



HAL
open science

Pattern Mining for Anomaly Detection in Graphs: Application to Fraud in Public Procurement

Lucas Potin, Rosa Figueiredo, Vincent Labatut, Christine Largeron

► **To cite this version:**

Lucas Potin, Rosa Figueiredo, Vincent Labatut, Christine Largeron. Pattern Mining for Anomaly Detection in Graphs: Application to Fraud in Public Procurement. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases 2023 (ECML PKDD), Sep 2023, Torino, Italy. 10.1007/978-3-031-43427-3_5 . hal-04131485

HAL Id: hal-04131485

<https://hal.science/hal-04131485v1>

Submitted on 16 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Pattern Mining for Anomaly Detection in Graphs: Application to Fraud in Public Procurement

Lucas Potin¹, Rosa Figueiredo¹, Vincent Labatut¹, and Christine Largeron²

¹ Laboratoire Informatique d'Avignon – UPR 4128, F-84911, Avignon, France
`{firstname}-{lastname}@univ-avignon.fr`

² Laboratoire Hubert Curien – UMR 5516, F-42023, Saint-Etienne, France
`christine.largeron@univ-st-etienne.fr`

Abstract. In the context of public procurement, several indicators called red flags are used to estimate fraud risk. They are computed according to certain contract attributes and are therefore dependent on the proper filling of the contract and award notices. However, these attributes are very often missing in practice, which prohibits red flags computation. Traditional fraud detection approaches focus on tabular data only, considering each contract separately, and are therefore very sensitive to this issue. In this work, we adopt a graph-based method allowing leveraging relations between contracts, to compensate for the missing attributes. We propose PANG (Pattern-Based Anomaly Detection in Graphs), a general supervised framework relying on pattern extraction to detect anomalous graphs in a collection of attributed graphs. Notably, it is able to identify *induced* subgraphs, a type of pattern widely overlooked in the literature. When benchmarked on standard datasets, its predictive performance is on par with state-of-the-art methods, with the additional advantage of being explainable. These experiments also reveal that induced patterns are more discriminative on certain datasets. When applying PANG to public procurement data, the prediction is superior to other methods, and it identifies subgraph patterns that are characteristic of fraud-prone situations, thereby making it possible to better understand fraudulent behavior.

Keywords: Pattern Mining · Graph Classification · Public Procurement · Fraud Detection.

Cite as: L. Potin, R. Figueiredo, V. Labatut & C. Largeron. “Pattern Mining for Anomaly Detection in Graphs: Application to Fraud in Public Procurement”, *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Springer, 2023. DOI: TBD

1 Introduction

Public procurement refers to the purchase of goods, services and works by a public authority (the buyer), from a legal entity governed by public or private

law (the winner). In the European Union, when the contract exceeds some price threshold, the buyer must first advertise a call for tenders defining its needs in detail, and later the corresponding award notice, which describes the content of the contract eventually concluded with one or more winners. These documents must be published in the *Official Journal of the European Union* (OJEU). The online version of this journal, called the *Tenders Electronic Daily* (TED) [11], publishes more than 650,000 procurement notices a year. Consequently, the public procurement sector provides a huge amount of publicly available data.

Historically, anomalies in public procurement, which refer to doubtful behavior, are linked to specific characteristics associated with contracts. In the literature, these characteristics are called *red flags*, and are used as indicators of potential fraud [16,39,13,15]. For instance, modifying the contract price during the procedure, or receiving a single offer for a given call for tenders, are typically considered as red flags [36]. But the information required to compute these red flags is not always available. In the French subset of the TED, some essential attributes are largely missing [37], e.g. the number of offers answering a call for tenders is not documented in 30% of the cases. For such contracts, one can compute only *partial* red flags, in the best of cases, or even no red flags at all.

Anomaly detection approaches are commonly used in fraud detection [38]. However, when applied to public procurement, most studies are based on *tabular* data [4,3], i.e. each contract is considered separately, as a set of attribute values. Only a very few authors try to take advantage of the *relationships* between contracts by adopting a graph-based approach. Fazekas & Tóth propose the CRI, a composite score combining several red flags, and leverage graphs [14], but only to visualize its distribution over their dataset. Wachs *et al.* [46] use graphs in order to estimate the proportion of red flags in the core agents, i.e. buyers and winners with the most frequent relationships, compared to the others. However, to the best of our knowledge, no method in the literature dedicated to anomaly or fraud detection in public procurement uses graphs to create predictive models.

This leads us to propose a graph-based method to identify anomalies in public procurement. Our work makes three main contributions. First, we propose the *PANG* framework (Pattern-Based Anomaly Detection in Graphs), that leverages pattern mining to solve this problem. When evaluated on a benchmark of standard datasets, its performance is on par with state-of-the-art methods, with the additional advantage of being explainable. In addition, it allows looking for different types of patterns, including *induced* subgraphs, which are generally overlooked in the literature. Our second contribution is to show empirically that such subgraphs can result in better classification performance on certain datasets. As a third contribution, we apply our generic framework to public procurement data, and identify the relevant patterns characterizing risky behaviors.

The rest of the article is structured as follows. Section 2 gives an overview of the literature regarding graph anomaly detection and graph pattern mining. Section 3 introduces the terminology used throughout this paper, as well as our problem formulation. Section 4 describes our framework PANG and assesses its

performance on standard datasets. Section 5 applies PANG to public procurement. Finally, we comment the main aspects of our work in Section 6.

2 Related Work

The goal of anomaly detection is to detect behaviors significantly differing from expected norms. The methods dealing with this task on graphs either focus on single elements (vertices, edges) or larger structures (subgraphs, graphs) [2,30,24]. When considering whole graphs, the task can be seen as a classification problem consisting in labelling the graph as normal or anomalous. The standard approach consists in building a vector-based representation of the graph, in order to apply classic data mining tools [30]. Most recent works focus on deep learning methods such as Graph Neural Networks (GNN) [10,28,29], which not only learn this representation, but also tackle the classification task. However, one limitation of these methods lies in the lack of explainability: while some approaches have been proposed to make GNNs explainable [51], achieving this goal is non-trivial, especially when considering graphs with edge features. An alternative is to build the representation in a more controlled way, in order to retain its semantics [50]. Among the methods following this path, pattern-based approaches rely on the subgraphs that compose the graphs [1]. They require retrieving the most characteristic of these patterns, generally the most frequent ones, in order to represent each graph in terms of absence or presence of these patterns.

