An expert system for detecting automobile insurance fraud using social network analysis

Lovro Šubelj*, Štefan Furlan, Marko Bajec

Faculty of Computer and Information Science, University of Ljubljana, Tržaška 25, SI-1001 Ljubljana, Slovenia

Abstract

The article proposes an expert system for detection, and subsequent investigation, of groups of collaborating automobile insurance fraudsters. The system is described and examined in great detail, several technical difficulties in detecting fraud are also considered, for it to be applicable in practice. Opposed to many other approaches, the system uses networks for representation of data. Networks are the most natural representation of such a relational domain, allowing formulation and analysis of complex relations between entities. Fraudulent entities are found by employing a novel assessment algorithm, *Iterative Assessment Algorithm* (*IAA*), also presented in the article. Besides intrinsic attributes of entities, the algorithm explores also the relations between entities. The prototype was evaluated and rigorously analyzed on real world data. Results show that automobile insurance fraud can be efficiently detected with the proposed system and that appropriate data representation is vital.

Key words: Fraud detection, Automobile insurance, Social network analysis, Link analysis, Assessment propagation

1. Introduction

Fraud is encountered in a variety of domains. It comes in all different shapes and sizes, from traditional fraud, e.g. (simple) tax cheating, to more sophisticated, where entire *groups* of individuals are collaborating in order to commit fraud. Such groups can be found in the automobile insurance domain.

Here fraudsters stage traffic accidents and issue fake insurance claims to gain (unjustified) funds from their general or vehicle insurance. There are also cases where an accident has never occurred, and the vehicles have only been placed onto the road. Still, the majority of such fraud is not planned (*opportunistic fraud*) – an individual only seizes the opportunity arising from the accident and issues exaggerated insurance claims or claims for past damages.

Staged accidents have several common characteristics. They occur in late hours and non-urban areas in order to reduce the probability of witnesses. Drivers are usually younger males, there are many passengers in the vehicles, but never children or elders. The police is always called to the scene to make the subsequent acquisition of means easier. It is also not uncommon that all of the participants have multiple (serious) injuries, when there is almost no damage on the vehicles. Many other suspicious characteristics exist, not mentioned here.

The insurance companies place the most interest in organized groups of fraudsters consisting of drivers, chiropractors, garage mechanics, lawyers, police officers, insurance workers and others. Such groups represent the majority of revenue leakage. Most of the analyses agree that approximately 20% of all insurance claims are in some way fraudulent (various resources). But most of these claims go unnoticed, as fraud investigation is usually done by hand by the domain expert or investigator and is only rarely computer supported. Inappropriate representation of data is also common, making the detection of groups of fraudsters extremely difficult. An expert system approach is thus needed.

Jensen (1997) has observed several technical difficulties in detecting fraud (various domains). Most hold for (automobile) insurance fraud as well. Firstly, only a small portion of accidents or participants is fraudulent (*skewed class distribution*) making them extremely difficult to detect. Next, there is a severe lack of *labeled* data sets as labeling is expensive and time consuming. Besides, due to sensitivity of the domain, there is even a lack of unlabeled data sets. Any approach for detecting such fraud should thus be founded on moderate resources (data sets) in order to be applicable in practice. Fraudsters are very innovative and new types of fraud emerge constantly. Hence, the approach must also be highly adaptable, detecting new types of fraud as soon as they are noticed. Lastly, it holds that fully autonomous detection of automobile insurance fraud is not possible in practice. Final assessment of potential fraud can only be made by the domain expert or investigator, who also determines further actions in resolving it. The approach should also support this investigation process.

Due to everything mentioned above, the set of approaches for detecting such fraud is extremely limited. We propose a novel expert system approach for detection and subsequent investigation of automobile insurance fraud. The system is focused on detection of groups of collaborating fraudsters, and their connecting accidents (non-opportunistic fraud), and not some isolated fraudulent entities. The latter should be done in-

*Corresponding author. Tel.: +386 1 4768 186.

Email addresses: lovro.subelj@fri.uni-lj.si (Lovro Šubelj), stefan.furlan@fri.uni-lj.si (Štefan Furlan), marko.bajec@fri.uni-lj.si (Marko Bajec) dependently for each particular entity, while in our system, the entities are assessed in a way that considers also the relations between them. This is done with appropriate representation of the domain – networks.

Networks are the most natural representation of any relational domain, allowing formulation of complex relations between entities. They also present the main advantage of our system against other approaches that use a standard *flat data* form. As collaborating fraudsters are usually related to each other in various ways, detection of groups of fraudsters is only possible with appropriate representation of data. Networks also provide clear visualization of the assessment, crucial for the subsequent investigation process.

The system assesses the entities using a novel *Iterative Assessment Algorithm* (*IAA* algorithm), presented in this article. No learning from initial labeled data set is done, the system rather allows simple incorporation of the domain knowledge. This makes it applicable in practice and allows detection of new types of fraud as soon as they are encountered. The system can be used with poor data sets, which is often the case in practice. To simulate realistic conditions, the discussion in the article and evaluation with the prototype system relies only on the data and entities found in the police record of the accident (main entities are participant, vehicle, collision[1], police officer).

The article makes an in depth description, evaluation and analysis of the proposed system. We pursue the hypothesis that automobile insurance fraud can be detected with such a system and that proper data representation is vital. Main contributions of our work are: (1) a novel expert system approach for the detection of automobile insurance fraud with networks; (2) a benchmarking study, as no expert system approach for detection of groups of automobile insurance fraudsters has yet been reported (to our knowledge); (3) an algorithm for assessment of entities in a relational domain, demanding no labeled data set (*IAA* algorithm); and (4) a framework for detection of groups of fraudsters with networks (applicable in other relational domains).

The rest of the article is organized as follows. In section 2 we discuss related work and emphasize weaknesses of other proposed approaches. Section 3 presents formal grounds of (social) networks. Next, in section 4, we introduce the proposed expert system for detecting automobile insurance fraud. The prototype system was evaluated and rigorously analyzed on real world data, description of the data set and obtained results are given in section 5. Discussion of the results is conducted in section 6, followed by the conclusion in section 7.

2. Related work

Our work places in the wide field of fraud detection. Fraud appears in many domains including telecommunications, banking, medicine, e-commerce, general and automobile insurance. Thus a number of expert system approaches for preventing, detecting and investigating fraud have been developed in the past. Researches have proposed using some standard methods of data mining and machine learning, *neural networks*, *fuzzy logic*, *genetic algorithms*, *support vector machines*, *(logistic) regression*, *consolidated (classification) trees*, approaches over *red-flags* or *profiles*, various statistical methods and other methods and approaches (Artis et al., 2002; Brockett et al., 2002; Bolton & Hand, 2002; Estevez et al., 2006; Furlan & Bajec, 2008; Ghosh & Schwartzbard, 1999; Hu et al., 2007; Kirkos et al., 2007; Perez et al., 2005; Rupnik et al., 2007; Quah & Sriganesh, 2008; Sanchez et al., 2009; Viaene et al., 2002, 2005; Weisberg & Derrig, 1998; Yang & Hwang, 2006). Analyses show that in practice none is significantly better than others (Bolton & Hand, 2002; Viaene et al., 2005). Furthermore, they mainly have three weaknesses. They (1) use inappropriate or inexpressive representation of data; (2) demand a labeled (initial) data set; and (3) are only suitable for larger, richer data sets. It turns out that these are generally a problem when dealing with fraud detection (Jensen, 1997; Phua et al., 2005).

In the narrower sense, our work comes near the approaches from the field of network analysis, that combine intrinsic attributes of entities with their relational attributes. Noble & Cook (2003) proposed detecting anomalies in networks with various types of vertices, but they focus on detecting suspicious structures in the network, not vertices (i.e. entities). Besides that, the approach is more appropriate for larger networks. Researchers also proposed detecting anomalies using measures of centrality (Freeman, 1977, 1979), random walks (Sun et al., 2005) and other (Holder & Cook, 2003; Maxion & Tan, 2000), but these approaches mainly rely only on the relational attributes of entities.

Many researchers have investigated the problem of classification in the relational context, following the hypothesis that classification of an entity can be improved by also considering its related entities (inference). Thus many approaches formulating *inference*, *spread* or *propagation* on networks have been developed in various fields of research (Brin & Page, 1998; Domingos & Richardson, 2001; Kleinberg, 1999; Kschischang & Frey, 1998; Lu & Getoor, 2003b; Minka, 2001; Neville & Jensen, 2000). Most of them are based on one of the three most popular (approximate) inference algorithms: *Relaxation Labeling (RL)* (Hummel & Zucker, 1983) from the computer vision community, *Loopy Belief Propagation (LBP)* on loopy (Bayesian) *graphical models* (Kschischang & Frey, 1998) and *Iterative Classification Algorithm (ICA)* from the data mining community (Neville & Jensen, 2000). For the analyses and comparison see (Kempe et al., 2003; Sen & Getoor, 2007).

Researchers have reported good results with these algorithms (Brin & Page, 1998; Kschischang & Frey, 1998; Lu & Getoor, 2003b; Neville & Jensen, 2000), however they mainly address the problem of learning from an (initial) labeled data set (*supervised learning*), or a partially labeled (*semi-supervised learning*) (Lu & Getoor, 2003a), therefore the approaches are generally inappropriate for fraud detection. The algorithm we introduce here, *IAA* algorithm, is almost identical to the *ICA* algorithm, however it was developed with different intentions in

[1] Throughout the article the term collision is used instead of (traffic) accident. The word accident implies there is no one to blame, which contradicts with the article.

mind – to assess the entities when no labeled data set is at hand (and not for improving classification with inference). Furthermore, *IAA* does not address the problem of *classification*, but *ranking*. Thus, in this way, it is actually a simplification of *RL* algorithm, or even Google's *PageRank* (Brin & Page, 1998), still it is not founded on the probability theory like the latter.

