respect to predicting the target outcome. As we have multiple and varied features, we use random forests to evaluate their importance before applying logistic regression to the most informative features.

3.4 Model evaluation

We select three measures to evaluate the performance of the fraud models as described below.

Receiver operating characteristic curve The area under the receiver operating characteristic curve (AUROC) is commonly used to measure the performance of binary classifiers (Hanley and McNeil, 1982). It measures the trade-off between the model's recall (or sensitivity) and specificity while varying the cut-off that determines whether an instance is classified as good or bad by the model. It summarizes the performance in a single number between 0.5 (random model) and 1 (perfect model), where a higher value indicates a better performance. For fraud, recall measures the ratio between fraudulent claims that are detected by the model and the total number of fraudulent claims. Specificity, on the other hand, is the number of non-fraudulent claims that the model predicts as non-fraudulent divided by the number of non-fraudulent claims in the data.

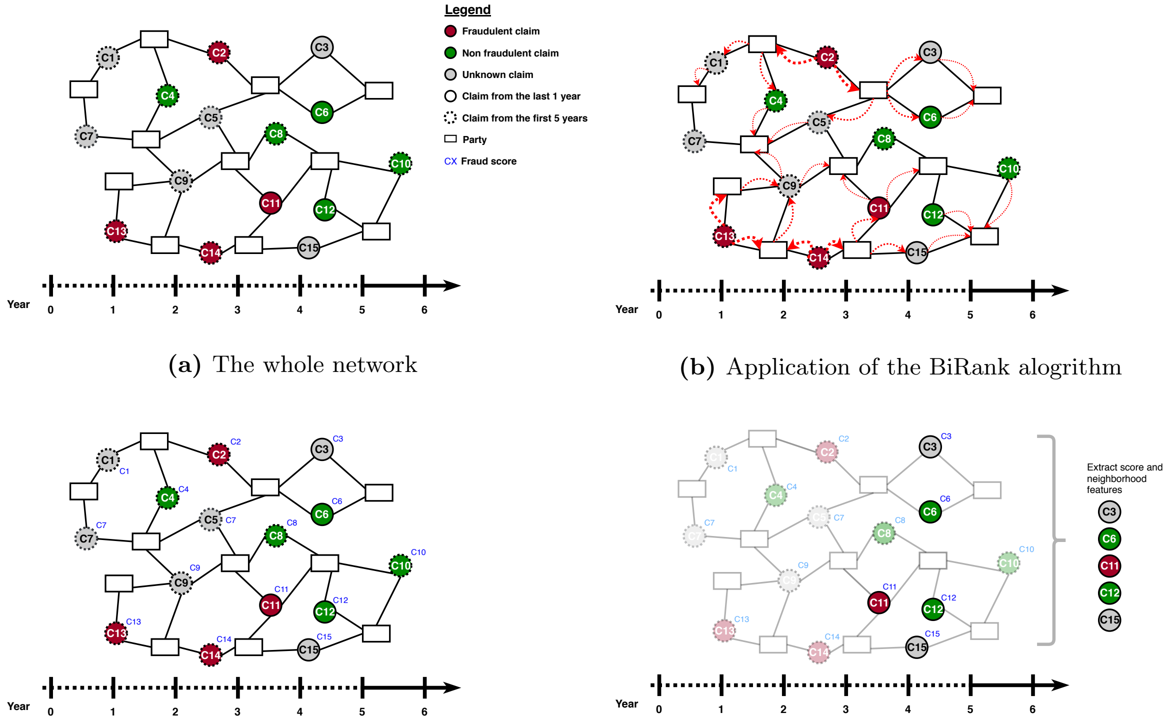
Precision-Recall curve The area under the precision-recall curve (AUPR) is a performance measure for binary classifiers that is useful when the class distribution of the dataset is imbalanced (Saito and Rehmsmeier, 2015). In contrast to AUROC, it is based on the model's precision and recall. In fraud detection, precision measures the ratio of fraudulent claims that are detected by the model and the number of claims that the model predicts as fraudulent. The precision-recall curve shows the trade-off between the recall and the precision and represents the model's performance independent of the cut-off. The area under the curve, AUPR, summarizes the curve in a single number. The AUPR of a perfect model is one whereas AUPR of a random model equals the ratio of the positive class or, in our case, the fraud ratio.

Top decile lift Lift is a performance measure that represents how much better a model is at detecting fraudulent claims than a random model (Lemmens and Croux, 2006; Verbeke et al., 2014). The top decile lift (TDL), for example, is computed by dividing the proportion of fraudulent claims among the 10% of claims with the highest predicted probability with the relative frequency of the fraudulent claims in the dataset. A random model has TDL equal to one, and increasing TDL values indicate a better performance. The TDL performance measure is valuable from a managerial perspective as it focuses on the most critical group of claims, i.e., those with the highest fraud probability. It enables us to control whether the targeted segment of suspicous claims indeed contains actual fraudulent claims.

4 Experimental set up

To evaluate the performance of our proposed technique, we set up experiments reflecting the deployment of the model in practice. Our goal is to compare the intrinsic, score and neighborhood features in their ability to detect fraud on the one hand and labelled claims on the other hand. We first apply the BiRank algorithm to calculate fraud scores, then we extract network features and finally we build predictive models using intrinsic and neighborhood features as well as score features.