There are different algorithms to extract *frequent* subgraphs from a collection of graphs [49,17], i.e. patterns appearing in more graphs than a fixed threshold. The main issue encountered with this approach is the pattern explosion problem, which states that the number of patterns increases exponentially when decreasing this threshold. To alleviate the computational cost, some algorithms mine more constrained patterns, such as *closed* frequent patterns [41], *maximal* frequent patterns [31], or *approximate* patterns [26]. As these notions are not the focus of this paper, we refer the reader to [34] for further details.

Moreover, all frequent patterns may not be relevant when dealing with a graph classification problem: some could occur equally in all classes, and thus provide no information to distinguish them. To overcome this issue, some methods have been proposed to mine *discriminative* patterns. Leap [48] relies on a notion of structural proximity when building its search tree, that lets it compare branches in order to avoid exploring those that are similar. CORK [44] is based on a metric that evaluates a pattern in relation to a collection of patterns already selected, which allows accounting for the proximity between frequent patterns. Moreover, this metric is submodular, and can thus be integrated into tools such as gSpan [49] to mine discriminative patterns efficiently. It also allows CORK to automatically select the number of patterns to extract. In [23], the notion of discriminative pattern is extended in order to mine *jumping emerging* patterns: subgraphs appearing in only one class. However, this notion is very restricted, as it requires that a pattern *never* appears in one of the two classes.

As a consequence, in practice, it often leads to very infrequent patterns [27]. Our objective is to propose a generic classification framework which allows choosing the number of discriminative patterns to keep, as well as their type and, then to apply it for identifying fraud in public procurement.

3 Problem Formulation

To detect fraud in public procurement, we adopt a network representation inspired by information retrieval or text mining, and previously successfully used for chemical compound classification [33]. In the same way that a document can be modeled as a bag-of-words, we propose to represent a graph as a bag-of-subgraphs, i.e. the set of its constituting subgraphs, called *patterns*. To do this, we construct a global dictionary constituted of the patterns appearing in a collection of attributed graphs. Based on this dictionary, each graph can then be represented as a fixed-length numerical vector, which can be used as an input by any standard machine learning algorithm. In this section, we first describe how we define such vector-based representation, and then formulate our anomaly detection task as a classification problem.

Definition 1 (Attributed Graph). *An attributed graph is defined as a tuple $G = (V, E, \mathbf{X}, \mathbf{Y})$ in which V is the set of n vertices, E the set of m edges of G , \mathbf{X} the $n \times d_v$ matrix whose row \mathbf{x}_i is the d_v -dimensional attribute vector associated with vertex $v_i \in V$, and \mathbf{Y} the $m \times d_e$ matrix whose row \mathbf{y}_i is the d_e -dimensional attribute vector associated with edge $e_i \in E$.*

As an illustration, we consider a collection of such graphs, as shown in Figure 1. In this example, each vertex has an attribute corresponding to its color (brown or purple) as well as each edge (green or red).

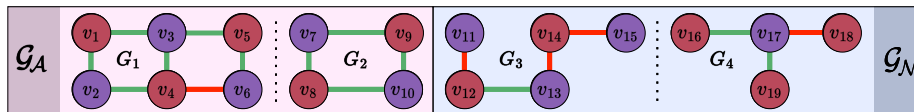


Fig. 1: A collection \mathcal{G} of graphs including the subsets of anomalous (\mathcal{G}_A) and normal (\mathcal{G}_N) graphs.

Let us assume that each graph G has a label ℓ_G picked in $\mathcal{L} = \{A, N\}$, denoting an anomalous or a normal graph, respectively. Importantly, this label is not known for all the graphs at our disposal. Let \mathcal{G} be the set of graphs whose label is known. The set \mathcal{G} can be split into two disjoint subsets: $\mathcal{G} = \mathcal{G}_A \cup \mathcal{G}_N$ ($\mathcal{G}_A \cap \mathcal{G}_N = \emptyset$). Set \mathcal{G}_A contains the anomalous graphs, and \mathcal{G}_N the normal ones. Using the labeled set of graphs \mathcal{G} , our aim is to train a classifier able to predict the unknown label for the other graphs. For this purpose, we use a pattern-based graph representation.

Definition 2 (General Pattern). Let $G = (V, E, \mathbf{X}, \mathbf{Y})$ be an attributed graph. A graph P is a pattern of G if it is isomorphic to a subgraph H of G , i.e. $\exists H \subseteq G : P \cong H$.

As we consider attributed graphs, we adopt the definition of a graph isomorphism proposed by Hsieh *et al.* [21], i.e. an isomorphism must preserve not only edges, but also vertex and edge attributes. We consider that P is a pattern for a set of graphs \mathcal{G} when P is a pattern of at least one of its graphs. Figure 2 shows three examples of patterns of G_1 , and therefore of \mathcal{G} , from Figure 1.

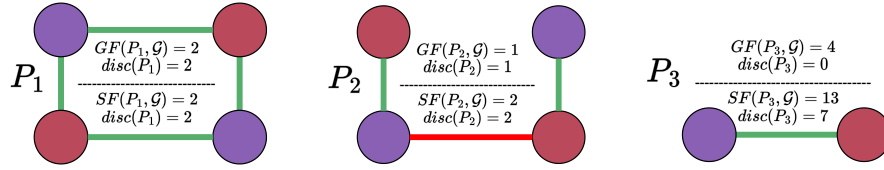


Fig. 2: Three examples of general patterns present in graph G_1 of Figure 1.

It should be noted that, according to Definition 2, a pattern P may not include all the edges originally present in G between the considered vertices. We can restrict this definition by considering *induced* patterns. Similarly to Definition 2, P is an *induced* pattern of G if it is isomorphic to an *induced* subgraph H of G .

