Guilt-by-Constellation: Fraud Detection by Suspicious Clique Memberships

Véronique Van Vlasselaer
KU Leuven
Veronique.VanVlasselaer@kuleuven.be

Leman Akoglu
Stony Brook University
leman@cs.stonybrook.edu

Tina Eliassi-Rad*
Rutgers University
tina@eliassi.org

Monique Snoeck
KU Leuven
Monique.Snoeck@kuleuven.be

Bart Baesens
KU Leuven
Bart.Baesens@kuleuven.be

Abstract

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

1. Introduction

Fraud detection is a research domain that highly relies on the automated process of finding anomalous behavior in massive amounts of data. Automated algorithms are able to guide fraud experts by identifying potential high-risk observations. Distinguishing legitimate observations from fraudulent ones, however, is a nontrivial task, mainly due to the extremely imbalanced character of fraud. Indeed, less than 1% of the observations in fraud data sets are fraudulent. Traditionally, fraud is detected by applying simple if-then rules – e.g. if the amount due is not paid within a certain period, then the entity or transaction is marked as fraudulent. While such decision rules can identify straightforward types of fraud, subtle and adversarially planned effects of fraud are not captured. Machine learning offers a plethora of powerful techniques that are able to learn from historical data and discover useful patterns from the data. Extracting meaningful features that capture abnormal behavior, is a crucial step in efficiently detecting fraud through machine learning. An example of such features are ones that capture the interrelations between fraudsters (such as their group behaviors). We explore the usefulness of such features in this work.

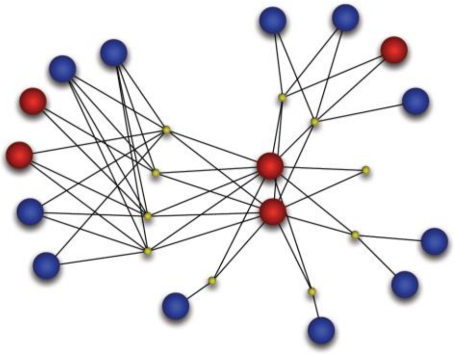


Figure 1. Subgraph of companies (large nodes) connected to their resources (small nodes). Fraudulent companies are red-colored, currently-legitimate companies are in blue. Companies form cliques (i.e., fully connected subgraphs) based on their use of the same set of resources.

So far, the fraud detection literature has mainly focused on analyzing *guilt-by-association* [1], i.e. how relationships with fraudsters affect the probability that a person of interest will commit fraud. For example, suppose there are two fraudsters *B* and *C* who are both connected to person *A* (let's say, by a friendship relation), then guilt-by-association analyzes each relationship to those neighbors separately. However, this approach does not take into account the relationships between the neighbors. In this work, we introduce *guilt-by-constellation* in which we derive suspicious cliques of nodes, and characterize each node in terms of its clique membership. A clique is a fully connected subgraph of the network where each node is

* This work was funded in part by LLNL under Contract DE-AC52- 07NA27344, by NSF CNS-1314603, by DTRA HDTRA1-10-1-0120, and by DAPRA under SMISC Program Agreement No. W911NF-12-C-0028.

connected to every other node in the subgraph. For example, suppose now that persons *B* and *C* also know each other and, as a consequence, persons *A*, *B* and *C* form a clique of friends. Guilt-by-constellation investigates whether this will have a stronger effect on the fraud probability of person *A*.

In this paper, we address social security fraud and show a successful example of how clique-based features are an important element in inferring future fraud. We define fraud as those companies that intentionally go bankrupt in order to avoid paying tax contributions: their debt to the government will be unrecoverable. We observe that after a certain time period a new company is founded which uses almost the same resources as the previous company, like machinery, equipment, employees, address, buyers, suppliers, etc. (see Figure 1). As opposed to many graph-related works, we exploit *bipartite* graphs, connecting two object types, i.e. companies and resources. We find that when a new company enters the market and inherits (a part of) the same set of resources previously associated with a fraudulent company (or companies), its fraud risk increases.

We introduce clique-based features which are shown to outperform previous approaches to this problem. In particular, we define both complete- and partial-cliques (i.e., companies share *all* or *part of* their resources with each other) and investigate: (1) Does the probability of perpetrating future fraud increase when fraudulent companies are closely connected to each other, i.e. they form a dense group where they all share the same (set of) resources? (2) If a new company enters such a group, what would we say about its probability to commit fraud?