Our model evaluation procedure requires careful design since it has to reflect the use and limitations of the predictive model in a practical setting. For example, when calculating the fraud scores the fraudulent claims that serve as the source of information in the query vector $\mathbf{c}^0$, will by design obtain a high score when the BiRank algorithm is applied. Therefore, they can not be included in the dataset for which we build the supervised fraud detection model. To circumvent this problem, and to reflect how the model would be deployed in reality, we view the older claims as historic data and include fraud information about older fraudulent claims in the query vector $\mathbf{c}^0$. However, we use the newer claims as targets when building the analytical model.



(a) The whole network

(b) Application of the BiRank alogrithm

(c) Nodes with fraud scores

(d) The most recent nodes

Figure 6: The set up for the BiRank algorithm and feature extraction. Red, green and gray circles represent fraudulent, non-fraudulent and unknown claims respectively. Rectangles represent parties. Whole and dashed lines around claims represent older and newer claims. Fraud scores are in blue.

Computation of fraud scores and network featurization The procedure to compute the fraud scores is shown in Figure 6. The claims in the data span a period of six years and we use all these claims and the involved parties to build the network. This is shown in Figure 6a. We use the first five years of the observation period as historic knowledge about fraud and the last, most recent, observation year for model building and evaluation. Red, green and gray nodes denote claims with the labels fraud, non-fraud and unknown, respectively. The rectangles denote the parties. A dashed line around a node means it was filed in the first five years and a whole line indicates the claim was filed in the most recent year.

We apply the BiRank algorithm, where the fraud specialised query vector $\mathbf{c}^0$ contains information about the known fraudulent claims of the first five years only. Figure 6b shows how the fraud influence originates from the red fraud nodes with dashed border and spreads through the network as the red arrows indicate. After the algorithm is applied, all nodes obtain a score. This is indicated in Figure 6c.

Our analytical model uses only the claims that were filed in the most recent year. Therefore, we extract score and neighborhood features as described in Section 3.2 for these claims only, see Figure 6d, and use these in our model building and evaluation.

Analytical model The datasets $\mathcal{D}^{\texttt{known}}$ and $\mathcal{D}^{\texttt{fraud}}$ are created by combining all labelled claims with a random sample of about 20 thousand claims that were filed in the most recent year. We start by setting aside a test set from both $\mathcal{D}^{\texttt{known}}$ and $\mathcal{D}^{\texttt{fraud}}$. They contain a random sample with 30% of the observations in each dataset while maintaining the original class balance. These datasets are used when measuring the out-of-sample performance of the final models. The remaining 70% of the observations make up the training datasets.

The training datasets are highly imbalanced, with a 4.9% and 1.8% minority class rate in $\mathcal{D}^{\texttt{known}}$ and $\mathcal{D}^{\texttt{fraud}}$, respectively. We use the SMOTE sampling technique to over sample the minority class and under sample the majority class (Chawla et al., 2002). As such, the ratio of the minority class in each sampled dataset is increased to 15%. We use these sampled datasets to evaluate the features' importance using random forests. We first tune the hyper-parameters of the random forests using ten-fold cross-validation on the training sets. We use grid search to find the optimal parameter values for the number of trees in the forest and the number of features sampled in each split. The search spaces for these two parameters are $\{100, 300, 500, 700, 900\}$ trees and $\{1, 3, \ldots, \text{NoF}\}$ features, where NoF is the number of features being considered. Using the optimal parameter values, we build random forest models for each of the two datasets and each feature group $x^{\texttt{intr}}$, $x^{\texttt{score}}$ and $x^{\texttt{nbh}}$. We then assess the importance of each feature in the constructed models. Subsequently, we build logistic regression models where we add one feature at a time based on the features' importance, starting with the most important one. To evaluate the performance of these logistic regression models we again use ten-fold cross-validation on the training sets. We inspect the performance increase when stepwise adding features. Finally, we build logistic regression models on the complete training set with the best performing features and evaluate the performance on the test datasets.

5 Results

5.1 Performance of features

The four plots in Figure 7 show the feature importance for each group of features ($x^{\texttt{intr}}$, $x^{\texttt{score}}$ and $x^{\texttt{nbh}}$) and each dataset ($\mathcal{D}^{\texttt{known}}$ and $\mathcal{D}^{\texttt{fraud}}$). We normalize the feature importance values such that they sum to one to give a clear indication of the relative contribution of each feature. The red bars denote the feature importance for the dataset $\mathcal{D}^{\texttt{fraud}}$ and the green bars refer to the dataset $\mathcal{D}^{\texttt{known}}$.

We detect some clear differences between the two datasets when using only intrinsic features. The features `amount`, `daysReport` and `police` are relatively more important in the dataset $\mathcal{D}^{\texttt{known}}$, whereas the features `claimAge`, `numContracts` and `pAge` are more important in $\mathcal{D}^{\texttt{fraud}}$. The most important score features for the dataset $\mathcal{D}^{\texttt{known}}$ are the maximum scores in the first and second order neighborhood, whereas the fraud score is most important for $\mathcal{D}^{\texttt{fraud}}$. For the neighborhood features, the difference in feature importance between the two datasets is less obvious. However, we observe clear differences in the importance of features when they are combined in the random forest models. The most important features for the dataset $\mathcal{D}^{\texttt{known}}$ are intrinsic features, e.g., `amount`, `daysReport`, `numContracts`, `pAge` and `police`, whereas the most important features for $\mathcal{D}^{\texttt{fraud}}$ are score and neighborhood features, most notably `scores0`, `n1.size` and `n1.q1`. Furthermore, intrinsic features are overall more important in $\mathcal{D}^{\texttt{known}}$ than