We conclude that due to the weaknesses mentioned, most of the proposed approaches are inappropriate for detection of (automobile) insurance fraud. Our approach differs, as it does not demand a labeled data set and is also appropriate for smaller data sets. It represents data with networks, which are one of the most natural representation and allow complex analysis without simplification of data. It should be pointed out that networks, despite their strong foundations and expressive power, have not yet been used for detecting (automobile) insurance fraud (at least according to our knowledge).

3. (Social) networks

Networks are based upon mathematical objects called *graphs*. Informally speaking, graph consists of a collection of points, called *vertices*, and links between these points, called *edges* (Fig. 1). Let V_G, E_G be a set of vertices, edges for some graph G respectively. We define G as $G = (V_G, E_G)$ where

$$V_G = \{v_1, v_2 \dots v_n\}, \quad (1)$$

$$E_G \subseteq \{\{v_i, v_j\}|\ v_i, v_j \in V_G \land i \neq j\}. \quad (2)$$

Note that edges are sets of vertices, hence they are not directed (*undirected graph*). In the case of *directed graphs* equation (2) rewrites to

$$E_G \subseteq \{(v_i, v_j)|\ v_i, v_j \in V_G \land i \neq j\}, \quad (3)$$

where edges are ordered pairs of vertices – (v_i, v_j) is an edge from v_i to v_j. The definition can be further generalized by allowing multiple edges between two vertices and loops (edges that connect vertices with themselves). Such graphs are called *multigraphs*. Examples of some simple (multi)graphs can be seen in Fig. 1.

In practical applications we usually strive to store some extra information along with the vertices and edges. Formally, we can define two labeling functions

$$l_{V_G}: \quad V_G \to \Sigma_{V_G}, \quad (4)$$

$$l_{E_G}: \quad E_G \to \Sigma_{E_G}, \quad (5)$$

where $\Sigma_{V_G}, \Sigma_{E_G}$ are (finite) alphabets of all possible vertex, edge labels respectively. *Labeled graph* can be seen in Fig. 1 (b).

We proceed by introducing some terms used later on. Let G be some undirected multigraph or an *underlying graph* of some directed multigraph – underlying graph consists of same vertices and edges as the original directed (multi)graph, only that all of its edges are set to be undirected. G naturally partitions into a set of *(connected) components* denoted $C(G)$. E.g. all three graphs in Fig. 1 have one connected component, when graphs in Fig. 2 consist of several connected components. From here on, we assume that G consists of a single connected component.

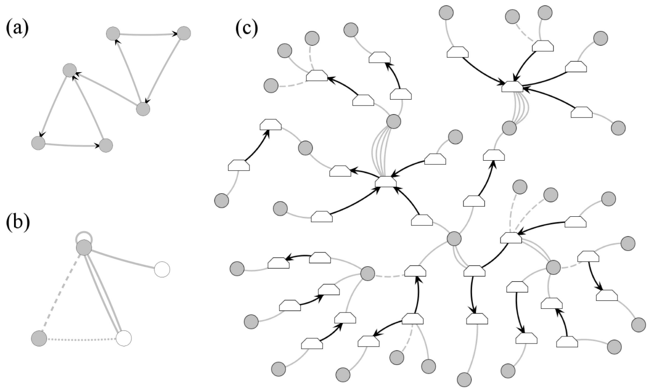

Fig. 1: (a) simple graph with directed edges; (b) undirected multigraph with labeled vertices and edges (labels are represented graphically); (c) network representing collisions where round vertices correspond to participants and cornered vertices correspond to vehicles. Collisions are represented with directed edges between vehicles.

Let v_i be some vertex in graph G, $v_i \in V_G$. *Degree* of the vertex v_i, denoted $d(v_i)$, is the number of edges incident to it. Formally,

$$d(v_i) = |\{e|\ e \in E_G \land v_i \in e\}|. \quad (6)$$

Let v_j be some other vertex in graph G, $v_j \in V_G$, and let $p(v_i, v_j)$ be a *path* between v_i and v_j. A path is a sequence of vertices on a way that leads from one vertex to another (including v_i and v_j). There can be many paths between two vertices. A *geodesic* $g(v_i, v_j)$ is a path that has the minimum size – consists of the least number of vertices. Again, there can also be many geodesics between two vertices.

We can now define the *distance* between two vertices, i.e. v_i and v_j, as

$$d(v_i, v_j) = |g(v_i, v_j)| - 1. \quad (7)$$

Distance between v_i and v_j is the number of edges visited when going from v_i to v_j (or vice versa). The *diameter* of some graph G, denoted $d(G)$, is a measure for the "width" of the graph. Formally, it is defined as the maximum distance between any two vertices in the graph,

$$d(G) = \max\{d(v_i, v_j)|\ v_i, v_j \in V_G\}. \quad (8)$$

All graphs can be divided into two classes. First are *cyclic* graphs, having a path $p(v_i, v_i)$ that contains at least two other vertices (besides v_i) and has no repeated vertices. Such path is called a *cycle*. Graphs in Fig. 1 (a) and (b) are both cyclic. Second class of graphs consists of *acyclic* graphs, more commonly known as *trees*. These are graphs that contain no cycle (see Fig. 1 (c)). Note that a simple undirected graph is a tree if and only if $|E_G| = |V_G| - 1$.

Finally, we introduce the *vertex cover* of a graph G. Let S be a subset of vertices, $S \subseteq V_G$, with a property that each edge in E_G has at least one of its incident vertices in S (covered by S). Such S is called a vertex cover. It can be shown, that finding a minimum vertex cover is *NP-hard* in general.

Graphs have been studied and investigated for almost 300 years thus a strong theory has been developed until today. There are also numerous practical problems and applications where graphs have shown their usefulness (e.g. Brin & Page, 1998) – they are the most natural representation of many domains and are indispensable whenever we are interested in relations between entities or in patterns in these relations. We emphasize this only to show that networks have strong mathematical, and also practical, foundation – *networks*[2] are usually seen as labeled, or *weighted*, multigraphs with both directed and undirected edges (see Fig. 1 (c)). Furthermore, vertices of a network usually represent some entities, and edges represent some relations between them. When vertices correspond to people, or groups of people, such networks are called *social networks*.

Networks often consist of densely connected subsets of vertices called *communities*. Formally, communities are subsets of vertices with many edges between the vertices within some community and only a few edges between the vertices of different communities. Girvan & Newman (2002) suggested identifying communities by recursively removing the edges between them – *between edges*. As it holds that many geodesics run along such edges, where only few geodesics run along edges within communities, between edges can be removed by using *edge betweenness* (Girvan & Newman, 2002). It is defined as

$$\begin{aligned} Bet(e_i) &= |\{g(v_i, v_j)|\; v_i, v_j \in V_G \wedge \\ &\quad \wedge\; g(v_i, v_j) \text{ goes along } e_i\}|, \end{aligned} \tag{9}$$

where $e_i \in E_G$. The edge betweenness $Bet(e_i)$ is thus the number of all geodesics that run along edge e_i.

For more details on (social) networks see e.g. (Newman, 2003, 2008).

4. Expert system for detecting automobile insurance fraud

As mentioned above, the proposed expert system uses (primarily constructed) networks of collisions to assign suspicion score to each entity. These scores are used for the detection of groups of fraudsters and their corresponding collisions. The *framework* of the system is structured into four *modules* (Fig. 2).

In the first module, different types of networks are constructed from the given data set. When necessary, the networks are also simplified – divided into natural communities that appear inside them. The latter is done without any loss of generality.

Networks from the first module naturally partition into several connected components. In the second module we investigate these components and output the suspicious, focusing mainly on their structural properties such as diameter, cycles, etc. Other components are discarded at the end of this module.

Not all entities in some suspicious component are necessarily suspicious. In the third module components are thus further analyzed in order to detect key entities inside them. They are found by employing *Iterative Assessment Algorithm (IAA)*, presented in this article. The algorithm assigns a suspicion score to each entity, which can be used for subsequent assessment and analysis – to identify suspicious groups of entities and their connecting collisions. In general, suspicious groups are subsets of suspicious components.

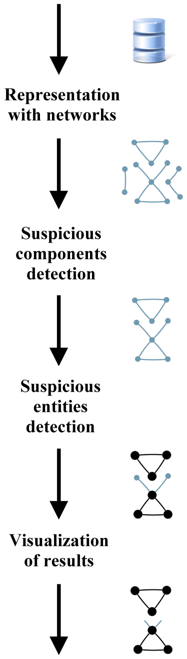
Representation with networks
Suspicious components detection
Suspicious entities detection
Visualization of results

Fig. 2: Framework of the proposed expert system for detecting (automobile insurance) fraud.

Note that detection of suspicious entities is done in two *stages* (second and third module). In the first stage, or the second module, we focus only on detecting suspicious components and in the second stage, third module, we also locate the suspicious entities within them. Hence the detection in the first, second stage is done at the level of components, entities respectively. The reason for this *hierarchical investigation* is that early stages simplify assessment in the later stages, possibly without any loss for detection (for further implications see section 6).

It holds that fully autonomous detection of automobile insurance fraud is not possible in practice. The obtained results should always be investigated by the domain expert or investigator, who determines further actions for resolving potential fraud. The purpose of the last, fourth, module of the system is thus to appropriately assess and visualize the obtained results, allowing the domain expert or investigator to conduct subsequent analysis.

First three modules of the system are presented in sections 4.1, 4.2, 4.3 respectively, when the last module is only briefly discussed in section 4.4.