Based on these analyses and observations, we define relational and clique-based features using a graph representation. Relational features aggregate the characteristics of close neighbors by treating each of them as a separate individual regardless of their links to other neighbors (i.e., guilt-by-association). Clique-based features, on the other hand, also take into account the connectivity within the neighborhood (i.e., guilt-by-constellation). In addition to networked features (which capture *peer pressure*), we incorporate intrinsic features in our models. These intrinsic features are able to detect new types of fraud (e.g., ones that are not imitated). Remark that our models are dynamically updated, by extracting time-dependent individual and clique membership scores for each company and by re-estimating the corresponding models. We contribute by proposing a novel approach to detect fraud by:

- Defining cliques in a bipartite graph where one type of nodes (i.e., companies) are connected to another type of nodes (i.e., resources) (see Section 4.3).
- Using a time-dependent *individual exposure score* (Section 4.2) of every node to label cliques in the network and infer a *suspiciousness score* (Section 4.3) for that clique.
- Featurizing new instances based on the properties of the cliques they belong to, and integrating the extracted features with intrinsic and relational features (see Section 4.4).

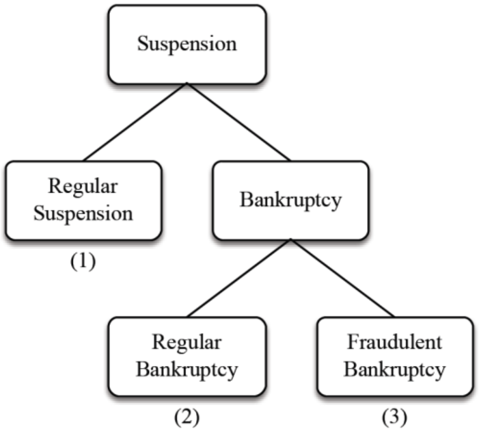


Figure 2. A company can end its economical lifecycle in three different ways: (1) regular suspension, (2) regular bankruptcy or (3) fraudulent bankruptcy.

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

2. Social Security Fraud Detection

Our proposed approach will be applied to social security fraud detection. While this is only one application to integrate clique memberships in detection algorithms, we believe that a similar approach is promising on comparable application domains, like credit card fraud detection, insurance fraud, opinion fraud, and so on.

In this paper, we study social security data acquired from the Belgian Social Security Institution. In general, the Belgian Social Security Institution keeps track of companies registered in Belgium and a set of resources. Those resources are associated with companies. Due to confidentiality issues, we cannot elaborate further on the exact type of resources but the reader may understand those resources in terms of machinery, equipment, address, employees, buyers, suppliers, etc. As resources do not uniquely belong to one company, they can be shared or transferred among several companies. In addition, a company can make use of multiple resources.

Companies need to contribute employer and employee taxes to the government. We say that if a company *intentionally* goes bankrupt so as not to pay its tax contributions, the company is fraudulent. Fraudulent companies often belong to a *web of fraud*, i.e. the resources of fraudulent companies are (partly) transferred to other companies which will commit fraud on their turn. E.g., fraudulent companies *A*, *B* and *C* operated at address *p* and used suppliers *a* and *b*. All those resources are now transferred to active company *D*. Company *D* is likely to commit fraud in the future.

So far, all fraudulent companies detected by subject matter experts are identified *ex post*. This means that the companies are already bankrupt with unrecoverable debts to the government. In this paper, the goal is to identify those companies *ex ante*, such that experts can closely follow up companies with a high risk of not paying off their taxes, and curtailing the growth of existing and new webs of fraud.

Remark that all fraudulent companies are bankrupt, but that not all bankrupt companies are fraudulent. This is depicted in Figure 2. While suspension is seen as a normal way of stopping a company's activities (i.e., all debts redeemed), bankruptcy indicates that the company did not succeed to pay back all its creditors. Distinguishing between regular and fraudulent bankruptcies, is subtle and hard to establish. Experts expect that some regular bankruptcies are in fact undetected fraudulent bankruptcies.

We will use the network of companies and resources to judge the fraud probability or risk of a set of active companies. Resources move in bulk from one fraudulent company to another, leaving a trail of fraud. Using the company-resource network, we propose to capture *clique* behavior of the resources to cluster together companies. We will extract both a fraud and bankruptcy score for each clique: resource involvement in many *fraudulent* companies increases the fraud risk of future companies that use the same set of resources. Resource involvement in many *bankruptcies* might increase the fraud risk as well, as this may uncover an undiscovered group of fraudulent companies. We expect that currently-legitimate members of cliques that are highly associated with fraud or bankruptcy, have a higher probability of committing fraud in the near future. In this work, we try to answer questions like (1) does *guilt-by-constellation* detect future fraud more efficiently (2) what effect does a suspicious (i.e., fraudulent) clique have on currently legitimate companies that are part of that clique? (3) what effect does a clique characterized only by (apparent) regular bankruptcies have on currently-legitimate companies that are part of that clique?