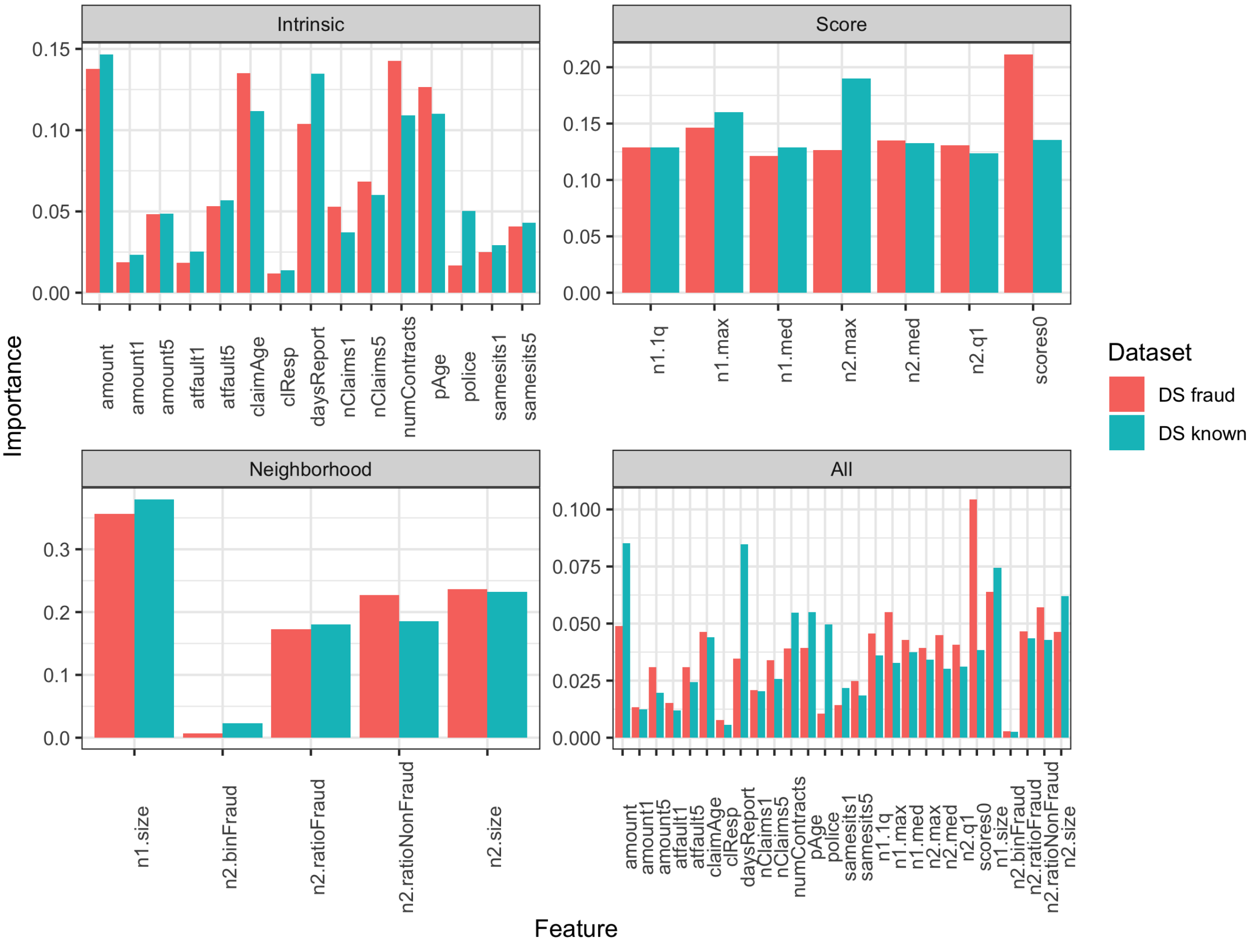


Figure 7: The importance of features in the random forest models using the two datasets ($\mathcal{D}^{\texttt{known}}$ and $\mathcal{D}^{\texttt{fraud}}$) and the four groups of features ($x^{\texttt{intr}}$, $x^{\texttt{score}}$ and $x^{\texttt{nbh}}$).

Table 9: Significant features in the logistic regression models with only intrinsic features.

	$\mathcal{D}^{\texttt{known}}$					$\mathcal{D}^{\texttt{fraud}}$			
	Estimate	Std. Error	z value	Pr(> |z|)		Estimate	Std. Error	z value	Pr(> |z|)
(Intercept)	-2.2511	0.0372	-60.57	0.0000	(Intercept)	-1.9177	0.0521	-36.83	0.0000
`amount`	0.1770	0.0269	6.59	0.0000	`numContracts`	0.1043	0.0468	2.23	0.0257
`daysReport`	-0.6346	0.0314	-20.24	0.0000	`amount`	0.3627	0.0451	8.04	0.0000
`claimAge`	-0.3509	0.0328	-10.70	0.0000	`claimAge`	-0.4830	0.0517	-9.34	0.0000
`pAge`	-0.0443	0.0305	-1.45	0.1461	`daysReport`	-0.2937	0.0482	-6.09	0.0000
`numContracts`	0.2422	0.0290	8.36	0.0000	`atfault5`	0.2727	0.0901	3.03	0.0025
`nClaims5`	-0.1788	0.0860	-2.08	0.0375	`samesits5`	-0.2994	0.0923	-3.24	0.0012
`atfault5`	-0.2319	0.0945	-2.45	0.0141					
`police1`	1.4167	0.0645	21.98	0.0000					
`amount5`	0.1805	0.0673	2.68	0.0074					
`samesits5`	0.1903	0.0664	2.87	0.0041					

Table 10: Significant features in the logistic regression models with only score features.