4.1. Representation with networks

Every entity's attribute is either *intrinsic* or *relational*. Intrinsic attributes are those, that are independent of the entity's sur-

[2] Throughout the article the terms graph and network are used as synonyms.

rounding (e.g. person's age), while the relational attributes represent, or are dependent on, relations between entities (e.g. relation between two colliding drivers). Relational attributes can be naturally represented with the edges of a network. Thus we get networks, where vertices correspond to entities and edges correspond to relations between them. Numerous different networks can be constructed, depending on which entities we use and how we connect them to each other.

The purpose of this first module of the system is to construct different types of networks, used later on. It is not immediately clear how to construct networks, that describe the domain in the best possible way and are most appropriate for our intentions. This problem arises as networks, despite their high expressive power, are destined to represent relations between only two entities (i.e. *binary relations*). As collisions are actually relations between multiple entities, some sort of projection of the data set must be made (for other suggestions see section 7).

Collisions can thus be represented with various types of networks, not all equally suitable for fraud detection. In our opinion, there are some guidelines that should be considered when constructing networks from any relational domain data (guidelines are given approximately in the order of their importance):

1. *Intention:* networks should be constructed so that they are most appropriate for our intentions (e.g. fraud detection)
2. *Domain:* networks should be constructed in a way that describes the domain as it is (e.g. connected vertices should represent some entities, also directly connected in the data set)
3. *Expressiveness:* expressive power of the constructed networks should be as high as possible
4. *Structure:* structure of the networks should not be used for describing some specific domain characteristics (e.g. there should be no cycles in the networks when there are no actual cycles in the data set). Structural properties of networks are a strong tool that can be used in the subsequent (investigation) process, but only when these properties were not artificially incorporated into the network during the construction process
5. *Simplicity:* networks should be kept as simple and sparse as possible (e.g. not all entities need to be represented by its own vertices). The hypothesis here is that simple networks would also allow simpler subsequent analysis and clearer final visualization (principle of *Occam's razor*[3])
6. *Uniqueness:* every network should uniquely describe the data set being represented (i.e. there should be a *bijection* between different data sets and corresponding networks)

Frequently all guidelines can not be met and some trade-off have to be made.

In general there are $\binom{3}{1} + \binom{3}{2} + (\binom{3}{2} + \binom{3}{3}) = 10$ possible ways how to connect three entities (i.e. collision, participant and vehicle), depending on which entities we represent with their own vertices. 7 of these represent participants with vertices and in 4 cases all entities are represented by their own vertices. For the reason of simplicity, we focus on the remaining 3 cases. In the following we introduce four different types of such networks, as an example and for later use. All can be seen in Fig. 3.

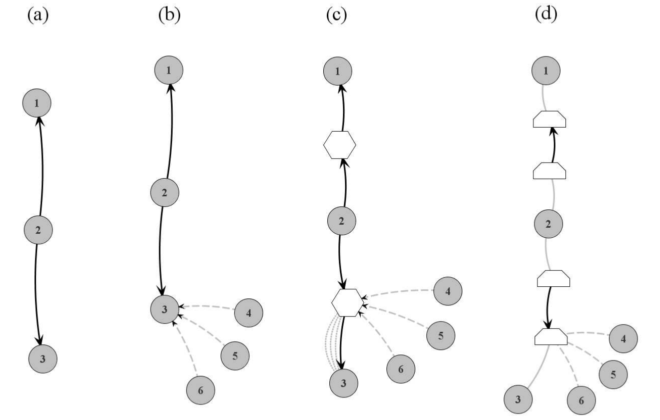

Fig. 3: Four types of networks representing same two collisions – (a) *drivers network*, (b) *participants network*, (c) *COPTA network* and (d) *vehicles network*. Rounded vertices correspond to participants, hexagons correspond to collisions and irregular cornered vertices correspond to vehicles. Solid directed edges represent involvement in some collision, solid undirected edges represent drivers (only for the vehicles network) and dashed edges represent passengers. Guilt in the collision is formulated with edge's direction.

The simplest way is to only connect the drivers who were involved in the same collision – *drivers networks*. Guilt in the collision is formulated with edge's direction. Note that drivers networks severely lack expressive power (guideline 3). We can therefore add the passengers and get *participants networks*, where passengers are connected with the corresponding drivers. Such networks are already much richer, but they have one major weakness – passengers "group" on the driver, i.e. it is generally not clear which passengers were involved in the same collision and not even how many passengers were involved in some particular collision (guidelines 3, 6).

This weakness is partially eliminated by *COnnect Passengers Through Accidents networks* (*COPTA networks*). We add special vertices representing collisions and all participants in some collision are now connected through these vertices. Passengers no longer group on the drivers but on the collisions, thus the problem is partially eliminated. We also add special edges between the drivers and the collisions, to indicate the number of passengers in the vehicle. This type of networks could be adequate for many practical applications, but it should be mentioned that the distance between two colliding drivers is now twice as large as before – the drivers are those that were directly related in the collision (guideline 2, 5).

Last type of networks are *vehicles networks* where special vertices are added to represent vehicles. Collisions are now represented by edges between vehicles, and driver and passengers are connected through them. Such networks provide good visualization of the collisions and also incorporate another entity, but they have many weaknesses as well. Two colliding drivers

[3] The principle states that the explanation of any phenomenon should make as few assumptions as possible, eliminating those making no difference in the assessment – entities should not be multiplied beyond necessity.

are very far apart and (included) vehicles are not actually of our interest (guideline 5). Such networks also seem to suggest that the vehicles are the ones, responsible for the collision (guideline 2). Vehicles networks are also much larger than the previous.

A better way to incorporate vehicles into networks is simply to connect collisions, in which the same vehicle was involved. Similar holds for other entities like police officers, chiropractors, lawyers, etc. Using special vertices for these entities would only unnecessarily enlarge the networks and consequently make subsequent detection harder (guidelines 1, 5). It is also true, that these entities usually aren't available in practice (sensitivity of the domain).

Summary of the analysis of different types of networks is given in table 1.

Guidelines and networks

	drivers	*particip.*	*COPTA*	*vehicles*	
Intention	+	++			5
		+	++		
Domain			−	−	4
Expressive.	−−	−		+	4
Structure					4
Simplicity	+		−	−−	3
Uniqueness		−	−	−	2
Total	0	**4**	−9	−8	
	−5	−1	**1**	−8	

Table 1: Comparison of different types of networks due to the proposed guidelines. Scores assigned to the guidelines are a choice made by the authors. Analysis for *Intention* (guideline 1), and total score, is given separately for second, third module respectively.

There is of course no need to use the same type of networks in every stage of the detection process (guideline 1). In the prototype system we thus use participants networks in the second module (section 4.2), as they provide enough information for initial suspicious components detection, and *COPTA* networks in the third module (section 4.3), whose adequacy will be clearer later. Other types of networks are used only for visualization purposes. Network scores, given in table 1, confirm this choice.

After the construction of networks is done, the resulting connected components can be quite large (depending on the type of networks used). As it is expected that groups of fraudsters are relatively small, the components should in this case be simplified. We suggest using edge betweenness (Girvan & Newman, 2002) to detect communities in the network (i.e. supersets of groups of fraudsters) by recursively removing the edges until the resulting components are small enough. As using edge betweenness assures that we would be removing only the edges between the communities, and not the edges within communities, simplification is done without any loss for generality.

4.2. *Suspicious components detection*

The networks from the first module consist of several connected components. Each component describes a group of related entities (i.e. participants, due to the type of networks used), where some of these groups contain fraudulent entities. Within this module of the system we want to detect such groups (i.e. *fraudulent components*) and discard all others, in order to simplify the subsequent detection process in the third module. Not all entities in some fraudulent component are necessarily fraudulent. The purpose of the third module is to identify only those that are.

Analyses, conducted with the help of a domain expert, showed that fraudulent components share several *structural characteristics*. Such components are usually much larger than other, non-fraudulent components, and are also denser. The underlying collisions often happened in suspicious circumstances, and the ratio between the number of collisions and the number of different drivers is usually close to 1 (for reference, the ratio for completely independent collisions is 2). There are vertices with extremely high degree and *centrality*. Components have a small diameter, (short) cycles appear and the size of the minimum vertex cover is also very small (all due to the size of the component). There are also other characteristics, all implying that entities, represented by such components, are unusually closely related to each other. Example of a fraudulent component with many of the mentioned characteristics is shown in Fig. 4.

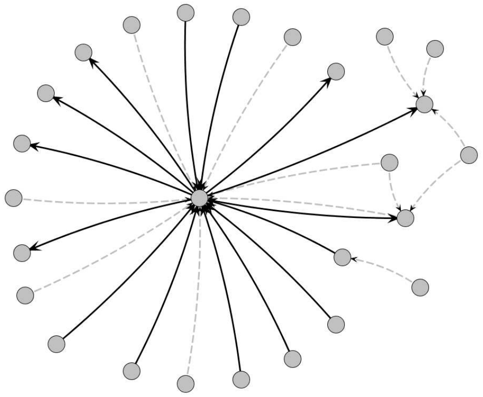

Fig. 4: Example of a component of participants network with many of the suspicious characteristics shared by fraudulent components.

We have thus identified several *indicators* of likelihood that some component is fraudulent (i.e. *suspicious component*). The detection of suspicious components is done by assessing these indicators. Only simple indicators are used (no combinations of indicators).

Formally, we define an ensemble of n indicators as $I = [I_1, I_1 \dots I_n]^T$. Let c be some connected component in network G, $c \in C(G)$, and let $H_i(c)$ be the value for c of the characteristic, measured by indicator I_i. Then

$$I_i(c) = \begin{cases} 1 & c \text{ has suspicious value of } H_i \\ 0 & \text{otherwise} \end{cases}. \qquad (10)$$

For the reason of simplicity, all indicators are defined as *binary attributes*. For the indicators that measure a characteristic that is independent of the structure of the component (e.g. number of vertices, collisions, etc.), simple *thresholds* are defined in order to distinguish suspicious components from others (due to this characteristic). These thresholds are set by the domain expert.