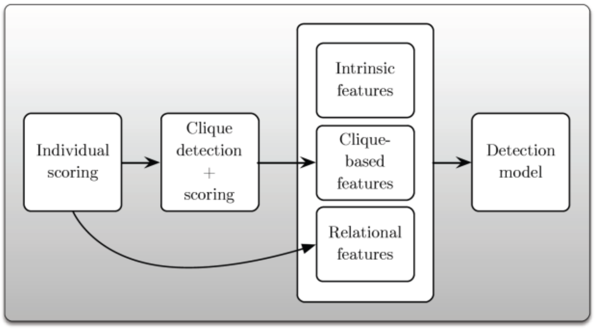


Figure 3. Flow-chart of detection process.

3. Related work

While previous literature acknowledges the importance of network analysis in fraud detection, most research focuses on the so-called *guilt-by-association*. Many works aggregate relational information in features such as degree, proportion, count, etc. [2,3,4] or apply inference procedures to spread the fraudulent influence throughout the whole network [5,6,7,8]. The aforementioned techniques neglect the density among the neighborhood of the node of interest, i.e. the extent to which the surrounding nodes are connected to each other as well. This is known as clusters, communities or cliques in the network [9]. Cortes et al. [10] formulated the idea to compute the *community of interest* (COI) centered around each node in the network and compare the overlap between COI's. A significant overlap with a fraudulent COI might indicate that the COI is also fraudulent. Fast et al. [3] developed a fraud detection approach for the National Association of Securities Dealers (NASD) which uses *tribes* or clusters of representatives. The authors focused on suspicious pairs of representatives that do not comply with a normal pattern in the industry. Akoglu et al. [6] proposed *FraudEagle*, a novel approach to spot fraudulent reviewers and reviews for opinion fraud detection. The authors used a co-clustering [11] technique to group together the top high-risk users for visualization purposes.

To the best of our knowledge, this paper is the first to define cliques in a bipartite graph and featurize currently-legitimate instances (here companies) based on their memberships in cliques.

4. Proposed method

4.1 Task description

The primary goal of this paper is to predict which currently active companies form a threat to perpetrate fraud in the future by estimating a detection model that consist of a combination of intrinsic, relational and clique-based features. Specifically, our approach consists of four steps, as illustrated in Figure 3:

1. *Individual scoring*: The influence of few known fraudulent (bankrupt) companies is spread through the network, deriving a time-dependent exposure score for every node. That is, each company and resource receive a score based on the presence of fraudulent (bankrupt) influence in their neighborhood.
2. *Clique detection and scoring*: Resources and companies that are frequently associated with each other are clustered in a clique. We aggregate the individual exposure scores of the involved companies and the resources to derive a suspiciousness score for each clique.
3. *Feature extraction*: We calculate the value of the features for each currently active company based on its clique memberships (31), and combine them with intrinsic (18) and relational (2) features. In total, we have 51 company characteristics.
4. *Model estimation*: We integrate all extracted features and try to predict which companies are highly sensitive to commit fraud in the future.

Next we introduce our definitions and notations. A network which includes two node types, is called a *bipartite graph*.

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

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

Let $\mathcal{V}_1$ be the set of company nodes, and $\mathcal{V}_2$ the set of all resource nodes, then a company is uniquely connected to resources and vice versa. At a certain timestamp *t*, all companies are labeled according to their fraud involvement $\ell_f(v_1) \in \{\text{legitimate}, \text{fraud}\}$ and their bankruptcy involvement $\ell_b(v_1) \in \{\text{active}, \text{bankrupt}\}$. Those labels are used to infer an individual fraud and bankruptcy exposure score for every company and resource.

4.2. Individual exposure score

Given a network of companies and resources, how can we use the label of few companies to infer a score of the other nodes. More specifically, the goal here is to derive an *exposure score* for each node, i.e. for each company and resource. As we are interested in both the fraudulence and the bankruptcy probability in social security fraud, we will infer both a *fraud* and *bankruptcy* exposure score. Those scores reveal whether the neighborhood of the company or the resource is frequently involved in fraud or bankruptcy respectively.

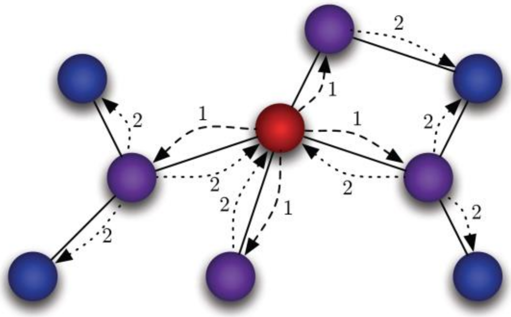


Figure 4. Illustration of the GOTCHA! propagation algorithm. The red node propagates its fraudulent influence to its neighbors (step 1). The neighbors absorb the influence and propagate on their turn their fraudulent influence to their neighbors (step 1 + 2). The iterations are repeated several times until convergence.