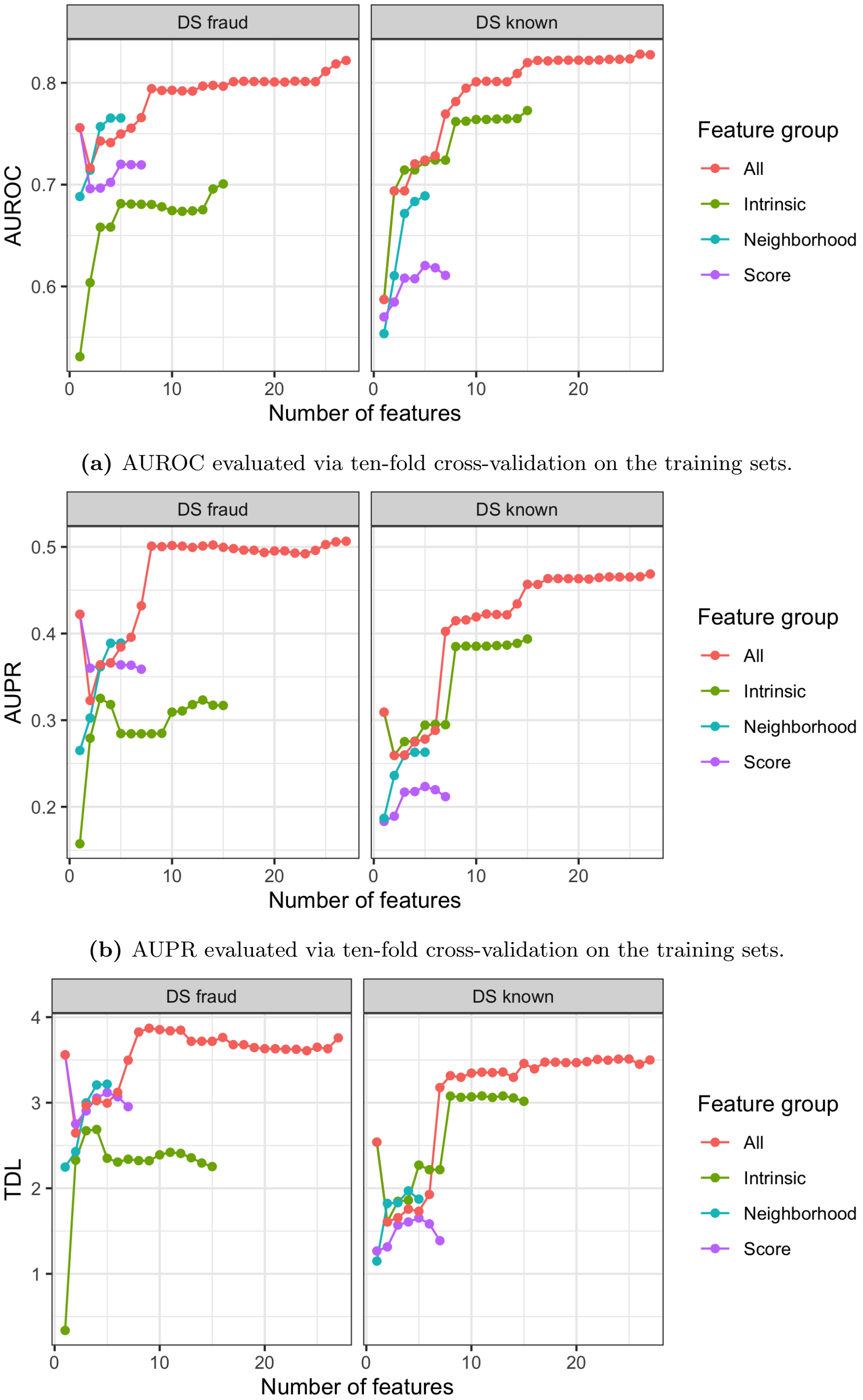
	$\mathcal{D}^{\texttt{known}}$					$\mathcal{D}^{\texttt{fraud}}$			
	Estimate	Std. Error	z value	Pr(> |z|)		Estimate	Std. Error	z value	Pr(> |z|)
(Intercept)	-1.7427	0.0276	-63.12	0.0000	(Intercept)	-1.9211	0.0521	-36.86	0.0000
`n2.max`	0.3075	0.0435	7.07	0.0000	`scores0`	0.2697	0.0558	4.83	0.0000
`n1.max`	-0.1042	0.0365	-2.85	0.0043	`n1.max`	0.3615	0.0487	7.43	0.0000
`scores0`	0.1847	0.0246	7.50	0.0000	`n2.med`	1.2744	0.2926	4.35	0.0000
`n2.med`	-0.4620	0.1618	-2.86	0.0043	`n2.q1`	-0.8991	0.2831	-3.18	0.0015
`n1.1q`	-0.2348	0.0400	-5.88	0.0000	`n1.1q`	-0.2336	0.0940	-2.48	0.0130
`n2.q1`	0.5190	0.1660	3.13	0.0018	`n1.med`	-0.1321	0.0769	-1.72	0.0858

in $\mathcal{D}^{\texttt{fraud}}$, while score features matter more in $\mathcal{D}^{\texttt{fraud}}$ than in $\mathcal{D}^{\texttt{known}}$.