Other characteristics are usually greatly dependent on the number of the vertices and edges in the component. A simple *threshold strategy* thus does not work. Values of such H_i could of course be "normalized" before the assessment (based on the number of vertices and edges), but it is often not clear how. Values could also be assessed using some (supervised) learning algorithm over a labeled data set, but a huge set would be needed, as the assessment should be done for each number of vertices and edges separately (owing to the dependence mentioned). What remains is to construct random networks of (presumably) honest behavior and assess the values of such characteristics using them.

No in-depth analysis of collisions networks has so far been reported, and it is thus not clear how to construct such random networks. General random network *generators* or *models*, e.g. (Barabasi & Albert, 1999; Eppstein & Wang, 2002), mainly give results far away from the collisions networks (visually and by assessing different characteristics). Therefore a sort of *rewiring* algorithm is employed, initially proposed by Ball et al. (1997) and Watts & Strogatz (1998).

The algorithm iteratively rewires edges in some component c, meaning that we randomly choose two edges in E_c, $\{v_i, v_j\}$ and $\{v_k, v_l\}$, and switch one of theirs incident vertices. The resulting edges are e.g. $\{v_i, v_l\}$ and $\{v_k, v_j\}$ (see Fig. 5). The number of vertices and edges does not change during the rewiring process and the values for some H_i can thus be assessed by generating a sufficient number of such random networks (for each component).

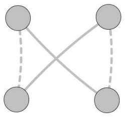

Fig. 5: Example of a rewired network. Dashed edges are rewired, i.e. replaced by solid edges.

The details of the rewiring algorithm are omitted due to space limitations, we only discuss two aspects. First, the number of rewirings should be kept relatively small (e.g. $< |E_c|$), otherwise the constructed networks are completely random with no trace of the one we start with – (probably) not networks representing a set of collisions. We also want to compare components to other random ones, which are similar to them, at least in the aspect of this rewirings. If a component significantly differs even from these similar ones, there is probably some severe anomaly in it.

Second, one can notice that the algorithm never changes the degrees of the vertices. As we wish to assess the degrees as well, the algorithm can be simply adopted to the task in an *ad hoc* fashion. We add an extra vertex v_e and connect all other vertices with it. As this vertex is removed at the end, rewiring one of the newly added edges with some other (old) edge changes the degrees of the vertices. Let $\{v_i, v_e\}$, $\{v_k, v_l\}$ be the edges being rewired and let $\{v_i, v_l\}$, $\{v_k, v_e\}$ be the edges after the rewiring. The (true) degree of vertex v_i, v_k was increased, decreased by one respectively.

To assess the values of indicators we separately construct random components for each component $c \in C(G)$ and indicator $I_i \in I$, and approximate the distributions for characteristics H_i (H_i are seen as random variables). A statistical test is employed to test the *null hypothesis*, if the observed value $H_i(c)$ comes from the distribution for H_i. The test can be *one* or *two-tailed*, based on the nature of characteristic H_i. In the case of one-tailed test, where large values of H_i are suspicious, we get

$$I_i(c) \quad = \quad \begin{cases} 1 & \hat{P}_c(H_i \geq H_i(c)) < t_i \\ 0 & \text{otherwise} \end{cases} , \tag{11}$$

where *probability density function* $P(H_i)$ is approximated with the generated distribution $\hat{P}_c(H_i)$ and t_i is a *critical threshold* or acceptable *Type I error* (e.g. set to 0.05). In the case of two-tailed test the equation (11) rewrites to

$$I_i(c) \quad = \quad \begin{cases} 1 & \begin{array}{c} \hat{P}_c(H_i \geq H_i(c)) < t_i/2 \vee \\ \hat{P}_c(H_i \leq H_i(c)) < t_i/2 \end{array} \\ 0 & \text{otherwise} \end{cases} . \tag{12}$$

Knowing the values for all indicators I_i we can now indicate the suspicious components in $C(G)$. The simplest way to accomplish this is to use a *majority classifier* or *voter*, indicating all the components, for which at least half of the indicators is set to 1, as suspicious. Let $S(G)$ be a set of suspicious components in a network G, $S(G) \subseteq C(G)$, then

$$S(G) \quad = \quad \{c |\, c \in C(G) \wedge \sum_{i=1}^{n} I_i(c) \geq n/2\}. \tag{13}$$

When fraudulent components share most of the characteristics, measured by the indicators, we would clearly indicate them (they would have most, at least half, of the indicators set). Still, the approach is rather naive having three major weaknesses (among others). (1) there is no guarantee that the threshold $n/2$ is the best choice; (2) we do not consider how many components have some particular indicator set; and (3) all indicators are treated as equally important. Normally, we would use some sort of supervised learning technique that eliminates this weaknesses (e.g. regression, neural networks, classification trees, etc.), but again, due to the lack of labeled data and skewed class distribution in the collisions domain, this would only rarely be feasible (the size of $C(G)$ is even much smaller then the size of the actual data set).

To cope with the last two weaknesses mentioned, we suggest using *principal component analysis of RIDITs* (*PRIDIT*) proposed by Brockett & Levine (1977) (see (Brockett, 1981)), which has already been used for detecting fraudulent insurance claim files (Brockett et al., 2002), but not for detecting groups of fraudsters (i.e. fraudulent components). The *RIDIT* analysis was first introduced by Bross (1958).

RIDIT is basically a scoring method that transforms a set of *categorical* attribute values into a set of values from interval $[-1, 1]$, thus they reflect the probability of an occurrence of some particular categorical value. Hence, an *ordinal scale* attribute is transformed into an *interval scale* attribute. In our case, all I_i are simple binary attributes, and the *RIDIT* scores, denoted R_i, are then just

$$R_i(c) \quad = \quad \left\{ \begin{array}{ll} \hat{p}_i^0 & I_i(c) = 1 \\ -\hat{p}_i^1 & I_i(c) = 0 \end{array} \right. , \tag{14}$$

where $c \in C(G)$, $\hat{p}_i^1$ is the *relative frequency* of I_i being equal to 1, computed from the entire data set, and $\hat{p}_i^0 = 1 - \hat{p}_i^1$.

We demonstrate the technique with an example. Let $\hat{p}_i^1$ be equal to 0.95 – almost all of the components have the indicator I_i set. The *RIDIT* score for some component c, with $I_i(c) = 1$, is then just 0.05, as the indicator clearly gives a poor indication of fraudulent components. On the other hand, for some component c, with $I_i(c) = 0$, the *RIDIT* score is -0.95, since the indicator very likely gives a good indication of the non-fraudulent components. Similar intuitive explanation can be made by setting $\hat{p}_i^1$ to 0.05. A full discussion of *RIDIT* scoring is omitted, for more details see (Brockett, 1981; Brockett & Levine, 1977).

Introduction of *RIDIT* scoring diminishes previously mentioned second weakness. To also cope with the third, we make use of the *PRIDIT* technique. The intuition of this technique is that we can weight indicators in some ensemble by assessing the agreement of some particular indicator with the entire ensemble. We make a (probably incorrect) assumption that indicators are independent.

Formally, let W be a vector of *weights* for the ensemble of *RIDIT scorers* R_i for indicators I_i, denoted $W = [w_1, w_2 \dots w_n]^T$, and let R be a matrix with i, j^{th} component equal to $R_j(c)$, where c is an i^{th} component in $C(G)$. Matrix product RW gives the ensemble's score for all the components, i.e. i^{th} component in vector RW is equal to the weighted linear combination of *RIDIT* scores for i^{th} component in $C(G)$. Denote $S = RW$, we can then assess indicators agreement with entire ensemble as (written in matrix form)

$$R^T S / \parallel R^T S \parallel . \tag{15}$$

Equation (15) computes normalized scalar products of columns of R, which corresponds to the returned values of *RIDIT* scorers, and S, which is the overall score of the entire ensemble (for each component in $C(G)$). When the returned values of some scorer are completely orthogonal to the ensemble's scores, the resulting normalized scalar product equals 0, and reaches its maximum, or minimum, when they are perfectly aligned.

Equation (15) thus gives scorers (indicators) agreement with the ensemble and can be used to assign new weights, i.e. $W^1 = R^T S / ||R^T S||$. Greater weights are assigned to the scorers that kind of agree with the general belief of the ensemble. Denote $S^1 = RW^1$, then S^1 is a vector of overall scores using these newly determined weights. There is of course no reason to stop the process here, as we can iteratively get even better weights. We can write

$$W^i \quad = \quad \frac{R^T S^{i-1}}{||R^T S^{i-1}||} = \frac{R^T R W^{i-1}}{||R^T R W^{i-1}||} \tag{16}$$

for $i \geq 1$, which can be used to iteratively compute better and better weights for an ensemble of *RIDIT* scorers R_i, starting with some weights, e.g. $W^0 = [1, 1 \dots 1]$ – the process converges to some fixed point no matter the starting weights (due to some assumptions). It can be shown that the fixed point is actually the *first principal component* of the matrix $R^T R$ denoted W^∞. For more details on *PRIDIT* technique see (Brockett et al., 2002).

We can now score each component in $C(G)$ using the *PRIDIT* technique for indicators I_i and output as suspicious all the components, with a score greater than 0. Thus

$$S(G) \quad = \quad \{c | \, c \in C(G) \wedge R(c) W^\infty \geq 0\}, \tag{17}$$

where $R(c)$ is a row of matrix R, that corresponds to component c. Again there is no guarantee, that the threshold 0 is the best choice. Still, if we know the expected proportion of fraudulent components in the data set (or e.g. expected number of fraudulent collisions), we can first rank the components using *PRIDIT* technique and then output only the appropriate proportion of most highly ranked components.