Definition 3 (Induced Subgraph). Let $G = (V, E, \mathbf{X}, \mathbf{Y})$ be an attributed graph. The subgraph $H = (V_H, E_H, \mathbf{X}_H, \mathbf{Y}_H)$ induced by a vertex subset $V_H \subseteq V$ is such that $E_H = \{(u, v) \in E : u, v \in V_H\}$, and \mathbf{X}_H and \mathbf{Y}_H retain only the rows of \mathbf{X} and \mathbf{Y} matching V_H and E_H , respectively.

In Figure 2, P_1 is an induced pattern of G_1 . On the contrary, P_2 is a general pattern of G_1 , but not an induced pattern, because edge (v_3, v_5) from G_1 has no image in P_2 . We consider that P is an induced pattern of \mathcal{G} when P is an induced pattern of at least one of its graphs. To measure the importance of a pattern in \mathcal{G} , we now need the notion of *graph frequency*.

Definition 4 (Graph Frequency). The graph frequency $GF(P, \mathcal{G})$ of a pattern P in \mathcal{G} is the number of graphs in \mathcal{G} having P as a pattern:

$$GF(P, \mathcal{G}) = |\{G \in \mathcal{G} : \exists H \subseteq G \text{ s.t. } P \cong H\}|.$$

It indicates the number of graphs having a specific pattern, but does not give any information about the number of times the pattern appears in these graphs. For this, we use the *subgraph frequency*.

Definition 5 (Subgraph Frequency). The subgraph frequency $SF(P, \mathcal{G})$ of a pattern P in \mathcal{G} is its total number of occurrences over all $G \in \mathcal{G}$:

$$SF(P, \mathcal{G}) = \sum_{G \in \mathcal{G}} |\{H \subseteq G : P \cong H\}|.$$

Graph frequency can be used to define the notion of *closed* pattern, which in turn allows finding a more compact set of relevant patterns.

Definition 6 (Closed Pattern). *A pattern P of \mathcal{G} is said to be closed if it has no supergraph P' , or equivalently if P is not the subgraph of any graph P' , such that $GF(P', \mathcal{G}) = GF(P, \mathcal{G})$.*

As a consequence, the set of closed patterns is a subset of the set of general patterns. In our example, there is no supergraph of P_1 appearing in 2 graphs, which makes it a closed pattern of \mathcal{G} .

Regardless of the type of pattern, we note \mathcal{P}_A and \mathcal{P}_N the sets of patterns of \mathcal{G}_A and \mathcal{G}_N , respectively, and \mathcal{P} the complete set of patterns of \mathcal{G} : $\mathcal{P} = \mathcal{P}_A \cup \mathcal{P}_N$. Not all patterns are equally relevant to solve a given task. For instance, in Figure 2, P_3 is much more common than both other patterns in \mathcal{G} from Figure 1. To distinguish them, we rely on the discrimination score from [44], that characterizes each pattern according to its frequency in the two subsets.

Definition 7 (Discrimination Score). *The discrimination score of a pattern P of \mathcal{G} is defined as $disc(P) = |F(P, \mathcal{G}_A) - F(P, \mathcal{G}_N)|$, where F is GF or SF .*

Our definition generalizes that of [44], so that it can be applied to both frequencies (GF and SF). A score close to 0 indicates a pattern that is as frequent in \mathcal{G}_A as in \mathcal{G}_N , while a higher score means that the pattern is more frequent in one of the two subsets. We use this score to rank the patterns in \mathcal{P} , and select the s most discriminative ones ($1 \leq s \leq |\mathcal{P}|$). Some methods, like CORK [44], estimate s automatically, which can be an advantage or a drawback, depending on the level of control desired by the user.

The resulting subset $\mathcal{P}_s \subseteq \mathcal{P}$ constitutes our dictionary, which means that s lets us control the dimension of our graph representation. The representation of each graph $G_i \in \mathcal{G}$ is a vector $\mathbf{h}_i \in \mathbb{R}^s$ whose components measure how important each pattern of \mathcal{P}_s is to G_i . These measures can be computed according to different formula, as discussed in Section 4. Finally, we build the matrix $\mathbf{H} \in \mathbb{R}^{|\mathcal{G}| \times s}$ by considering the vector representations of all the graphs in \mathcal{G} .

Based on this graph representation, our anomaly detection problem amounts to classifying graphs with unknown labels as anomalous or normal. More formally, given the training set composed of a set of graphs $\mathcal{G} = \{G_i, i = 1, \dots, |\mathcal{G}|\}$ with the labels $\ell_{G_i} \in \mathcal{L}$ and the vector representations \mathbf{h}_i , the goal is to learn a function $f : \mathbb{R}^s \rightarrow \{A, N\}$, which associates a label (anomalous or normal) to the vector representation of an unlabeled graph.

4 PANG Framework

4.1 Description of the Framework

To solve our classification problem, we propose the PANG framework (Pattern-Based Anomaly Detection in Graphs), whose source code is publicly available

online³. A preliminary step consists in extracting the graphs, but as it is data-dependent, we defer its description to Section 5.1. The rest of the process is constituted of four steps, as represented in Figure 3:

1. Identify all the patterns of \mathcal{G} and build \mathcal{P} .
2. Select the most discriminative patterns \mathcal{P}_s among them.
3. Use these patterns to build the vector-based representation of each graph.
4. Train a classifier to predict the graph labels based on these representations.

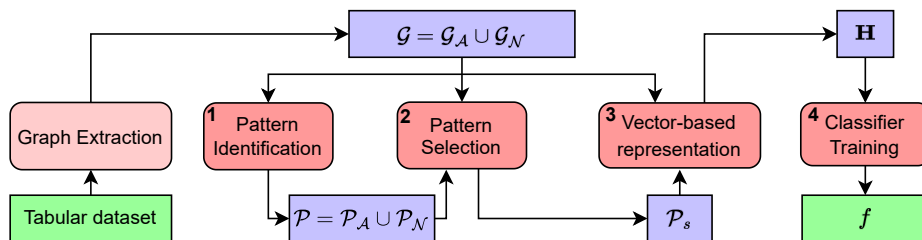


Fig. 3: Processing steps of the proposed PANG framework.