Given a matrix representation of a bipartite graph $\boldsymbol{M}$ of size $r \times c$ with r the number of resources and c the number of companies, we want to diffuse or *propagate* the effect of a limited number of known fraudulent (bankrupt) companies through the network (see Figure 4). In earlier work [7], we proposed the *GOTCHA!* propagation algorithm to derive an *exposure score* for each node in a bipartite graph. In short, the *GOTCHA!* propagation algorithm inherits concepts from the Personalized PageRank as proposed by Page and Brin [12] to compute the ranking of *n* web pages, and:

$$(\vec{r}_i) = c \cdot \boldsymbol{A} \cdot (\vec{r}_i) + (1 - c) \cdot \vec{v}_i \qquad (1)$$

with $\vec{r}_i$ a vector containing the PageRank scores, depending with a probability c on the scores of the neighboring nodes as denoted by the adjacency matrix $\boldsymbol{A}$ of size $n \times n$, and a probability $1 - c$ on a personalized vector $\vec{v}_i$. We call c the restart probability and $\vec{v}_i$ the restart vector.

In order to face the challenges imposed by the fraud detection domain, we will change Eq. 1 such that the *GOTCHA!* propagation algorithm complies with the following requirements concerning fraud: (1) *Bipartite graphs:* Resources are an important indicator of fraud. The network consists of both companies and resources. (2) *Focus on fraudulent influence:* Rather than diffusing any influence through the network, the algorithm should emphasize fraud and only allow to propagate fraudulent influence through the network.

(3) *Dynamical character*: First, relationships are time-dependent. The edges in the network should be temporally weighted, giving a high weight to recent relationships. Second, fraud is time-dependent. Companies that were recently caught should diffuse a higher fraudulent influence in the network. (4) *Degree-independent propagation*: fraud affects each resource equally whether the company has many resources or not. In order to avoid that low-degree companies spread more fraudulent influence to their resources than high-degree companies, the fraudulent influence that companies spread through the network is proportional to the number of associated resources. Next, we will explain how each requirement is implemented.

As our propagation algorithm scores both companies and resources, we transform the bipartite adjacency matrix $\boldsymbol{M}$ to a unipartite adjacency matrix according to requirement (1):

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{0}_{cxc} & \boldsymbol{M}' \\ \boldsymbol{M} & \boldsymbol{0}_{rxr} \end{pmatrix} \qquad (2a)$$

with $\boldsymbol{Q}$ a symmetrical matrix.

In order to propagate only fraudulent influence through the network, we will change the starting vector $\vec{v}$, such that requirement (2) is fulfilled:

$$\begin{cases} v_j = 0, & \text{if entry } j \text{ is a resource or a legitimate company} \\ v_j = 1, & \text{if entry } j \text{ is a fraudulent company} \end{cases}$$

Requirement (3) takes into account the dynamic structure of networks. A network is a representation of a real-world concept where relationships between nodes are constantly added and removed. In a social security context, a new relation pops up in the network if a company starts using a specific resource. A current relation is removed if the company stops using a resource. As we believe that past relations are equally important to present global shifting patterns of the resources, instead of removing relationships, we opt to decay the influence between the two nodes dependent on the recency of the relation:

$$w_{i,j} = m_{i,j} \cdot e^{-\alpha h}$$

with $m_{i,j} \in \boldsymbol{M}$ the relation between resource i and company j (0, if no relation), $w_{i,j} \in \boldsymbol{W}$ the weighted relation between resource i and company j (0, if no relation), α the decay value, and h the number of weeks passed since the relationship was still active. The value of α defines the pace at which the relationship degrades. Accordingly, Eq. 2a equals:

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{0}_{cxc} & \boldsymbol{W}' \\ \boldsymbol{W} & \boldsymbol{0}_{rxr} \end{pmatrix} \qquad (2b)$$

The same reasoning holds for valuing the importance of the fraudulent influence of companies. The starting vector $\vec{v}$ is exponentially decayed, and defined as:

$$v_j = v_j \cdot e^{-\alpha h}$$

with α the decay value, and h the number of weeks passed since fraud was detected at company j.

Finally, in order to avoid emphasizing low-degree fraudulent companies, the starting vector is adapted such that the neighborhood of each company absorbs an equal amount of fraudulent influence according to requirement (4):

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

with $\vec{z}$ the degree-adapted starting vector which is the element-wise product of $\vec{v}$ the time-weighted starting vector and $\vec{d}$ the degree vector.