4.3. Suspicious entities detection

In the third module of the system key entities are detected inside each previously identified suspicious component. We focus on identifying key participants, that can be later used for the identification of other key entities (collisions, vehicles, etc.). Key participants are identified by employing *Iterative Assessment Algorithm (IAA)* that uses intrinsic and relational attributes of the entities. The algorithm assigns a *suspicion score* to each participant, which corresponds to the likelihood of it being fraudulent.

In classical approaches over flat data, entities are assessed using only their intrinsic attributes, thus they are assessed in complete *isolation* to other entities. It has been empirically shown that the *assessment* can be improved by also considering the related entities, more precisely, by considering the assessment of the related entities (Chakrabarti et al., 1998; Domingos & Richardson, 2001; Lu & Getoor, 2003a,b; Neville & Jensen, 2000). The assessment of an entity is *inferred* from the assessments of the related entities and *propagated* onward. Still, incorporating only the intrinsic attributes of the related entities generally doesn't improve, or even deteriorates, the assessment (Chakrabarti et al., 1998; Oh et al., 2000).

The proposed *IAA* algorithm thus assesses the entities by also considering the assessment of their related entities. As these related entities were also assessed using the assessments of their related entities, and so on, the entire network is used in the assessment of some particular entity. This could not be achieved otherwise, as the formulation would surely be too complex. We proceed by introducing *IAA* in a general form.

Let c be some suspicious component in network G, $c \in S(G)$, and let v_i be one of its vertices, $v_i \in V_c$. Furthermore, let $N(v_i)$ be a set of neighbor vertices of v_i (i.e. vertices at distance 1) and $V(v_i) = N(v_i) \cup \{v_i\}$, and let $E(v_i)$ be a set of edges incident to v_i (i.e. $E(v_i) = \{e | \, e \in E_c \wedge v_i \in e\}$). Let also en_i be an entity corresponding to vertex v_i and $N(en_i)$, $V(en_i)$ be a set of entities that corresponds to $N(v_i)$, $V(v_i)$ respectively. We define the suspicion score s, $s(\cdot) \geq 0$, for the entity en_i as

$$\begin{aligned} s(en_i) &= AM(s(N(en_i)), V(en_i), V(v_i), E(v_i)) \\ &= AM(i, c), \end{aligned} \tag{18}$$

where AM is some *assessment model* and $s(N(en_i))$ is a set of suspicion scores for entities in $N(en_i)$. The suspicion of some entity is dependent on the assessment of the related entities (first argument in equation (18)), on the intrinsic attributes of related entities and itself (second argument), and on the relational attributes of the entity (last two arguments). We assume that AM is *linear* in the assessments of the related entities (i.e. $s(N(en_i))$) and that it returns higher values for fraudulent entities.

For some entity en_i, when the suspicion scores of the related entities are known, en_i can be assessed using equation (18). Commonly, none of the suspicion scores are known preliminary (as the data set is unlabeled), and the equation thus cannot be used in a common manner. Still, one can incrementally assess the entities in an *iterative* fashion, similar to e.g. (Brin & Page, 1998; Kleinberg, 1999).

Let $s^0(\cdot)$ be some set of suspicion scores, e.g. $s^0(\cdot) = 1$. We can then assess the entities using scores $s^0(\cdot)$ and equation (18), and get better scores $s^1(\cdot)$. We proceed with this process, iteratively refining the scores until some stopping criteria is reached. Generally, on the k^{th} iteration, entities are assessed using

$$\begin{aligned} s^k(en_i) &= AM(s^{k-1}(N(en_i)), V(en_i), V(v_i), E(v_i)) \\ &= AM(i, k, c). \end{aligned} \tag{19}$$

Note that the choice for $s^0(\cdot)$ is arbitrary – the process converges to some *fixed point* no matter the starting scores (due to some assumptions). Hence, the entities are assessed without preliminary knowing any suspicion score to bootstrap the procedure.

We present the *IAA* algorithm below.

IAA algorithm

```
$s^0(\cdot) = 1$
$k = 1$
WHILE NOT *stopping criteria* DO
    FOR $\forall v_i, en_i$ DO
        $s^k(en_i) = \alpha s^{k-1}(en_i) + (1-\alpha) AM(i, k, c)$
    FOR $\forall v_i, en_i$: $v_i$ *non-bucket* DO
        *normalize* $s^k(en_i)$
    $k = k + 1$
RETURN $s^k(\cdot)$
```

Entities are iteratively assessed using model AM (α is a *smoothing parameter* set to e.g. 0.75). In order for the process to converge, scores corresponding to *non-bucket* vertices are normalized at the end of each iteration. Due to the fact that relations represented by the networks are often not binary, there are usually some vertices only serving as *buckets* that store the suspicion assessed at this iteration to be propagated on the next. *Non-bucket* vertices correspond to entities that are actually being assessed and only these scores should be normalized (for binary relations all the vertices are of this kind). Structure of such *bucket* networks would typically correspond to *bipartite graphs*[4] – bucket vertices would only be connected to non-bucket vertices (and vice versa). In the case of *COPTA* networks, used in this module of the (prototype) system, bucket vertices are those representing collisions.

One would intuitively run the algorithm until some fixed point is reached, i.e. when the scores no longer change. We empirically show that, despite the fact that iterative assessment does indeed increase the performance, such approach actually decreases it. The reason is that the scores *over-fit* the model. We also show, that superior performance can be achieved with a dynamic approach – by running the algorithm for $d(c)$ iterations (diameter of component c). For more see sections 5, 6.

Note that if each subsequent iteration of the algorithm actually increased the performance, one could assess the entities directly. When AM is linear in the assessments of related entities, the model could be written as a set of *linear equations* and solved exactly (analytically).

An arbitrary model can be used with the algorithm. We propose several linear models based on the observation that in many of these bucket networks the following holds: *every entity is well defined with (only) the entities directly connected to it, considering the context observed*. E.g. in the case of *COPTA* networks, every collision is connected to its participants, who are clearly the ones who "define" the collision, and every participant is connected with its collisions, which are the precise aspect of the participant we wish to investigate when dealing with fraud detection. Similar discussion could be made for movie-actor, corporate board-director and other well known collaboration networks. A model using no attributes of the entities is thus simply the sum of suspicion scores of the related entities (we omit the arguments of the model)

$$AM_{raw} = \sum_{\{v_i, v_j\} \in E(v_i)} s(en_j). \tag{20}$$

Our empirical evaluation shows that even such a simple model can achieve satisfactory performance.

To incorporate entities' attributes into the model, we introduce *factors*. These are based on intrinsic or relational attributes of entities. The intuition behind the first is that some intrinsic attributes' values are highly correlated with fraudulent activity. Suspicion scores of corresponding entities should in this case be increased and also propagated on the related entities. Moreover, many of the relational attributes (i.e. labels of the edges) increase the likelihood of fraudulent activity – the propagation of suspicion over such edges should also be increased.

Let l_{E_G} be the edge labeling function and Σ_{E_G} the alphabet of all possible edge labels, i.e. $\Sigma_{E_G} = \{Driver, Passenger \ldots\}$ (for *COPTA* networks). Furthermore, let En be a set of all entities en_i. We define F_{int}, F_{rel} to be the factors, corresponding to intrinsic, relational attributes respectively, as

$$F_{int}: \quad En \rightarrow [0, \infty), \tag{21}$$

$$F_{rel}: \quad \Sigma_{E_G} \rightarrow [0, \infty). \tag{22}$$

[4] In the social science literature bipartite graphs are known as *collaboration networks*.

Improved model incorporating these factors is then

$$AM_{bas} = F_{int}(en_i) \sum_{e=\{v_i,v_j\}\in E(v_i)} F_{rel}(l_{E_G}(e))\, s(en_j). \quad (23)$$

Factors F_{int} are computed from (similar for F_{rel})

$$F_{int}(en_i) = \prod_k F^k_{int}(en_i) \quad (24)$$

where

$$F^k_{int}(en_i) = \begin{cases} 1/(1-f^k_{int}(en_i)) & f^k_{int}(en_i) \geq 0 \\ 1+f^k_{int}(en_i) & \text{otherwise} \end{cases} \quad (25)$$

and

$$f^k_{int}: \quad En \to (-1,1). \quad (26)$$

f^k_{int} are *virtual factors* defined by the domain expert. The transformation with equation (25) is done only to define factors on the interval $(-1,1)$, rather than on $[0,\infty)$. The first is more intuitive as e.g. two "opposite" factors are now f and $-f$, $f \in [0,1)$, opposed to f and $1/f$, $f > 0$, before.

Some virtual factor f^k_{int} can be an arbitrary function defined due to a single attribute of some entity, or due to several attributes formulating *correlations* between the attributes. When attributes' values correspond to some suspicious activity (e.g. collision corresponds to some classical *scheme*), factors are set to be close to 1, and close to -1, when values correspond to non-suspicious activity (e.g. children in the vehicle). Otherwise, they are set to be 0.

Note that assessment of some participant with models AM_{raw} and AM_{bas} is highly dependent on the number of collisions this participant was involved in. More precisely, on the number of the terms in the sums in equations (20), (23) (which is exactly the degree of the corresponding vertex). Although this property is not vain, we still implicitly assume we posses *all* of the collisions a certain participant was involved in. This assumption is often not true (in practice).