Next we evaluate the prediction of the target in each dataset using logistic regression with the most important features. We start by including only the most important feature and then add the features in decreasing order of importance according to the random forest models previously constructed. The plots in Figure 8 show the performance measured in AUROC, AUPR and TDL evaluated via ten-fold cross-validation on the constructed training sets. Clearly, in $\mathcal{D}^{\texttt{known}}$ the performance of the model with only intrinsic features is greater than in $\mathcal{D}^{\texttt{fraud}}$. In contrast, the models using only score and neighborhood features perform better in $\mathcal{D}^{\texttt{fraud}}$ compared to $\mathcal{D}^{\texttt{known}}$.

Table 11: Significant features in the logistic regression models with only neighborhood features.

	$\mathcal{D}^{\texttt{known}}$					$\mathcal{D}^{\texttt{fraud}}$			
	Estimate	Std. Error	z value	Pr(> |z|)		Estimate	Std. Error	z value	Pr(> |z|)
(Intercept)	-2.4184	0.0912	-26.51	0.0000	(Intercept)	-2.1173	0.0587	-36.07	0.0000
`n1.size`	0.6111	0.0407	15.02	0.0000	`n1.size`	1.1005	0.0740	14.87	0.0000
`n2.size`	-0.7372	0.0466	-15.84	0.0000	`n2.size`	-0.6127	0.0726	-8.44	0.0000
`n2.ratioNonFraud`	0.3002	0.0281	10.67	0.0000	`n2.ratioNonFraud`	0.3464	0.0444	7.80	0.0000
`n2.ratioFraud`	0.0971	0.0281	3.46	0.0005	`n2.ratioFraud`	0.3621	0.0495	7.32	0.0000
`n2.binFraud`	0.7459	0.1017	7.34	0.0000					



(a) AUROC evaluated via ten-fold cross-validation on the training sets.

(b) AUPR evaluated via ten-fold cross-validation on the training sets.

(c) TDL evaluated via ten-fold cross-validation on the training sets.

Figure 8: The performance of logistic regression models for the two datasets ($\mathcal{D}^{\texttt{known}}$ and $\mathcal{D}^{\texttt{fraud}}$) and the four groups of features ($x^{\texttt{intr}}$, $x^{\texttt{score}}$ and $x^{\texttt{nbh}}$). We add the features in decreasing order of importance, starting with the most important feature according to the random forest models previously constructed.

Table 12: Significant features in the logistic regression models with all features.

	$\mathcal{D}^{\texttt{known}}$					$\mathcal{D}^{\texttt{fraud}}$			
	Estimate	Std. Error	z value	Pr(> |z|)		Estimate	Std. Error	z value	Pr(> |z|)
(Intercept)	-2.5352	0.0436	-58.15	0.0000	(Intercept)	-2.3559	0.0657	-35.86	0.0000
`amount`	0.1040	0.0274	3.80	0.0001	`n1.size`	1.1768	0.0789	14.92	0.0000
`daysReport`	-0.7116	0.0355	-20.05	0.0000	`n2.ratioNonFraud`	0.2645	0.0446	5.93	0.0000
`n1.size`	0.6186	0.0468	13.23	0.0000	`n1.max`	1.2969	0.1129	11.49	0.0000
`n2.size`	-2.1622	0.0972	-22.23	0.0000	`amount`	0.1840	0.0466	3.95	0.0001
`pAge`	-0.0539	0.0321	-1.68	0.0931	`n2.ratioFraud`	-0.1251	0.0662	-1.89	0.0590
`numContracts`	0.1874	0.0304	6.17	0.0000	`claimAge`	-0.3384	0.0528	-6.41	0.0000
`police1`	1.5206	0.0687	22.13	0.0000	`n2.size`	-2.2489	0.1620	-13.88	0.0000
`claimAge`	-0.2967	0.0334	-8.89	0.0000	`n1.med`	0.0890	0.0551	1.62	0.1060
`n2.ratioFraud`	-0.1833	0.0424	-4.33	0.0000	`n2.max`	0.5301	0.0991	5.35	0.0000
`n2.ratioNonFraud`	0.1441	0.0312	4.62	0.0000					
`n2.max`	0.7173	0.0565	12.70	0.0000					
`n1.max`	0.9963	0.0727	13.71	0.0000					

The final models are found using stepwise logistic regression with a combination of forward and backward selection of features on the training sets. Tables 9, 10, 11 and 12 show the results for the two datasets and the four groups of features. Less features are significant in the models for $\mathcal{D}^{\texttt{fraud}}$ compared to $\mathcal{D}^{\texttt{known}}$. The score features dominate the logistic regression model for $\mathcal{D}^{\texttt{fraud}}$ fitted on the combined set of all features.