Eq. 1 implies a matrix inversion. This is often not feasible in practice, due to the large size of input graphs[1]. As such, we will use the power-iteration method which iterates the following equation until convergence:

$$\vec{r}_{k+1} = c \cdot \boldsymbol{Q_{norm}} \cdot \vec{r}_k + (1-c) \cdot \vec{z}_{norm} \qquad (3)$$

with $\vec{r}_k$ the exposure scores after k iterations, c the restart probability, $\boldsymbol{Q_{norm}}$ the row-normalized connectivity matrix and $\vec{z}_{norm}$ the normalized degree-adapted restart vector such that $\vec{z}_{norm}$ sums to 1. Vector $\vec{r}_0$ is a random vector with values between [0,1]. Convergence is reached after a predefined number of iterations or until the change in exposure scores is insignificant.

This step results in an exposure score for every node, i.e. companies and resources. For each node, we will compute both the fraud and bankruptcy exposure score. Those scores will be used in the next steps to characterize each identified clique in terms of its fraud and bankruptcy accumulation.

4.3 Clique detection and scoring

Given present and past relationships of the companies and their resources, can we build cliques of

[1] Complexity of the best performing algorithm for matrix inversion equals O($n^{2.373}$) [13].

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

Clique Detection According to [14], a community is defined as a tightly connected group of nodes or subgraph in the network. A *clique* is the strongest definition and requires that all objects of a subgraph are connected to each other. In bipartite graphs, we define a clique as a subgraph in which each type-one object is connected to each type-two object. This means that we induce a subgraph from the network in which all companies are connected to all resources and vice versa. Note that our approach only tends to find company cliques, and uses resources to associate the companies.

We apply a bottom-up approach to find all cliques in the network, which we describe in detail as follows. First, we start by enumerating all pairs of companies that share at least two resources. Since we are inclined to analyze strong relationships between companies, we require that each clique contains at least two companies and two resources. For each two companies in the data set, we list all of their shared resources. Next, we merge any two pairs of companies that share the same resources (or an intersection of the resources). If two pairs can be merged together in a complete-clique based on an *exact match* of all resources, the original pairs are deleted from the set of cliques. If the resources of two pairs of companies *partially overlap*, the two pairs are merged if both groups share at least two resources together. Those cliques are considered partial-cliques. The original pairs are kept in the set of cliques. This step is repeated until no newly created cliques can be merged together, i.e. until there is no exact or partial overlap between the new cliques in the set. We illustrate examples of the types of cliques this procedure creates in Figure 5.

Typically, a clique either consists of many companies that share only few resources or few companies with many resources. Since we do not delete partially overlapping groups, some cliques might be contained in other cliques (see the bottom figure in Figure 5). Thus, we are able to obtain insights in the intensity of the relationships between companies. For example, the bottom figure illustrates that company *B* is part of a “large” partial-clique α that connects it to companies *A* and *C*. This clique is formed based on two shared resources (c-d). Yet, company *B* is also contained in clique β based on four shared resources (a-d). As such, company *A* will have a larger influence on company *B* than company *C*, as company *A* is stronger connected to company *B* than to company *C*.

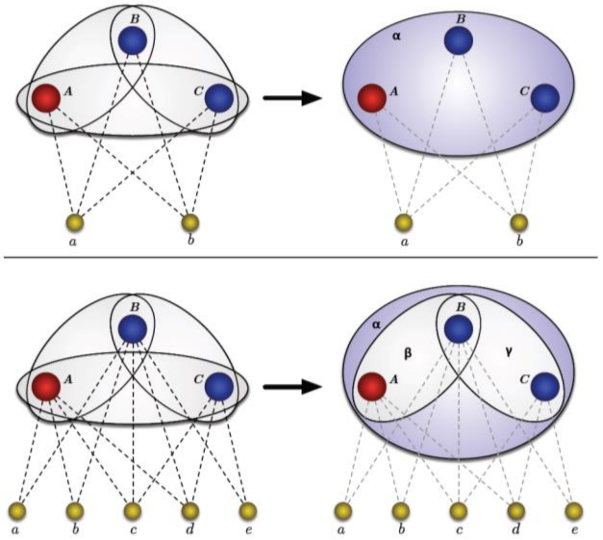


Figure 5. Clique detection process. Companies A, B and C share the same (set of) resources. The top figure illustrates the merging process for an exact match between pairs of companies. The original pairs are deleted from the final set. Only clique α is in the remaining set of cliques. The bottom figure represents a partial overlap between pairs of companies. Here, the original pairs β and γ, together with a new clique α are all added to the new set of cliques.

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

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

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

4.4. Feature extraction

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

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

Relational The fraud and bankruptcy exposure score embody the proximity of fraudulent or bankrupt influence in the company's neighborhood. A high *fraud score* indicates that many companies in the surrounding environment were already caught by perpetrating fraudulent activities. The *bankruptcy score* reveals the extent to which neighboring companies are bankrupt. These scores are computed in Section 4.2.