We propose a third model diminishing the mentioned assumption. Let $\overline{d_G}$ be the average degree of the vertices in network G, $\overline{d_G} = ave\{d(v_k)|\ v_k \in V_G\}$. The model is

$$AM^{mean}_{\cdot} = \frac{\overline{d_G}+d(v_i)}{2}\frac{AM_{\cdot}}{d(v_i)} = \left(1+\frac{\overline{d_G}}{d(v_i)}\right)\frac{AM_{\cdot}}{2}, \quad (27)$$

where $AM_{\cdot}$ can be any of the models AM_{raw}, AM_{bas}. $AM^{mean}_{\cdot}$ averages terms in the sum of the model $AM_{\cdot}$, and multiplies this average by the mean of vertex's degree and the average degree over all the vertices in V_G. Thus a sort of *Laplace smoothing* is employed that pulls the vertex degree toward the average, in order to diminish the importance of this parameter in the final assessment. Empirical analysis in section 5 shows that such a model outperforms the other two.

Knowing scores $s(\cdot)$ for all the entities in some connected component $c \in G$, one can rank them according to the suspicion of their being fraudulent. In order to also compare the entities from various components, scores must be normalized appropriately (e.g. multiplied with the number of collisions represented by component c).

4.4. Final remarks

In the previous section (third module of the system) we focused only on detection of fraudulent participants. Their suspicion scores can now be used for assessment of other entities (e.g. collisions, vehicles), using one of the assessment models proposed in section 4.3.

When all of the most highly ranked participants in some suspicious component are directly connected to each other (or through buckets), they are proclaimed to belong to the same group of fraudsters. Otherwise they belong to several groups. During the investigation process, the domain expert or investigator analyzes these groups and determines further actions for resolving potential fraud. Entities are investigated in the order induced by scores $s(\cdot)$.

Networks also allow a neat visualization of the assessment (see Fig. 6).

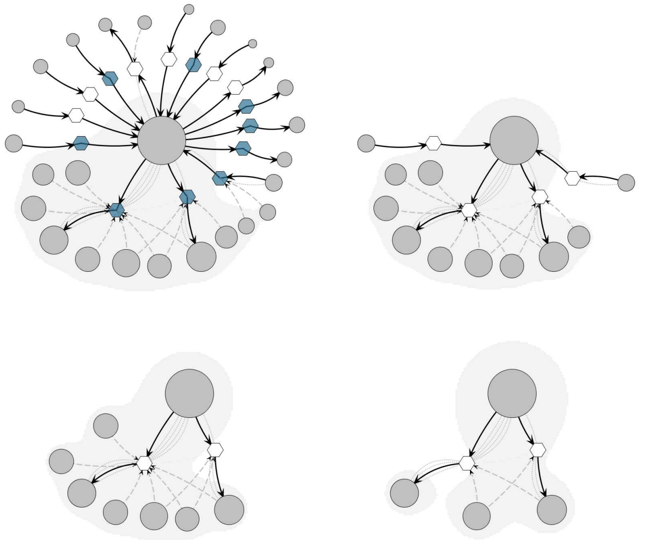

Fig. 6: Four *COPTA* networks showing same group of collisions. Size of the participants' vertices correspond to their suspicion score; only participants with score above some threshold, and connecting collisions, are shown on each network. The contour was drawn based on the *harmonic mean* distance to every vertex, weighted by the suspicion scores. (Blue) filled collisions' vertices in the first network correspond to collisions that happened at night.

5. Evaluation with the prototype system

We implemented a prototype system to empirically evaluate the performance of the proposition. Furthermore, various components of the system are analyzed and compared to other approaches. To simulate realistic conditions, the data set used for evaluation consisted only of the data, that can be easily (automatically) retrieved from police records (*semistructured data*). We report results of the assessment of participants (not e.g. collisions).

5.1. Data

The data set consists of 3451 participants involved in 1561 collisions in Slovenia between the years 1999 and 2008. The set was made by merging two data sets, one labeled and one unlabeled.

The first, labeled, consists of collisions corresponding to previously identified fraudsters and some other participants, which were investigated in the past. In a few cases, when *class* of a participant could not be determined, it was set according to the domain expert's and investigator's belief. As the purpose of our system is to identify groups of fraudsters, and not some isolated fraudulent collisions, (almost) all isolated collisions were removed from this set. It is thus a bit smaller (i.e. 211 participants, 91 collisions), but still large enough to make the assessment.

To achieve a more realistic class distribution and better statistics for *PRIDIT* analysis, the second larger data set was merged with the first. The set consists of various collisions chosen (almost) at random, although some of them are still related with others. Since random data sampling is not advised for relational data (Jensen, 1999), this set is used explicitly for *PRIDIT* analysis. Both data sets consist of only standard collisions (e.g. there are no chain collisions involving numerous vehicles or coaches with many passengers).

Class distribution for the data set can be seen in table 2.

Class distribution

	Count	*Proportion*	
Fraudster	46	1.3%	21.8%
Non-fraudster	165	4.8%	78.2%
Unlabeled	3240	93.9%	

Table 2: Class distribution for the data set used in the analysis of the proposed expert system.

The entire assessment was made using the merged data set, while the reported results naturally only correspond to the first (labeled) set. Note that the assessment of entities in some connected component is completely independent of the entities in other components (except for *PRIDIT* analysis).

5.2. Results

Performance of the system depends on random generation of networks, used for detection of suspicious components (second module). We construct 200 random networks for each indicator and each component (equations (11), (12)), however, the results still vary a little. The entire assessment was thus repeated 20 times and the scores were averaged. To assess the ranking of the system, average *AUC* (*Area Under Curve*) scores were computed, $\overline{AUC}$. Results given in tables 5, 6, 7, 8 are all $\overline{AUC}$.

In order to obtain a standard for other analyses, we first report the performance of the system that uses *PRIDIT* analysis for fraudulent components detection, and *IAA* algorithm with model AM_{bas}^{mean} for fraudulent entities detection, denoted IAA_{bas}^{mean} (see table 3). Various metrics are computed, i.e. *classification accuracy* (*CA*), *recall* (*true positive rate*), *precision* (*positive predictive value*), *specificity* (1− *false positive rate*), *F*1 *score* (*harmonic mean of recall and precision*) and $\overline{AUC}$. All but last are metrics that assess the classification of some approach, thus a threshold for suspicion scores must be defined. We report the results from the first run that minimize the total cost, assuming the cost of misclassified fraudsters and non-fraudsters is the same. Same holds for confusion matrix seen in table 4.

Golden standard

CA	0.8720
Recall	0.8913
Precision	0.6508
Specificity	0.8667
*F*1 *score*	0.7523
$\overline{AUC}$	**0.9228**

Table 3: Performance of the system that uses *PRIDIT* analysis with IAA_{bas}^{mean} algorithm. Various metrics are reported; all except $\overline{AUC}$ are computed so the total cost (on the first run) is minimal.

Confusion matrix

	Suspicious	*Unsuspicious*
Fraudster	41	5
Non-fraudster	22	143

Table 4: Confusion matrix for the system that uses *PRIDIT* analysis with IAA_{bas}^{mean} algorithm (determined so the total cost on the first run is minimal).

We proceed with an in-depth analysis of the proposed *IAA* algorithm. Table 5 shows the results of the comparison of different assessment models, i.e. IAA_{raw}, IAA_{bas}, IAA_{raw}^{mean} and IAA_{bas}^{mean}. Factors for models IAA_{bas} and IAA_{bas}^{mean} (equation (25)) were set by the domain expert, with the help of statistical analysis of data from collisions. To further analyze the impact of factors on final assessment, an additional set of factors was defined by the authors. Values were set due to authors' intuition; corresponding models are IAA_{int} and IAA_{int}^{mean}. Results of the analysis can be seen in table 6.

Assessment models

PRIDIT			
IAA_{raw}	IAA_{bas}	IAA_{raw}^{mean}	IAA_{bas}^{mean}
0.8872	0.9145	0.8942	0.9228

Table 5: Comparison of different assessment models for *IAA* algorithm (after *PRIDIT* analysis).

As already mentioned, the performance of the *IAA* algorithm depends on the number of iterations made in the assessment (see section 4.3). We have thus plotted the *AUC* scores with respect to the number of iterations made (for the first run), in order to clearly see the dependence; plots for IAA_{raw}^{mean}, IAA_{bas}^{mean} can be seen in Fig. 7, Fig. 8 respectively. We also show that superior performance can be achieved, if the number of iterations

Factors

ALL		
IAA_{raw}^{mean}	IAA_{int}^{mean}	IAA_{bas}^{mean}
0.8188	0.8435	0.8787

PRIDIT		
IAA_{raw}^{mean}	IAA_{int}^{mean}	IAA_{bas}^{mean}
0.8942	0.9086	0.9228

Table 6: Analysis of the impact of factors on the final assessment (on all the components and after *PRIDIT* analysis).

is set dynamically. More precisely, the number of iterations made for some component $c \in C(G)$ is

$$max\{\overline{d_G}, d(c)\}, \tag{28}$$

where $d(c)$ is the diameter of c and $\overline{d_G}$ the average diameter over all the components. All other results reported in this analysis used such a dynamic setting.

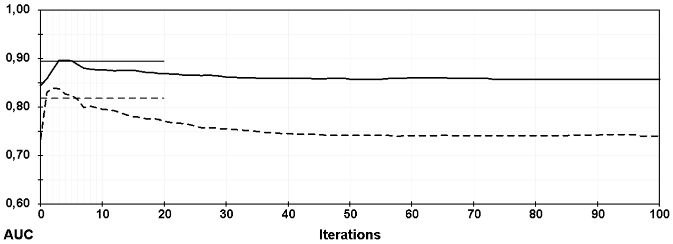

Fig. 7: *AUC* scores with respect to the number of iterations made in the *IAA* algorithm. Solid curves correspond to IAA_{raw}^{mean} algorithm after *PRIDIT* analysis and dashed curves to IAA_{raw}^{mean} algorithm ran on all the components. Straight line segments show the scores achieved with dynamic setting of the number of iterations (see text).