Step #1: Pattern Identification In order to create \mathcal{P} , we use an existing graph pattern extractor. Several tools are available to enumerate patterns, such as gSpan [49], FFSSM [22], or more recently TKG [17] and cgSpan [41].

gSpan and cgSpan respectively search the *frequent* and *closed frequent* patterns in a set of graphs. Both rely on an iterative procedure, which starts from the simplest pattern possible, i.e. a single vertex with a specific attribute, in order to initialize the list of ranked frequent patterns. At each step, the algorithm takes the most frequent pattern according to this list, and tries to extend it by adding an edge. This expansion results in a set of new patterns, which are added or not to the ranked list, according to their frequency. This list is updated over the iterations, until it is no longer possible to find any new pattern with a frequency potentially higher than a predefined threshold.

In the case of cgSpan, the algorithm is able to find the set of closed frequent patterns, which, as explained before, is included in the set of frequent patterns. A smaller set of patterns allows reducing the computation time during the pattern mining phase, but also at post-processing, e.g. when computing the discrimination scores, since there are fewer patterns to consider, and consequently a smaller size for the vector representation.

We choose to use gSpan [49] and cgSpan [41]. The former mines an important number of frequent patterns while requiring less memory than TKG. The latter is able to efficiently identify closed patterns. Both algorithms are implemented in Java, and are available as a part of software SPMF [18], which provides numerous

³ <https://github.com/CompNet/Pang/releases/tag/v1.0.0>

tools for pattern mining. The process used for the induced patterns is based on two steps: first, each pattern is extracted using one of these algorithms. Then, we filter the induced patterns using the ISMAGS algorithm [20] implemented in NetworkX [19].

Step #2: Discriminative Pattern Selection Next, we compute the discrimination score of each extracted pattern as explained in Definition 7. We keep the s most discriminative patterns to construct \mathcal{P}_s .

Step #3: Vector-Based Representation Once we have \mathcal{P}_s , we compute the vector representation of each graph in \mathcal{G} . In this work, we use several approaches. First, we build a binary vector indicating the presence or absence of each pattern in the considered graph. In that case, for each graph $G_i \in \mathcal{G}$ and each pattern $P_j \in \mathcal{P}$, H_{ij} equals 1 if this pattern P_j is present in G_i and 0 otherwise.

This representation is somewhat limited, though, as it ignores how much patterns are present in graphs. To solve this issue, we propose an integer representation based on the number of occurrences *in the graph*. This number is computed with the VF2 algorithm [6], available in Networkx [19]. Given a pattern P and a graph G , VF2 identifies the number of subgraph isomorphisms of P in G , which we store in H_{ij} .

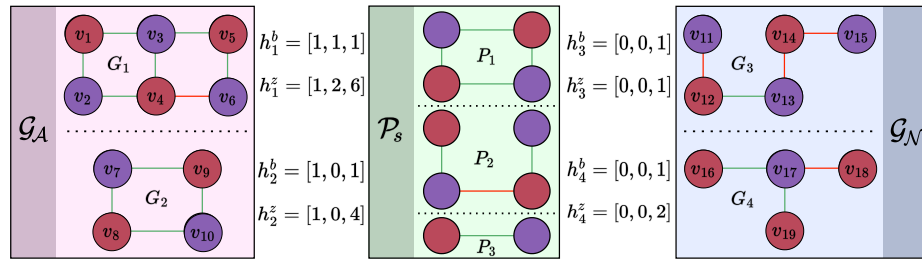


Fig. 4: Binary (\mathbf{h}_j^b) and integer (\mathbf{h}_j^z) vector-based representations of the graphs of Figure 1, using the patterns of Figure 2 as \mathcal{P}_s .

Figure 4 shows the representations obtained for the graphs of Figure 1, using the patterns from Figure 2 as \mathcal{P}_s . Vectors \mathbf{h}_j^b and \mathbf{h}_j^z denote the binary and integer representations of each graph G_j , respectively. It is worth noting that two different graphs can have the same vector representation, as is the case for the binary representation of G_3 and G_4 in our example.

For the sake of consistency, we compute the discrimination scores based on GF when using the binary representation, and on SF when using the integer one.

Step #4: Classifier Training After the previous step, each graph is represented by a fixed-sized vector, no matter its number of vertices or edges. We leverage this

representation to train a classifier into predicting the graph labels. Our framework is general and allows any classifier, but we select C-SVM [5] in this article, as it gives the best experimental results (cf. Appendix A for more classifiers).

4.2 Assessment on Benchmarks

Before focusing on fraud detection in public procurement, we assess PANG on FOPPA, the public procurement dataset that we use in our application, as well as four real-world datasets commonly used in the literature as benchmarks. These last datasets, our protocol and the results of this first experiment are detailed in the following. The FOPPA data is described in Section 5.1.

Experimental Protocol MUTAG [8] contains 188 graphs representing molecules, where vertices are atoms and edges bonds between them. The graphs are distributed over two classes, depending on the molecule mutagenicity. PTC_FR [45] contains 350 graphs, also representing molecules. There are also two graph classes, depending on the molecule carcinogenicity on male and female mice and rats. NCI1 [47] contains 4,110 graphs representing chemical compounds. Each vertex stands for an atom, while edges represent the bonds connecting them. Like before, there are two classes distinguishing the compounds depending on their carcinogenicity. D&D [9] is composed of 1,178 protein structures. Each vertex is an amino acid, and two vertices are connected if they are less than 6 angstroms apart. There are two graph classes corresponding to enzymes vs. non-enzymes. Table 1 shows the main characteristics of these datasets: number of graphs, and average numbers of vertices and edges.

Datasets	MUTAG	PTC_FR	NCI1	D&D
Number of graphs	188	350	4,110	1,178
Average number of vertices	17.93	25.56	29.87	284.72
Average number of edges	19.79	25.96	32.30	715.66

Table 1: Characteristics of the 4 benchmark datasets.

Regarding graph representations, we compute the six types proposed in PANG:

- PANG_GenBin: binary representation considering general patterns.
- PANG_GenOcc: integer representation considering general patterns.
- PANG_IndBin: binary representation using only induced patterns.
- PANG_IndOcc: integer representation using only induced patterns.
- PANG_CloBin: binary representation using only closed patterns.
- PANG_CloOcc: integer representation using only closed patterns.