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

1. COUNT: Number of cliques to which the company belongs.
2. AVERAGE: The characteristics as defined in Section 4.3 are averaged over all the cliques the company belongs to. For example, the average fraud count reflects the average number of fraudulent companies that reside in a clique.
3. MAXIMUM: The danger of considering the average values of all the associated cliques is that the effect of one highly suspicious clique can be filtered out by many innocent cliques. Therefore, we include the maximum value for each of the identified clique characteristics. For example, the maximum fraud count captures the maximum number of fraudulent companies that are within one clique.

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

All the aforementioned features are combined and passed to the detection process.

4.5. Detection model

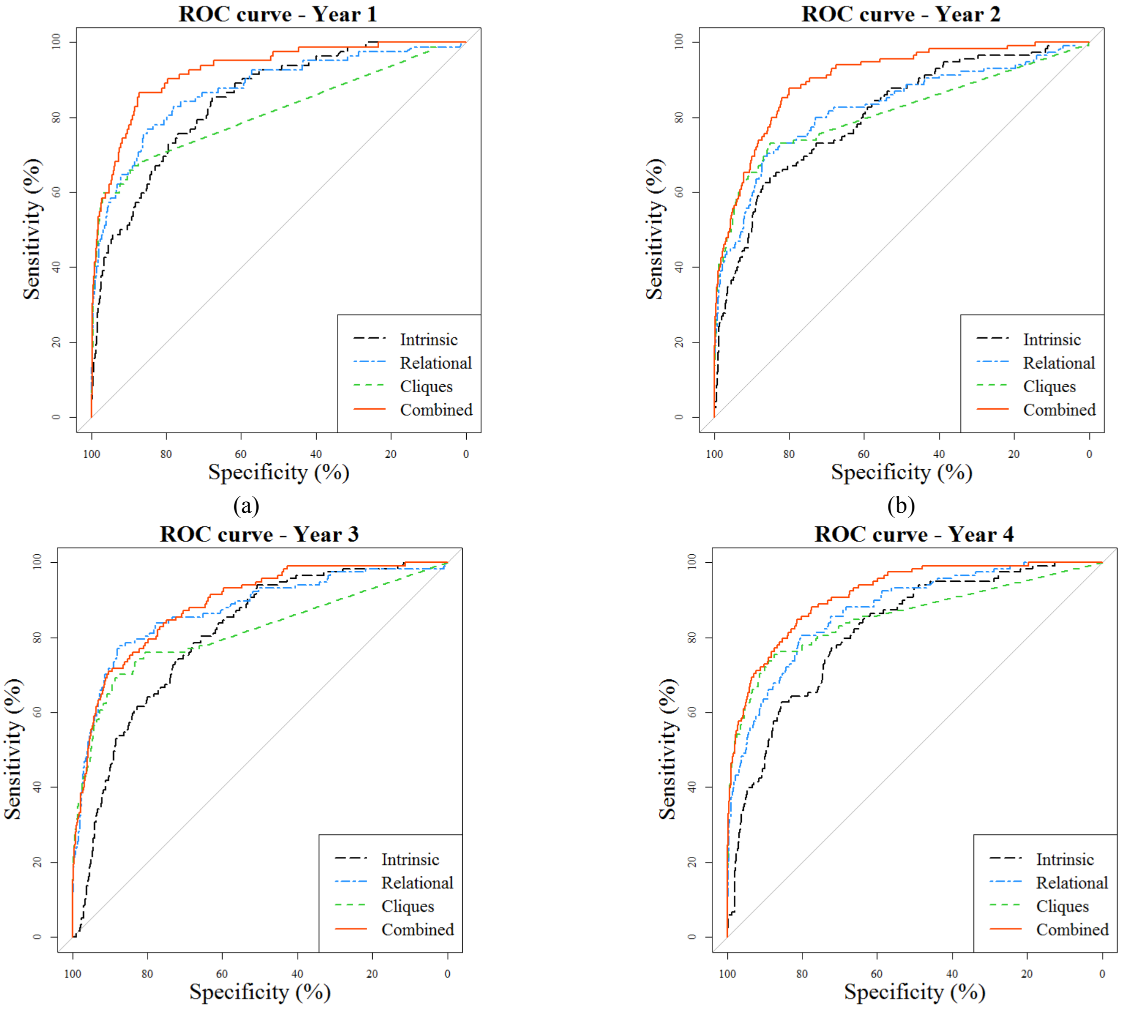
The data set provided by the social security institution is a dynamic data set which includes past and present company characteristics and past and present relationships between companies and their resources. In order to validate the detection power over time, we choose to (re-)estimate the model for four timestamps and three time windows. More concretely, for every timestamp, we extract the features of all active companies according to Section 4.4, and infer a model to predict which companies will perpetrate fraud within a certain time window. We define three time windows: short, medium or long term. Based on experts' knowledge, we arbitrarily set the time windows to 6, 12 and 24 months. While short-term models are able to capture new fraud mechanisms, long-term models have more evidence to learn from. The models are re-estimated on a yearly basis, i.e. for timestamps year 1 – 4. Due to confidentiality issues, we do not specify the exact timestamp.