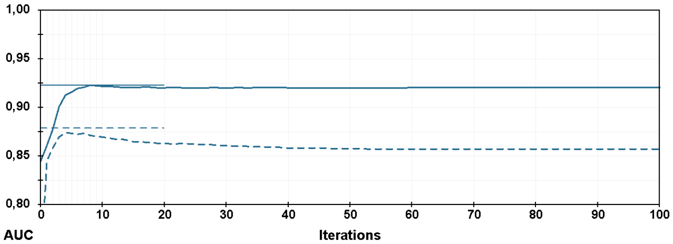

Fig. 8: *AUC* scores with respect to the number of iterations made in the *IAA* algorithm. Solid curves correspond to IAA_{bas}^{mean} algorithm after *PRIDIT* analysis and dashed curves to IAA_{bas}^{mean} algorithm ran on all the components. Straight line segments show the scores achieved with dynamic setting of the number of iterations (see text).

Due to the lack of other expert system approaches for detecting groups of fraudsters, or even individual fraudsters (according to our knowledge), no comparative analysis of such kind could be made. The proposed *IAA* algorithm is thus compared against several well known measures for anomaly detection in networks – *betweenness centrality* (*BetCen*), *closeness centrality* (*CloCen*), *distance centrality* (*DisCen*) and *eigenvector centrality* (*EigCen*) (Freeman, 1977, 1979). They are defined as

$$BetCen(v_i) = \sum_{v_j,v_k \in V_c} \frac{g_{v_j,v_k}(v_i)}{g_{v_j,v_k}}, \tag{29}$$

$$CloCen(v_i) = \frac{1}{n_c - 1} \sum_{v_j \in V_c \setminus v_i} d(v_i, v_j), \tag{30}$$

$$DegCen(v_i) = \frac{d(v_i)}{n_c - 1}, \tag{31}$$

$$EigCen(v_i) = \frac{1}{\lambda} \sum_{\{v_i,v_j\} \in E_c} EigCen(v_j), \tag{32}$$

where n_c is the number of vertices in component c, $n_c = |V_c|$, λ is a constant, g_{v_j,v_k} is the number of geodesics between vertices v_j and v_k and $g_{v_j,v_k}(v_i)$ the number of such geodesics that pass through vertex v_i, $i \neq j \neq k$. For further discussion see (Freeman, 1977, 1979; Newman, 2003).

These measures of centrality were used to assign suspicion score to each participant; scores were also appropriately normalized as in the case of *IAA* algorithm. For a fair comparison, measures were compared against the model that uses no intrinsic attributes of entities, i.e. IAA_{raw}^{mean}. The results of the analysis are shown in table 7.

IAA algorithm

ALL				
BetCen	*CloCen*	*DegCen*	*EigCen*	IAA_{raw}^{mean}
0.6401	0.8138	0.7428	0.7300	0.8188

PRIDIT				
BetCen	*CloCen*	*DegCen*	*EigCen*	IAA_{raw}^{mean}
0.6541	0.8158	0.8597	0.8581	0.8942

Table 7: Comparison of the *IAA* algorithm against several well known measures for anomaly detection in networks (on all the components and after *PRIDIT* analysis). For a fair comparison, no intrinsic attributes are used in the *IAA* algorithm (i.e. model AM_{raw}^{mean}).

Next, we analyzed different approaches for detection of fraudulent components (see table 8). The same set of 9 indicators was used for the majority voter (equation (13)) and for *(P)RIDIT* analysis (equation (17)). For the latter, we use a variant of *random undersampling* (*RUS*), to cope with skewed class distribution. We output most highly ranked components, thus the set of selected components contain 4% of all the collisions (in the merged data set) Analyses of automobile insurance fraud mainly agree that up to 20% of all the collisions are fraudulent, and up to 20% of the latter correspond to non-opportunistic fraud (various resources). However, for the majority voter, such an approach actually decreases performance – we therefore report results where all components, with at least half of the indicators set, are selected.

Several individual indicators, achieving superior performance, are also reported. Indicator I_{BetCen} is based on betweenness centrality (equation (30)), I_{MinCov} on minimum vertex cover and $I_{l^{-1}}$ on l^{-1} *measure* defined as the harmonic mean distance between every pair of vertices in some component c,

$$l^{-1} = \frac{1}{\frac{1}{2}n_c(n_c+1)} \sum_{v_i,v_j \in V_c, i \geq j} d(v_i, v_j)^{-1}. \quad (33)$$

Fraudulent components

I_{MinCov}	$I_{l^{-1}}$	I_{BetCen}	*MAJOR*	*RIDIT*	*PRIDIT*
		ALL			
0.6019	0.6386	0.6774	0.7946	0.6843	0.7114

I_{MinCov}	$I_{l^{-1}}$	I_{BetCen}	*MAJOR*	*RIDIT*	*PRIDIT*
		IAA^{mean}_{bas}			
0.6119	0.8494	0.8549	0.8507	0.9221	0.9228

Table 8: Comparison of different approaches for detection of fraudulent components (prior to no fraudulent entities detection and IAA^{mean}_{bas}).

We last analyze the importance of proper data representation for detection of groups of fraudsters – the use of networks. Networks were thus transformed into flat data and some standard unsupervised learning techniques were examined (e.q. *k-means*, *hierarchical clustering*). We obtained no results comparable to those given in table 3.

Furthermore, we tested nine standard supervised data-mining techniques to analyze the compensation of data labels for the inappropriate representation of data. We used (default) implementations of classifiers in *Orange* data-mining software (Demsar et al., 2004) and 20-*fold cross validation* was employed as the validation technique. Best performance, up to $AUC \approx 0.86$, was achieved with *Naive Bayes*, *support vector machines*, *random forest* and, interestingly, also *k-nearest neighbors* classifier. Scores for other approaches were below $AUC = 0.80$ (e.g. *logistic regression*, *classification trees*, etc.).

6. Discussion

Empirical evaluation from the previous section shows that automobile insurance fraud can be detected using the proposition. Moreover, the results suggest that appropriate data representation is vital – even a simple approach over networks can detect a great deal of fraud. The following section discusses the results in greater detail (in the order given).

Almost all of the metrics obtained with *PRIDIT* analysis and IAA^{mean}_{bas} algorithm, *golden standard*, are very high (table 3). Only precision appears low, still this results (only) from the skewed class distribution in the domain. The $F1$ measure is consequently also a bit lower, else the performance of the system is more than satisfactory. The latter was confirmed by the experts and investigators from a Slovenian insurance company, who were also pleased with the visual representation of the results.

The confusion matrix given in table 4 shows that we correctly classified almost 90% of all fraudsters and over 85% of nonfraudsters. Only 5 fraudsters were not detected by the prototype system. We thus obtained a particularly high recall, which is essential for all fraud detection systems. The majority of unlabeled participants were classified as unsuspicious (not shown in table 4), but the corresponding collisions are mainly isolated and the participants could have been trivially eliminated anyway (for our purposes).

We proceed with discussion of different assessment models (table 5). Performance of the simplest of the models IAA_{raw}, which uses no domain expert's knowledge, could already prove sufficient in many circumstances. It can still be significantly improved by also considering the factors, set by the domain expert (model IAA_{bas}). Model $IAA^{mean}_{\cdot}$ further improves the assessment of both (simple) models, confirming the hypothesis behind it (see section 4.3). Although the models (probably incorrectly) assume that the fraudulence of an entity is linear (in the fraudulences of the related entities), they give a good approximation of the fraudulent behavior.

The analysis of the factors used in the models confirms their importance for the final assessment. As expected, model IAA^{mean}_{bas} outperforms IAA^{mean}_{int}, and the latter outperforms IAA^{mean}_{raw} (table 6). First, this confirms the hypothesis that domain knowledge can be incorporated into the model using factors (as defined in section 4.3). Second, it shows that better understanding of the domain can improve assignment of factors. Combination of both makes the system extremely flexible, allowing for detection of new types of fraud immediately after they have been noticed by the domain expert or investigator.

As already mentioned, running the *IAA* algorithm for too long over-fits the model and decreases algorithm's final performance (see Fig. 7, Fig. 8, note different scales used). Early iterations of the algorithm still increase the performance in all cases analyzed, which proves the importance of iterative assessment as opposed to *single-pass* approach. However, after some particular number of iterations has been reached, performance decreases (at least slightly). Also note that the decrease is much larger in the case of AM^{mean}_{raw} model than AM^{mean}_{bas}, indicating that the latter is superior to the first. We propose to use this *decrease in performance* as an additional evaluation of any model used with *IAA*, or similar, algorithm.

It is preferable to run the algorithm for only a few iterations for one more reason. Networks are often extremely large, especially when they describe many characteristics of entities. In this case, running the algorithm until some fixed point is simply not feasible. Since the prototype system uses only the basic attributes of the entities, the latter does not present a problem.

The number of iterations that achieves the best performance clearly depends on various factors (data set, model, etc.). Our evaluation shows that superior, or at least very good, performance (Fig. 7, Fig. 8) can be achieved with the use of a dynamic setting of the number of iterations (equation (28)).

When no detection of fraudulent components is done, the comparison between *IAA* algorithm and measures of centrality shows no significant difference (table 7). On the other hand, when we use *PRIDIT* analysis for fraudulent components detection, the *IAA* algorithm dominates others. Still, the results obtained with *DegCen* and *EigCen* are comparable to those obtained with supervised approaches over flat data. This shows that even a simple approach can detect a reasonably large portion of fraud, if appropriate representation of data is used (networks).