We compare our results with four different types of baselines. First, as an alternative pattern-based method, we use CORK (cf. Section 2), which automatically estimates the size of the representation. The second baseline type is graph kernels. We use the kernel matrices of the graphs as representations, associating each row of the matrix with the corresponding graph. These matrices are computed from the implementation of the WL kernel [42] and the WL_OA [25] kernel, both available in the GraKel [43] library. The third type is whole graph embedding neural methods, for which we use Graph2Vec [35], available in the KarateClub library [40]. We set an embedding size of 128, which is standard in the literature. For each of these representations, we train a C-SVM as indicated in Step 4 of Section 4.1.

The fourth baseline type is Graph Neural Networks, with DGCNN [52]. This method produces a graph representation, which can be fetched to the SVM, but it can also perform the classification step directly. The results reported here are the best ones, obtained in this second setting, using the implementation from StellarGraph [7], with the optimal parameter values as indicated in [52].

Experimental Results We adopt a 10-fold cross-validation to assess classifier performance. Table 2 shows the average F -Score (with standard deviation) for the *Anomalous* class. Each column corresponds to one of the considered datasets: 4 benchmarks and FOPPA.

Representation	MUTAG	NCI1	D&D	PTC	FOPPA
PANG_GenBin	0.85 (0.05)	0.79 (0.02)	0.77 (0.03)	0.60 (0.13)	0.93 (0.02)
PANG_GenOcc	0.87 (0.04)	0.77 (0.02)	0.75 (0.02)	0.58 (0.10)	0.91 (0.03)
PANG_IndBin	0.87 (0.05)	0.79 (0.02)	0.76 (0.03)	0.59 (0.13)	0.95 (0.01)
PANG_IndOcc	0.87 (0.03)	0.79 (0.01)	0.75 (0.03)	0.56 (0.07)	0.92 (0.02)
PANG_CloBin	0.86 (0.05)	0.78 (0.03)	0.75 (0.03)	0.57 (0.15)	0.94 (0.03)
PANG_CloOcc	0.88 (0.04)	0.76 (0.02)	0.71 (0.04)	0.54 (0.11)	0.92 (0.02)
CORK	0.66 (0.08)	0.78 (0.02)	0.73 (0.03)	0.54 (0.06)	0.63 (0.05)
WL	0.86 (0.06)	0.83 (0.01)	0.82 (0.01)	0.57 (0.06)	0.90 (0.05)
WL_OA	0.86 (0.06)	0.81 (0.03)	0.77 (0.03)	0.55 (0.11)	0.90 (0.05)
Graph2Vec	0.84 (0.07)	0.82 (0.01)	0.72 (0.03)	0.61 (0.11)	0.91 (0.04)
DGCNN	0.86 (0.04)	0.74 (0.01)	0.79 (0.01)	0.58 (0.05)	0.89 (0.01)

Table 2: F -Scores (\pm standard deviation) for the *Anomalous* class.

No method dominates the others over all datasets, therefore we can assume that some graph representations are more relevant to model certain systems. We plan to investigate this question further, but this is out of this article’s scope. The performance of PANG is systematically above CORK, its most similar method. This is because, on the considered datasets, CORK identifies a very restricted set of discriminative patterns and trades classification performance against representation size. Moreover, PANG is on par with the remaining methods on

NCI1, D&D and PTC, and has the best performance on MUTAG and, importantly, on FOPPA, our application dataset. Thus, we assume that PANG is able to capture the same information as embedding- and GNN-based methods. On the one hand, it requires numerous patterns to be mined, and is therefore more time-consuming than these methods. On the other hand, it has the advantage of being interpretable, allowing us to identify the most discriminative patterns. This is why we apply it to fraud detection in public procurement, in Section 5.

5 Public Procurement Use Case

In this section, we apply PANG to real data representing public procurement. We first describe the process used to extract graphs from a database of French public procurement contracts (Section 5.1), then we discuss our results (Section 5.2).

5.1 Extraction of the Graph Dataset

Raw Data The FOPPA [37] database lists all French contracts award notices published at the European level. Each such contract involves *at least* two economic *agents*: a buyer and a winner, and may be constituted of several lots. It is described by a collection of attributes such as the total price, the number of offers, the bid ranking criteria, and whether the procedure was accelerated. In this paper, we consider the specific subset of contracts concerning period 2015–19, containing 417,809 lots.

Contract Filtering We could apply our graph extraction process to the whole set of French contracts, however this would result in a single graph, combining heterogeneous activity domains and agent types. Yet, some attributes, for example the weight of social and environmental criteria, directly depend on these domains and types [32]. Instead, we select only a part of the available data to constitute a collection of consistent contracts. For this purpose, we filter them according to five aspects: agent category, activity sector, temporal period, geographic region and size. Regarding the agents, we focus on municipalities, because they are very numerous, and automating their identification is more straightforward than for the other types of public agents. For each municipality present in the dataset, we build a subset of contracts containing not only its own contracts, but also those involving their winners, as well as the other municipalities with which they have obtained contracts. The other four filters allow us to control the size of these subsets of contracts, while retaining a certain homogeneity: we keep only those related to works, covering periods of one year, and involving only suppliers belonging to the same French administrative subdivision.

After this filtering, we obtain a collection of contract subsets containing a total of 25,252 contracts. For each contract, we compute a standard red flag from the literature, in order to model how fraudulent it could be. A contract is red flagged if the number of offers received is exactly 1, which reveals a lack of competition [36].

Graph Extraction For each contract subset obtained after the filtering, we extract a graph G . We consequently build a set of graphs, corresponding to \mathcal{G} in Section 4. In the context of public procurement, due to the complexity of the data, one can extract various types of graphs [14], depending on what the vertices, edges, and their attributes, represent.

We use vertices to model agents, and edges to represent relationships between them, i.e. their joint involvement in at least one contract. Each vertex has an attribute, indicating whether the agent is a buyer or a winner, while each edge has an attribute related to the number of lots contracted between a buyer and a winner. We limit the latter to three levels: 1) exactly one lot; 2) between 2 and 5 lots; and 3) 6 lots or more. This allows us to identify cases where a buyer has many contracts with a single winner, a behavior generally associated with red flags in the literature [12].