Even though multi-collinearity of the features, in particular the extracted network features, complicates the interpretation of the sign and size of a calibrated effect, we extract some interesting findings from the output of the logistic regression models. Notably, in the intrinsic features models, see Table 9, `atfault5` and `samesits5` have opposite effects in the two datasets. For example, `atfault5` has a negative coefficient in $\mathcal{D}^{\texttt{known}}$, but a positive coefficient in $\mathcal{D}^{\texttt{fraud}}$. Policyholders who have been at fault in multiple claims in the past years are more likely to file another fraudulent claim. Similarly, `samesits5` has a positive coefficient in $\mathcal{D}^{\texttt{known}}$ but a negative coefficient in $\mathcal{D}^{\texttt{fraud}}$. In addition, the feature `police1`, which indicates that the police was called to the scene of the incident, is significant with a positive sign in $\mathcal{D}^{\texttt{known}}$ but it is not significant in $\mathcal{D}^{\texttt{fraud}}$.

Table 10 shows the results for the models with only score features, where for example both maximum statistics are significant in $\mathcal{D}^{\texttt{known}}$. They are also the most important features according to the random forest model. In $\mathcal{D}^{\texttt{fraud}}$, the `scores0` feature is most important and has a positive coefficient, which indicates that a higher fraud score detects fraudulent claims to a greater extent. Comparing these results to Figure 8 the `scores0` feature has a lot of predictive power on its own, as it is the first feature that is added to the model. The performance drops

Table 13: Performance of the four models on the test datasets.

	$\mathcal{D}^{\texttt{known}}$			$\mathcal{D}^{\texttt{fraud}}$		
Features	AUROC	AUPR	TDL	AUROC	AUPR	TDL
Intrinsic	0.691	0.1214	2.85	0.662	0.0301	2.137
Score	0.634	0.0883	2.25	0.660	0.0402	2.812
Neighborhood	0.681	0.1051	2.65	0.719	0.0481	3.262
All	0.725	0.1312	3.457	0.792	0.0810	3.824

when more features are added, due to the multi-collinearity of the score features.

For the neighborhood features in Table 11, the indicator feature `n2.binFraud` is significant for detecting the labeled claims but not for detecting fraudulent claims.

Table 12 displays the results for the models with all features. In $\mathcal{D}^{\texttt{known}}$ twelve features are significant, with half of these being intrinsic features. Nine features are significant in $\mathcal{D}^{\texttt{fraud}}$ and only two of them are intrinsic features.

In a final step we evaluate the logistic regression models on the test datasets. Table 13 shows that the intrinsic model is better for dataset $\mathcal{D}^{\texttt{known}}$, while both network models perform better on $\mathcal{D}^{\texttt{fraud}}$. Combining the different types of features available leads to the best performance on the test set, both for $\mathcal{D}^{\texttt{known}}$ as well as $\mathcal{D}^{\texttt{fraud}}$.

The approach is feasible for deployment as it performs well in terms of runtime. Even though all the claims and involved parties as registered over a period of six years were used to build the social network, running the BiRank algorithm to score the most recent claims took about eight minutes for each value of α on a standard business server. The training of the algorithms (logistic regression and random forests) is not time consuming since we work with tabular datasets with carefully engineered features (only those listed in Tables 1, 5 and 7), with 10-fold cross validation the total training and validation time was about 32 minutes. The extraction of score and neighborhood features was the most time consuming part and took about five hours for the entire dataset. In practice, the model would be used to score new claims on a weekly or monthly basis to mark suspicious claims, in which case the number of new claims is less and runtime is reasonable. Since these computations would be run overnight in practice, this is not an issue for the insurance company we worked with. All programming and data analysis was performed in R.

6 Conclusion

The insurance industry is increasingly relying on data science and machine learning to provide novel insights in its practices, such as for pricing and reserving. Insurance companies possess a lot of data about past claims, as well as policyholders and their claim history. In that data lies knowledge, which goes beyond simple statistics, that can be leveraged for more accurate data-driven predictions for the companies to better manage their practices and accommodate their heterogeneous customer base with tailor made policies and tariffs. Data science applied for fraud detection is another opportunity for the companies to find hidden patterns in the data, which ultimately will help them save money and strengthen their position, by giving their customers better service and lower tariffs.

This study presents a novel approach for fraud detection in insurance using social network analytics. We leverage an insurance company's database of claims, policyholders, brokers, experts and garages to build a bipartite network that represents the social structure between the various parties involved in claims. Taking a holistic view across multiple lines of business, we unravel stronger ties among fraudulent claims when compared to the links between fraudulent and non-fraudulent claims. We use this empirical finding to rank claims with respect to fraud using a personalized PageRank algorithm (in casu: BiRank with a fraud specific query vector). The resulting fraud scores bring added value when detecting fraud and so are the score neighborhood-structure features derived from the network. Our experiments reveal a clear difference between the features that are predictive for claims with a known label (i.e., investigated claims) versus claims with a confirmed fraud label. The classical intrinsic features registered for claims and policyholders are good at distinguishing claims with a known label, which is intuitive as these are traditionally used by the insurance company to flag suspicious claims for further investigation. Features distinguishing fraudulent claims from the other claims in the dataset are the score and neighborhood features. We demonstrate the fitness of the proposed network representation as well as its added value in fraud detection strategies. Not only previous fraudulent behavior of a policyholder, but also a claim's entire social network is of great importance to detect new fraudulent claims. As such, our method is 'white box', we can interpret the network derived features, as opposed to the network representation learning approaches which are currently prominent in the literature. Resent research on using features derived from variations of the personalized PageRank algorithm, has demonstrated its potential to capture complex non-linear effects and correlations between interconnected objects, such as borrowers in credit scoring (Óskarsdóttir et al., 2019; Bravo and Óskarsdóttir, 2020).