Table 1. Extremely skewed data distribution for the social security institution

	Short term	Medium term	Long term
Year 1	0.03%	0.06%	0.11%
Year 2	0.04%	0.08%	0.16%
Year 3	0.04%	0.09%	0.16%
Year 4	0.07%	0.11%	0.16%

Fraud data sets commonly represent an extremely skewed distribution. This means that often less than 1% of the observations are fraudulent. In particular, Table 1 represents the data distribution for the social security data set. Less than 0.05%, 0.10% and 0.20% of the companies will be fraudulent on short, medium and long term respectively. In order to rebalance the data set, we apply SMOTE (Synthetic Minority Oversampling Technique) as proposed by [15] on the training set. Based on empirical evidence of [15], the oversampling and undersampling percentage are set to 400% and 200% respectively.

Previous literature acknowledges that the featurization of network-related characteristics of an object might create a multitude of input features which can deteriorate the results, and suggests the use of ensemble methods to carefully select the most



(a)

(b)

(c)

(d)

Figure 6. ROC curves for the different timestamps of our analysis. Notice that the combined model which includes all of the intrinsic, relational, and clique-based features outperforms the models using any one of those features alone.

important features [16]. Our models are estimated using Random Forests [17]. This ensemble method infers a set of decision trees by randomly selecting features. A voting process decides the label of each instance.

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

5. Empirical Evaluation

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

5.1. Data set

For each timestamp, approximately 220,000 active companies and 5 million resources are registered with the social security institution. In order to derive exposure scores (Section 4.2) and suspicious clique memberships (Section 4.3), we include past fraudulent and bankrupt companies to the bipartite graph. Only regularly suspended companies were excluded from the analysis, as those companies do not contribute to the fraud detection process. In Year 4, the bipartite graph consists of around 400,000 companies and 5.6 million resources.

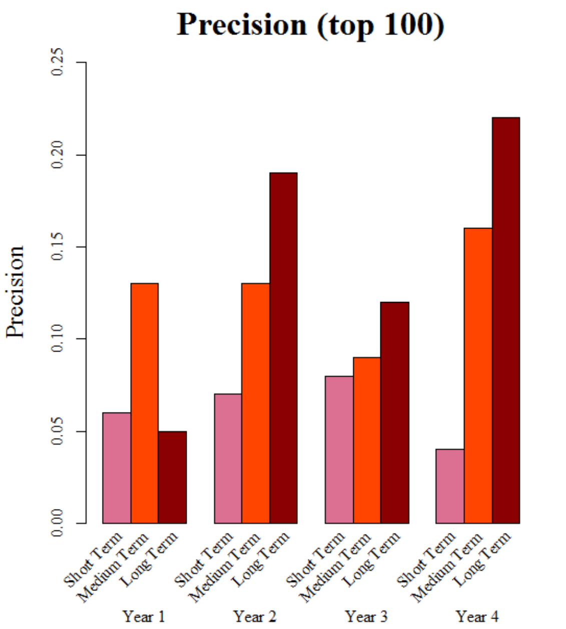


Figure 8. Precision of the top 100 most high-risk companies. Generally speaking, long-term models perform better than short- and medium-term models.

5.2. Performance over time

Figure 6 depicts the ROC curves of the various timestamps of our analysis. All ROC curves present the model performance for a long-term time window. The ROC curves indicate that the combined models generate better results. In addition, a pairwise t-test confirms that the combined approach performs significantly better than the other models for all timestamps and time windows ($\alpha < 0.05$). Especially the steep slope of the curve clearly indicates that the combined model is particularly good in classifying companies as fraudulent that have a high cut-off value (i.e., companies with a high fraud probability according to the model are in reality often sensitive to fraud). This high true positive rate is particularly important because experts have limited resources available to investigate high-risk companies, and are able to inspect only a few companies in each timestamp.

Note from the figures that the clique-based and the combined model have a similar increase for high cut-off values. This might indicate that the clique-based features are mainly responsible for the high prediction power of the combined model when only a limited number of companies is selected. The relational model also follows a steep increase, but especially lifts up the curve of the combined model in the middle, when the clique-based model performs poorly.

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

5.3. Precision

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

In Figure 7, we illustrate the precision for the combined model for each timestamp and each time window. Except for Year 1 where we have limited networked data, long-term models have a higher precision. More than 20 out of 100 companies that are classified as fraudulent in Year 4, do indeed perpetrate fraud in the future. This means that high-risk companies already radiate suspicious behavior and characteristics even before they effectively perform fraudulent activities.