The analysis of different approaches for detection of fraudulent components produces no major surprises (table 8) – the best results are obtained using *(P)RIDIT* analysis. Note that a single indicator can match the performance of majority classifier *MAJOR*, confirming its naiveness (see section 4.2); exceptionally high $\overline{AUC}$ score obtained by *MAJOR*, prior to no fraudulent entities detection, only results from the fact, that the returned set of suspicious components is almost 10-times smaller than for other approaches. The precision of the approach is thus much higher, but for the price of lower recall (useless for fraud detection).

We have already discussed the purpose of hierarchical detection of groups of fraudsters – to simplify detection of fraudulent entities with appropriate detection of fraudulent components. Another implication of such an approach is also simpler, or is some cases even feasible, *data collection* process. As the detection of components is done using only the relations between entities (relational attributes), a large portion of data can be discarded without knowing the values of any of the intrinsic attributes. This characteristic of the system is vital when deploying in practice – (complete) data often cannot be obtained for all the participants, due to sensitivity of the domain.

Last, we briefly discuss the applicability of the proposition in other domains. The presented *IAA* algorithm can be used for arbitrary assessment of entities over some relational domain, exploring the relations between entities with no demand for an (initial) labeled data set. When every entity is well defined with (only) the entities directly related to it, considering the context observed, one of the proposed assessment models can also be used. Furthermore, the presented framework (four modules of the system) could be employed for fraud detection in other domains. The system is also applicable for use in other domains, where we are interested in groups of related entities sharing some particular characteristics. The framework exploits the relations between entities, in order to improve the assessment, and is structured hierarchically, to make it applicable in practice.

7. Conclusion

The article proposes a novel expert system approach for detection of groups of automobile insurance fraudsters with networks. Empirical evaluation shows that such fraud can be efficiently detected using the proposition and, in particular, that proper representation of data is vital. For the system to be applicable in practice, no labeled data set is used. The system rather allows the imputation of domain expert's knowledge, and it can thus be adopted to new types of fraud as soon as they are noticed. The approach can aid the domain investigator to detect and investigate fraud much faster and more efficiently. Moreover, the employed framework is easy to implement and is also applicable for detection (of fraud) in other relational domains.

Future research will be focused on further analyses of different assessment models for *IAA* algorithm, considering also the nonlinear models. Moreover, the *IAA* will be altered into an *unsupervised algorithm*, learning the factors of the model in an unsupervised manner during the actual assessment. The factors would thus not have to be specified by the domain expert. Applications of the system in other domains will also be investigated.

Acknowledgements

Authors thank Matjaž Kukar, from University of Ljubljana, and Jure Leskovec, (currently) from Cornell University, for all the effort and useful suggestions; Tjaša Krisper Kutin for lectoring the article; and Optilab d.o.o. for the data used in the study. This work has been supported by the Slovene Research Agency *ARRS* within the research program P2-0359.

References

Artis, M., Ayuso, M., & Guillen, M. (2002). Detection of automobile insurance fraud with discrete choice models and misclassified claims. *Journal of Risk and Insurance*, *69*(3), 325–340.

Ball, F., Mollison, D., & Scalia-Tomba, G. (1997). Epidemics with two levels of mixing. *Annals of Applied Probability*, *7*(1), 46–89.

Barabasi, A. L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, *286*(5439), 509–512.

Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, *17*(3), 235–249.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, *30*(1-7), 107–117.

Brockett, P. L., Derrig, R. A., Golden, L. L., Levine, A., & Alpert, M. (2002). Fraud classification using principal component analysis of RIDITs. *Journal of Risk and Insurance*, *69*(3), 341–371.

Brockett, P. L., & Levine, A. (1977). Characterization of RIDITs. *Annals of Statistics*, *5*(6), 1245–1248.

Brockett, P. L. (1981). A note on the numerical assignment of scores to ranked categorical-data. *Journal of Mathematical Sociology*, *8*(1), 91–101.

Bross, I. (1958). How to use RIDIT analysis. *Biometrics*, *14*(1), 18–38.

Chakrabarti, S., Dom, B., & Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. In *Proceedings of the International Conference on Management of Data*, (pp. 307–318).

Demsar, J., Zupan, B., Leban, G., & Curk, T. (2004). Orange: From experimental machine learning to interactive data mining. In *Knowledge Discovery in Databases: PKDD 2004*, (pp. 537–539). Springer Berlin, Heidelberg.

Domingos, P., & Richardson, M. (2001). Mining the network value of customers. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, (pp. 57–66).

Eppstein, D., & Wang, J. (2002). A steady state model for graph power laws. In *Proceedings of the International Workshop on Web Dynamics*.

Estevez, P. A., Held, C. M., & Perez, C. A. (2006). Subscription fraud prevention in telecommunications using fuzzy rules and neural networks. *Expert Systems with Applications*, *31*(2), 337–344.

Freeman, L. C. (1979). Centrality in social networks - conceptual clarification. *Social Networks*, *1*(3), 215–239.

Freeman, L. (1977). A set of measures of centrality based on betweenness. *Sociometry*, *40*(1), 35–41.

Furlan, S., & Bajec, M. (2008). Holistic approach to fraud management in health insurance. *Journal of Information and Organizational Sciences*, *32*(2), 99–114.

Ghosh, A. K., & Schwartzbard, A. (1999). A study in using neural networks for anomaly and misuse detection. In *Proceedings of the Conference on USENIX Security Symposium*, (pp. 12–12).

Girvan, M., & Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences USA*, *99*(12), 7821–7826.

Holder, L. B., & Cook, D. J. (2003). Graph-based relational learning: current and future directions. *SIGKDD Explorations*, *5*(1), 90–93.

Hummel, R. A., & Zucker, S. W. (1983). On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *5*(3), 267–287.

Hu, W., Liao, Y., & Vemuri, V. R. (2007). Robust anomaly detection using support vector machines. In *Proceedings of the International Conference on Machine Learning*.

Jensen, D. (1997). Prospective assessment of AI technologies for fraud detection and risk management. In *Proceedings of the AAAI Workshop on AI Approaches to Fraud Detection and Risk Management*, (pp. 34–38).

Jensen, D. (1999). Statistical challenges to inductive inference in linked data. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, (pp. 59–62).

Kempe, D., Kleinberg, J., & Tardos, E. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, (pp. 137–146).

Kirkos, E., Spathis, C., & Manolopoulos, Y. (2007). Data mining techniques for the detection of fraudulent financial statements. *Expert Systems with Applications*, *32*(4), 995–1003.

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

Kschischang, F. R., & Frey, B. J. (1998). Iterative decoding of compound codes by probability propagation in graphical models. *IEEE Journal on Selected Areas in Communications*, *16*(2), 219–230.

Lu, Q., & Getoor, L. (2003a). Link-based classification using labeled and unlabeled data. In *Proceedings of the ICML Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*.

Lu, Q., & Getoor, L. (2003b). Link-based text classification. In *Proceedings of the IJCAI Workshop on Text Mining and Link Analysis*.

Maxion, R. A., & Tan, K. M. C. (2000). Benchmarking anomaly-based detection systems. In *Proceedings of the International Conference on Dependable Systems and Networks*, (pp. 623–630).

Minka, T. (2001). Expectation propagation for approximate bayesian inference. In *Proceedings of the Conference on Uncertanty in Artificial Intelligence*, (pp. 362–369).

Neville, J., & Jensen, D. (2000). Iterative classification in relational data. In *Proceedings of the Workshop on Learning Statistical Models from Relational Data*, (pp. 13–20).

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, *45*(2), 167–256.

Newman, M. E. J. (2008). Mathematics of networks. In *The New Palgrave Encyclopedia of Economics*. Palgrave Macmillan, Basingstoke.

Noble, C. C., & Cook, D. J. (2003). Graph-based anomaly detection. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 631–636).

Oh, H. J., Myaeng, S. H., & Lee, M. H. (2000). A practical hypertext categorization method using links and incrementally available class information. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval*, (pp. 264–271).

Perez, J. M., Muguerza, J., Arbelaitz, O., Gurrutxaga, I., & Martin, J. I. (2005). Consolidated tree classifier learning in a car insurance fraud detection domain with class imbalance. In *Proceedings of the International Conference on Advances in Pattern Recognition*, (pp. 381–389).

Phua, C., Lee, V., Smith, K., & Gayler, R. (2005). A comprehensive survey of data mining-based fraud detection research. *Artificial Intelligence Review*.

Quah, J. T. S., & Sriganesh, M. (2008). Real-time credit card fraud detection using computational intelligence. *Expert Systems with Applications*, *35*(4), 1721–1732.

Rupnik, R., Kukar, M., & Krisper, M. (2007). Integrating data mining and decision support through data mining based decision support system. *Journal of Computer Information Systems*, *47*(3), 89–104.

Sanchez, D., Vila, M. A., Cerda, L., & Serrano, J. M. (2009). Association rules applied to credit card fraud detection. *Expert Systems with Applications*, *36*(2), 3630–3640.

Sen, P., & Getoor, L. (2007). Link-based classification. Tech. Rep. CS-TR-4858, University of Maryland.

Sun, J., Qu, H., Chakravarti, D., & Faloutsos, C. (2005). Relevance search and anomaly detection in bipartite graphs. *ACM SIGKDD Explorations Newsletter*, *7*(2), 48–55.

Viaene, S., Dedene, G., & Derrig, R. A. (2005). Auto claim fraud detection using bayesian learning neural networks. *Expert Systems with Applications*, *29*(3), 653–666.

Viaene, S., Derrig, R. A., Baesens, B., & Dedene, G. (2002). 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.

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*(6684), 440–442.

Weisberg, H. I., & Derrig, R. A. (1998). Quantitative methods for detecting fraudulent automobile bodily injury claims. *Risques*, *35*, 75–101.

Yang, W. S., & Hwang, S. Y. (2006). A process-mining framework for the detection of healthcare fraud and abuse. *Expert Systems with Applications*, *31*(1), 56–68.