We consider that an edge is anomalous if it represents *at least* one red flagged contract, i.e. a contract that received exactly one offer. The label of a graph depends on its total number of anomalous edges: normal if there are fewer than 2, anomalous otherwise. Our graph extraction method produces 389 normal and 330 anomalous graphs. Table 3 shows the main characteristics of the resulting FOPPA dataset, which is publicly available online with our source code⁴.

Graph Class	Average number of vertices (std)	Average number of edges (std)
<i>Anomalous</i>	15.76 (5.56)	17.09 (7.86)
<i>Normal</i>	12.54 (5.41)	12.59 (6.90)

Table 3: Characteristics of the graphs extracted from the FOPPA dataset.

5.2 Results on Public Procurement Data

Comparison With a Tabular Representation In order to study the impact of our graph-based representations, we compare them to a baseline relying on the traditional *tabular* approach. For each contract, we use as predictive features 15 fields available in FOPPA. We select only relevant fields such as the type of procedure, or the presence of a framework agreement. With these features, we aim to predict a binary class, based on the same red flag as before: the number of offers for the contract. Class 0 contains the contracts with more than 1 tender, and Class 1 those with a unique tender. Note that the predictive features are independent from the number of offers.

Like for the graphs, we train an SVM with 10-fold cross-validation, on the same 25,252 contracts obtained after the filtering step. However, the resulting prediction is defined at the *contract* level (one row in the tabular data), whereas

⁴ <https://github.com/CompNet/Pang/releases/tag/v1.0.0>

PANG works at the agent level (one graph in the collection). To compare these results, we need to group the tabular predictions by agent. For this purpose, we proceed as in Section 5.1, by considering any agent with two red flagged contracts or more as anomalous.

Type of data	Anomalous Class	Normal Class
Tabular Data	0.19 (0.01)	0.66 (0.01)
PANG_IndBin	0.95 (0.01)	0.93 (0.02)

Table 4: *F*-Scores (\pm standard deviation) for both classes, obtained with the tabular and graph data.

Table 4 compares the obtained performance with our best graph-based results. The *F*-Scores are averaged over the 10 folds, with standard deviation, for the *Anomalous* and *Normal* classes. For the same contracts and classifier (C-SVM), the graphs allow us to predict fraudulent behaviors much more efficiently than the tabular data, notably for anomalous agents. This clearly confirms the interest of taking advantage of relationships between agents to tackle fraud detection, especially when red flags are missing.

Discrimination Score When applied to our dataset, gSpan returns a total of 15,793 distinct patterns. Figure 5.a shows the distribution of their discrimination score. It is in $[0; 20]$ for most patterns (85%), which can thus be considered as non-discriminative. Figure 5.b shows examples of 2 discriminative patterns, with respective scores of 64 and 91. Both of them include several relations with an intermediary number of lots, which are rather common in large graphs, and more often associated with anomalous graphs.

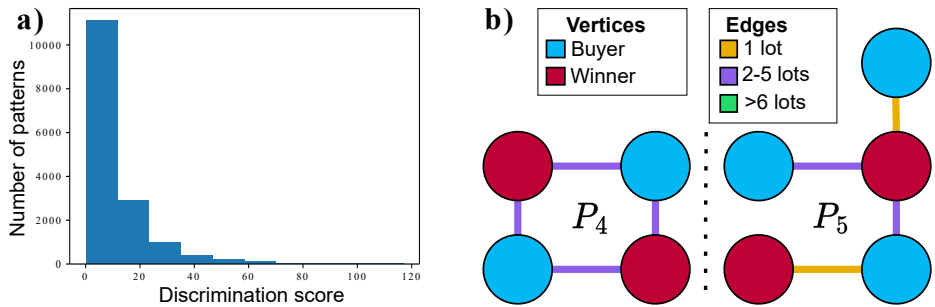


Fig. 5: (a) Distribution of the patterns in function of their discrimination scores (b) Examples of discriminative patterns.

Impact of the Number of Discriminative Patterns We now study how the performance is affected by the number s of patterns in \mathcal{P}_s , i.e. the vector representation size. Table 5 shows how the F -Score changes depending on s , for anomalous and normal graphs. The last row indicates the performance obtained with all the identified patterns ($s = |\mathcal{P}|$). A representation based on only 100 patterns, i.e. less than 1% of the 15,793 patterns, is sufficient to reach the 0.8 bar for both classes. This represents around 90% of the maximal F -Score, obtained with all patterns. Therefore, only a small number of patterns are required to convey the information necessary to tackle the classification task.

Representation Size s	Anomalous Class	Normal Class
10	0.66 (0.05)	0.73 (0.05)
50	0.74 (0.05)	0.77 (0.04)
100	0.81 (0.05)	0.83 (0.04)
150	0.88 (0.03)	0.88 (0.03)
(all) 15,793	0.93 (0.02)	0.93 (0.02)

Table 5: F -Score (\pm standard deviation) depending on parameter s , the size of \mathcal{P}_s .

Impact of the Type of Patterns We also study how the type of pattern influences the constitution of \mathcal{P}_s , and therefore the classification performance. For this purpose, we set $s = 100$, and compare the six representations proposed by PANG, as we did in Section 4.2. Table 6 shows the F -Score obtained with each representation, for both classes.

Representation Type	Anomalous Class	Normal Class
PANG_GenBin	0.81 (0.05)	0.83 (0.04)
PANG_GenOcc	0.73 (0.07)	0.79 (0.05)
PANG_IndBin	0.84 (0.03)	0.85 (0.03)
PANG_IndOcc	0.82 (0.05)	0.84 (0.04)
PANG_CloBin	0.84 (0.04)	0.85 (0.04)
PANG_CloOcc	0.83 (0.05)	0.85 (0.04)

Table 6: F -Score (\pm standard deviation) depending on the pattern type of the representation.

Representations based on induced and closed patterns systematically lead to better results. Yet, a manual examination of \mathcal{P}_s reveals that the discrimination scores of their selected patterns are similar to the general case. The worst selected pattern reaches a score of 67 for general patterns, vs. 61 for induced and 64 for

closed patterns. The difference lies in the *nature* of the selected patterns, which are more diverse than when mining general patterns. For induced and closed patterns, \mathcal{P}_s includes respectively 16 and 13 patterns that do not appear when dealing with general patterns.