The results of the methodology presented in this paper facilitate continued research for social network based fraud detection in insurance. Firstly, in our methodology we omitted the time-weighting of fraud influence and of edges, an approach presented in Van Vlasselaer et al. (2016). This would be a very interesting addition and potentially may help to better understand fraud recency in the insurance context. Secondly, in this paper, we used SMOTE, a state-of-the-art method that is common in the fraud literature, to rebalance the dataset. More in depth research on the difference between such methods is possible, which we would like to explore in our future work. Thirdly, there is selection bias with respect to the fraudulent claims, since only suspicious claims undergo a fraud investigation. Because of this potential bias, we decided to consider three types of labels: unknown, fraud and non-fraud, and deployed our one-versus-all approach with the two datasets, $\mathcal{D}^{\texttt{known}}$ and $\mathcal{D}^{\texttt{fraud}}$. However, the availability of labelled as well as unlabelled claims is an ideal premise for a semi-supervised machine learning approach. We would like to explore such methods in order to accommodate for the low ratio of labelled claims which is an inherent problem we faced in our research. Such an approach was developed amongs others for rejected loan applications by Li et al. (2020) and could potentially help in finding the rare fraudulent claims in our case. Next, graph based anomaly detection methods would be an interesting approach to find the rare fraudulent claims in the network. Such methods are becoming increasingly more prominent in the network science research community. Finally, with the fast growing development of network representation learning, there is an opportunity to learn low dimensional vector representations for the nodes in a network. These methods use deep learning approaches to learn structural properties of the nodes in a network and the resulting representations can subsequently be used with a supervised machine learning technique to classify the nodes, like we did here for fraud. In our future work we will especially focus on frameworks such as GraphSAGE that leverage feature rich nodes, such as the claims in our case (Hamilton et al., 2017).

Acknowledgement

This work was supported by the Ageas Research chair at KU Leuven and KU Leuven's research council [project COMPACT C24/15/001]. This support is gratefully acknowledged.

This research has been financed in part by the NeEDS research project, an EC H2020 MSCA RISE project with Grant agreement No. 822214.

References

Jing Ai, Patrick L Brockett, and Linda L Golden. Assessing consumer fraud risk in insurance claims: An unsupervised learning technique using discrete and continuous predictor variables. *North American Actuarial Journal*, 13(4):438–458, 2009.

Jing Ai, Patrick L Brockett, Linda L Golden, and Montserrat Guillén. A robust unsupervised method for fraud rate estimation. *Journal of Risk and Insurance*, 80(1):121–143, 2013.

Martin A Andresen and Marcus Felson. The impact of co-offending. *The British Journal of Criminology*, 50(1):66–81, 2009.

Bart Baesens, Veronique Van Vlasselaer, and Wouter Verbeke. *Fraud analytics using descriptive, predictive, and social network techniques: a guide to data science for fraud detection.* John Wiley & Sons, 2015.

Christopher M Bishop. *Pattern recognition and machine learning.* Springer, 2006.

João Botelho and Cláudia Antunes. Combining social network analysis with semi-supervised clustering: a case study on fraud detection. In *Proceedings of Mining Data Semantics (MDS'2011) in Conjunction with SIGKDD*, 2011.

Cristián Bravo and María Óskarsdóttir. Evolution of credit risk using a personalized pagerank algorithm for multilayer networks. *arXiv preprint arXiv:2005.12418*, 2020.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

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.

Steven B Caudill, Mercedes Ayuso, and Montserrat Guillén. Fraud detection using a multinomial logit model with missing information. *Journal of Risk and Insurance*, 72(4):539–550, 2005.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

EIOPA. Big data analytics in motor and health insurance: a thematic review, 2019. URL https://eiopa.europa.eu/Publications/EIOPA_BigDataAnalytics_ThematicReview_April2019.pdf.

FBI. Investigating insurance fraud, 2012. URL https://www.fbi.gov/news/stories/investigating-insurance-fraud.

Ali Ghorbani and Sara Farzai. Fraud detection in automobile insurance using a data mining based approach. *Int. J. Mechatron. Electr. Comput. Technol*, 8(27):3764–3771, 2018.

Linda L Golden, Patrick L Brockett, Montserrat Guillén, and Danae Manika. apridit unsupervised classification with asymmetric valuation of variable discriminatory worth. *Multivariate behavioral research*, 55(5):685–703, 2020.

Donatien Hainaut. A self-organizing predictive map for non-life insurance. *European Actuarial Journal*, 9(1):173–207, 2019.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.

Xiangnan He, Ming Gao, Min-Yen Kan, and Dingxian Wang. Birank: Towards ranking on bipartite graphs. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):57–71, 2017.

Soheil Jamshidi and Mahmoud Reza Hashemi. An efficient data enrichment scheme for fraud detection using social network analysis. In *Sixth International Symposium on Telecommunications (IST)*, pages 1082–1087. IEEE, 2012.

David Jensen. Prospective assessment of AI technologies for fraud detection: A case study. In *AAAI Workshop on AI Approaches to Fraud Detection and Risk Management*, pages 34–38, 1997.

Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

Ilker Kose, Mehmet Gokturk, and Kemal Kilic. An interactive machine-learning-based electronic fraud and abuse detection system in healthcare insurance. *Applied Soft Computing*, 36:283–299, 2015.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *Proceedings of the 19th international conference on World Wide Web*, pages 591–600. AcM, 2010.

Aurélie Lemmens and Christophe Croux. Bagging and boosting classification trees to predict churn. *Journal of Marketing Research*, 43(2):276–286, 2006.

Yaqi Li, Chun Yan, Wei Liu, and Maozhen Li. A principle component analysis-based random forest with the potential nearest neighbor method for automobile insurance fraud identification. *Applied Soft Computing*, 70:1000–1009, 2018.

Zhiyong Li, Xinyi Hu, Ke Li, Fanyin Zhou, and Feng Shen. Inferring the outcomes of rejected loans: an application of semisupervised clustering. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 2020.

Gabriele Lohmann, Daniel S Margulies, Annette Horstmann, Burkhard Pleger, Joeran Lepsien, Dirk Goldhahn, Haiko Schloegl, Michael Stumvoll, Arno Villringer, and Robert Turner. Eigenvector centrality mapping for analyzing connectivity patterns in fMRI data of the human brain. *PloS one*, 5(4): e10232, 2010.

Qing Lu and Lise Getoor. Link-based classification. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 496–503, 2003.

Wei Min, Zhengyang Tang, Min Zhu, Yuxi Dai, Yan Wei, and Ruinan Zhang. Behavior language processing with graph based feature generation for fraud detection in online lending. In *WSDM workshop on Misinformation and Misbehavior Mining on the Web*. MIS2, 2018.

Mark Newman. *Networks: an introduction*. Oxford University Press, 2010.

Ke Nian, Haofan Zhang, Aditya Tayal, Thomas Coleman, and Yuying Li. Auto insurance fraud detection using unsupervised spectral ranking for anomaly. *The Journal of Finance and Data Science*, 2(1):58–75, 2016.

María Óskarsdóttir, Cristián Bravo, Carlos Sarraute, Jan Vanthienen, and Bart Baesens. The value of big data for credit scoring: Enhancing financial inclusion using mobile phone data and social network analytics. *Applied Soft Computing*, 74:26–39, 2019.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

Juyong Park and Albert-László Barabási. Distribution of node characteristics in complex networks. *Proceedings of the National Academy of Sciences*, 104(46):17916–17920, 2007.

Albert J Reiss Jr. Co-offending and criminal careers. *Crime and justice*, 10:117–170, 1988.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

Russell G Smith, Penny Jorna, Josh Sweeney, and Georgina Fuller. Counting the costs of crime in Australia: a 2011 estimate. *Research and public policy series*, 129, 2014. URL `https://aic.gov.au/publications/rpp/rpp129`.

Eugen Stripling, Bart Baesens, Barak Chizi, and Seppe vanden Broucke. Isolation-based conditional anomaly detection on mixed-attribute data to uncover workers' compensation fraud. *Decision Support Systems*, 111:13–26, 2018.

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.

Sharmila Subudhi and Suvasini Panigrahi. Use of optimized fuzzy c-means clustering and supervised classifiers for automobile insurance fraud detection. *Journal of King Saud University-Computer and Information Sciences*, 2017.

G Ganesh Sundarkumar and Vadlamani Ravi. A novel hybrid undersampling method for mining unbalanced datasets in banking and insurance. *Engineering Applications of Artificial Intelligence*, 37: 368–377, 2015.

Mohammad A Tayebi and Uwe Glässer. Social network analysis in predictive policing. In *Social Network Analysis in Predictive Policing*, pages 7–14. Springer, 2016.

Fredy Troncoso and Richard Weber. A novel approach to detect associations in criminal networks. *Decision Support Systems*, 128:113159, 2020.

M. Vere Van Koppen, Christianne J De Poot, Edward R Kleemans, and Paul Nieuwbeerta. Criminal trajectories in organized crime. *The British Journal of Criminology*, 50(1):102–123, 2009.

Véronique Van Vlasselaer, Tina Eliassi-Rad, Leman Akoglu, Monique Snoeck, and Bart Baesens. Gotcha! network-based fraud detection for social security fraud. *Management Science*, 63(9):3090–3110, 2016.

Wouter Verbeke, David Martens, and Bart Baesens. Social network analysis for customer churn prediction. *Applied Soft Computing*, 14:431–446, 2014.

Verbond van Verzekeraars. Dutch insurance industry in figures 2016, 2016. URL https://www.verzekeraars.nl/publicaties/verzekerd-van-cijfers.

Verbond van Verzekeraars. Cbv factsheet, 2019. URL https://www.verzekeraars.nl/media/6611/factsheet-verzekeringsfraude-oktober-2019.pdf.

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.

Stijn Viaene, Mercedes Ayuso, Montserrat Guillen, Dirk Van Gheel, and Guido Dedene. Strategies for detecting fraudulent claims in the automobile insurance industry. *European Journal of Operational Research*, 176(1):565–583, 2007.

Yibo Wang and Wei Xu. Leveraging deep learning with lda-based text analytics to detect automobile insurance fraud. *Decision Support Systems*, 105:87–95, 2018.

Danielle E Warren and Maurice E Schweitzer. When lying does not pay: How experts detect insurance fraud. *Journal of Business Ethics*, 150(3):711–726, 2018.