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

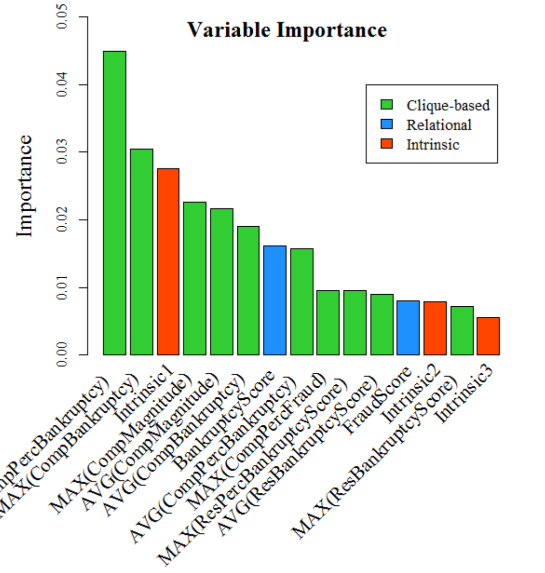


Figure 8. Variable importance of the top 15 features in the combined model.

5.4. Variable importance

We would like to assess which variables contribute to the high prediction power of the estimated detection models.

Figure 8 illustrates that the top 15 most important variables are mainly clique-based features, although one of the intrinsic features also has a high explanatory power. Note that the most important clique-based variables are bankruptcy, rather than the fraud related variables. We can conclude that an environment which is highly sensitive to bankruptcy might actually be a construction with hidden fraud.

6. Conclusions

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

7. References

[1] Koutra, D., Ke, T. Y., Kang, U., Chau, D. H. P., Pao, H. K. K., and Faloutsos, C., “Unifying guilt-by-association approaches: Theorems and fast algorithms”, PKDD, Springer, 2011, pp. 245-260.

[2] Neville, J., Şimşek, Ö., Jensen, D., Komoroske, J., Palmer, K., and Goldberg, H., “Using relational knowledge discovery to prevent securities fraud”, KDD, ACM, 2005, pp. 449-458.

[3] Fast, A., Friedland, L., Maier, M., Taylor, B., Jensen, D., Goldberg, H. G., and Komoroske, J., “Relational data pre-processing techniques for improved securities fraud detection”, KDD, ACM, 2007, pp. 941-949.

[4] Van Vlasselaer, V., Meskens, J., Van Dromme, D., and Baesens, B., “Using social network knowledge for detecting spider constructions in social security fraud”, ASONAM, ACM, 2013, pp. 813-820.

[5] Pandit, S., Chau, D. H., Wang, S., and Faloutsos, C., “Netprobe: a fast and scalable system for fraud detection in online auction networks”, WWW, ACM, 2007, pp. 201-210.

[6] Akoglu, L., Chandy, R., and Faloutsos, C., “Opinion fraud detection in online reviews by network effects”, ICWSM, 2013.

[7] Van Vlasselaer, V., Eliassi-Rad, T., Akoglu, L., Snoeck, M., Baesens, B., “Gotcha! Network-based fraud detection for social security fraud”, under peer review, 2014.

[8] Akoglu, L., McGlohon, M., and Faloutsos, C., “Oddball: Spotting anomalies in weighted graphs”, PAKDD, Springer, 2010, pp. 410-421.

[9] Newman, M., Networks: an introduction. Oxford University Press, 2010.

[10] Cortes, C., Pregibon, D., and Volinsky, C., “Communities of interest”, Lecture Notes in Computer Science, 2001, pp. 105-114.

[11] Chakrabarti, D., Papadimitriou, S., Modha, D. S., and Faloutsos, C., “Fully automatic cross-associations”, KDD, ACM, 2004, pp. 79-88.

[12] Page, L., Brin, S., Motwani, R., and Winograd, T., “The PageRank citation ranking: Bringing order to the web”, Tech. rep., Stanford Digital Library Technologies Project, 1999.

[13] Raz, R., “On the complexity of matrix product.”, STOC, ACM, 2002, pp. 144-151.

[14] Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D. U., “Complex networks: Structure and dynamics”, Physics reports, 2006, pp. 175-308.

[15] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P., “SMOTE: synthetic minority over-sampling technique”, Journal of Artificial Intelligence Research, 2011.

[16] Gallagher, B., Tong, H., Eliassi-Rad, T., & Faloutsos, C., “Using ghost edges for classification in sparsely labeled networks”, KDD, ACM, 2008, pp. 256-264.

[17] Breiman, L., “Random forests. Machine learning”, Machine learning, 2001, pp. 5-32.