Interpretation of Fraudulent Behavior through Pattern Analysis An important advantage of our framework is the identification of the most discriminative patterns, and thus the possibility to leverage human expertise to interpret these patterns and better understand the reasons why an agent is considered fraudulent. For illustration, Figure 6 shows two discriminative patterns, P_4 and P_6 . Pattern P_4 represents a relationship between two winners and two buyers, with more than one contract between them. This type of pattern occurs more frequently in graphs with more contracts, which is typical of anomalous graphs. Pattern P_6 has a winner connected to several buyers, and a single of these edges is green. This can be interpreted as favoritism: a winner works much more with a municipality than with the others.

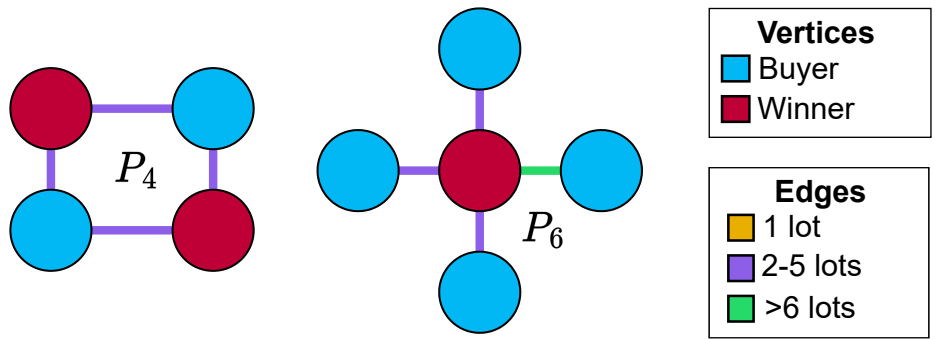


Fig. 6: Examples of discriminative patterns characteristic of class *Anomalous*.

6 Conclusion

In this paper, we propose PANG, a pattern-based generic framework that represents graphs as vectors, by identifying and leveraging their most discriminative subgraphs. We show how PANG, coupled with a standard classifier such as SVM, can detect fraud in public procurement, by applying it to an existing database (FOPPA). Traditional fraud detection approaches typically use tabular data to compute red flags to estimate risk, and fail when these data are incomplete. PANG leverages relational information between economical agents, and our experiments confirm that the use of graphs makes it possible to overcome this issue. They also show that prediction performance can be improved by mining closed or induced patterns, which constitute a set of predictors less redundant

than general patterns. Finally, in this context, a clear advantage of PANG relies on the explainability of these discriminative patterns, which can be interpreted and associated with human behaviors such as favoritism.

References

1. Acosta-Mendoza, N., Gago-Alonso, A., Carrasco-Ochoa, J.A., Francisco Martínez-Trinidad, J., Eladio Medina-Pagola, J.: Improving graph-based image classification by using emerging patterns as attributes. *Engineering Applications of Artificial Intelligence* **50**, 215–225 (2016). <https://doi.org/10.1016/j.engappai.2016.01.030>
2. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery* **29**(3), 626–688 (2014). <https://doi.org/10.1007/s10618-014-0365-y>
3. Carneiro, D., Veloso, P., Ventura, A.: Network analysis for fraud detection in Portuguese public procurement. In: *International Conference on Intelligent Data Engineering and Automated Learning*. Lecture Notes in Computer Science, vol. 12490, pp. 390–401. Springer (2020). https://doi.org/10.1007/978-3-030-62365-4_37
4. Carvalho, R.N., Matsumoto, S., Laskey, K.B., Costa, P.C.G., Ladeira, M., Santos, L.L.: Probabilistic ontology and knowledge fusion for procurement fraud detection in Brazil. In: *International Workshop on Uncertainty Reasoning for the Semantic Web*. Lecture Notes in Computer Science, vol. 7123, pp. 19–40. Springer (2013). https://doi.org/10.1007/978-3-642-35975-0_2
5. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* **2**(3), 1–27 (2011). <https://doi.org/10.1145/1961189.1961199>
6. Cordella, L.P., Foggia, P., Sansone, C., Vento, M.: A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(10), 1367–1372 (2004). <https://doi.org/10.1109/TPAMI.2004.75>
7. CSIRO’s Data61: Stellargraph machine learning library (2018), <https://github.com/stellargraph/stellargraph>
8. Debnath, A.S., Lopez, R.L., Debnath, G., Shusterman, A., Hansch, C.: Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry* **34**(2), 786–797 (1991). <https://doi.org/10.1021/jm00106a046>
9. Dobson, P.D., Doig, A.J.: Distinguishing enzyme structures from non-enzymes without alignments. *Journal of Molecular Biology* **330**(4), 771–783 (2003). [https://doi.org/10.1016/s0022-2836\(03\)00628-4](https://doi.org/10.1016/s0022-2836(03)00628-4)
10. Dou, Y., Shu, K., Xia, C., Yu, P.S., Sun, L.: User preference-aware fake news detection. In: *44th ACM SIGIR International Conference on Research and Development in Information Retrieval*. pp. 2051–2055 (2021). <https://doi.org/10.1145/3404835.3462990>
11. European Union: Tenders Electronic Daily (2023), <https://ted.europa.eu/>
12. Falcón-Cortés, A., Aldana, A., Larralde, H.: Practices of public procurement and the risk of corrupt behavior before and after the government transition in México. *EPJ Data Science* **11**, 19 (2022). <https://doi.org/10.1140/epjds/s13688-022-00329-7>
13. Fazekas, M., Tóth, I.J.: New ways to measure institutionalised grand corruption in public procurement. Technical report, U4 Anti-Corruption Resource Centre (2014), https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2519385

14. Fazekas, M., Tóth, I.J.: From corruption to state capture: A new analytical framework with empirical applications from Hungary. *Political Research Quarterly* **69**(2), 320–334 (2016). <https://doi.org/10.1177/1065912916639137>
15. Ferwerda, J., Deleanu, I., Unger, B.: Corruption in public procurement: finding the right indicators. *European Journal on Criminal Policy and Research* **23**(2), 245–267 (2017). <https://doi.org/10.1007/s10610-016-9312-3>
16. Ferwerda, J., Deleanu, I.S.: Identifying and reducing corruption in public procurement in the EU. Technical report, European Commission (2013), https://ec.europa.eu/anti-fraud/sites/antifraud/files/docs/body/identifying_reducing_corruption_in_public_procurement_en.pdf
17. Fournier-Viger, P., Cheng, C., Chun-Wei Lin, J., Yun, U., Kiran, R.U.: TKG: Efficient mining of top-k frequent subgraphs. In: *International Conference on Big Data Analytics*. Lecture Notes in Computer Science, vol. 11932, pp. 209–226. Springer (2019). https://doi.org/10.1007/978-3-030-37188-3_13
18. Fournier-Viger, P., Lin, J.C.W.: The SPMF open-source data mining library version 2. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Lecture Notes in Computer Science, vol. 9853, pp. 36–40. Springer (2016). https://doi.org/10.1007/978-3-319-46131-1_8
19. Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using NetworkX. In: *7th Python in Science Conference*. pp. 11–15 (2008), https://conference.scipy.org/proceedings/SciPy2008/paper_2/
20. Houbraeken, M., Demeyer, S., Michoel, T., Audenaert, P., Colle, D., Pickavet, M.: The index-based subgraph matching algorithm with general symmetries (ISMAGS): Exploiting symmetry for faster subgraph enumeration. *PLoS ONE* **9**(5), e97896 (2014). <https://doi.org/10.1371/journal.pone.0097896>
21. Hsieh, S.M., Hsu, C.C., Hsu, L.F.: Efficient method to perform isomorphism testing of labeled graphs. In: *International Conference on Computational Science and Its Applications*. Lecture Notes in Computer Science, vol. 3984, pp. 422–431. Springer (2006). https://doi.org/10.1007/11751649_46
22. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraphs in the presence of isomorphism. In: *3rd IEEE International Conference on Data Mining* (2003). <https://doi.org/10.1109/icdm.2003.1250974>
23. Kane, B., Cuissart, B., Crémilleux, B.: Minimal jumping emerging patterns: Computation and practical assessment. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Lecture Notes in Computer Science, vol. 9077, pp. 722–733. Springer (2015). https://doi.org/10.1007/978-3-319-18038-0_56
24. Kim, H., Lee, B.S., Shin, W.Y., Lim, S.: Graph anomaly detection with graph neural networks: Current status and challenges. *IEEE Access* **10**, 111820–111829 (2022). <https://doi.org/10.1109/access.2022.3211306>
25. Kriege, N.M., Giscard, P.L., Wilson, R.: On valid optimal assignment kernels and applications to graph classification. In: *30th International Conference on Neural Information Processing Systems*. pp. 1623–1631 (2016), https://proceedings.neurips.cc/paper_files/paper/2016/hash/0efe32849d230d7f53049ddc4a4b0c60-Abstract.html
26. Li, R., Wang, W.: REAFUM: Representative approximate frequent subgraph mining. In: *SIAM International Conference on Data Mining*. pp. 757–765 (2015). <https://doi.org/10.1137/1.9781611974010.85>
27. Loyola-González, O., Medina-Pérez, M.A., Choo, K.R.: A review of supervised classification based on contrast patterns: Applications, trends, and challenges. *Journal of Grid Computing* **18**(4), 797–845 (2020). <https://doi.org/10.1007/s10723-020-09526-y>

28. Luo, X., Wu, J., Yang, J., Xue, S., Peng, H., Zhou, C., Chen, H., Li, Z., Sheng, Q.Z.: Deep graph level anomaly detection with contrastive learning. *Scientific Reports* **12**(1) (2022). <https://doi.org/10.1038/s41598-022-22086-3>
29. Ma, R., Pang, G., Chen, L., van den Hengel, A.: Deep graph-level anomaly detection by glocal knowledge distillation. In: 15th ACM International Conference on Web Search and Data Mining. ACM (2022). <https://doi.org/10.1145/3488560.3498473>
30. Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q.Z., Xiong, H., Akoglu, L.: A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering* **in press** (2021). <https://doi.org/10.1109/TKDE.2021.3118815>
31. Malik, R., Khan, K.U., Nawaz, W.: Maximal gSpan: Multi-document summarization through frequent subgraph mining. In: 17th International Conference on Ubiquitous Information Management and Communication. pp. 1–7 (2023). <https://doi.org/10.1109/imcom56909.2023.10035618>
32. Maréchal, F., Morand, P.H.: Are social and environmental clauses a tool for favoritism? analysis of French public procurement contracts. *European Journal of Political Economy* **73**, 102140 (2022). <https://doi.org/10.1016/j.ejpoleco.2021.102140>
33. Métivier, J.P., Lepailleur, A., Buzmakov, A., Poezevara, G., Crémilleux, B., Kuznetsov, S.O., Le Goff, J., Napoli, A., Bureau, R., Cuissart, B.: Discovering structural alerts for mutagenicity using stable emerging molecular patterns. *Journal of Chemical Information and Modeling* **55**(5), 925–940 (2015). <https://doi.org/10.1021/ci500611v>
34. Mooney, C.H., Roddick, J.F.: Sequential pattern mining - approaches and algorithms. *ACM Computing Surveys* **45**(2), 1–39 (2013). <https://doi.org/10.1145/2431211.2431218>
35. Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., Jaiswal, S.: graph2vec: Learning distributed representations of graphs. In: 13th International Workshop on Mining and Learning with Graphs. p. 21 (2017), <https://arxiv.org/abs/1707.05005>
36. National Fraud Authority: Red flags for integrity: Giving the green light to open data solutions. Tech. rep., Open Contracting Partnership, Development Gateway (2016), <https://www.open-contracting.org/wp-content/uploads/2016/11/OCP2016-Red-flags-for-integrityshared-1.pdf>
37. Potin, L., Labatut, V., Figueiredo, R., Largeron, C., Morand, P.H.: FOPPA: a database of French Open Public Procurement Award notices. Tech. rep., Avignon Université (2022), <https://hal.archives-ouvertes.fr/hal-03796734>
38. Pourhabibi, T., Ong, K.L., Kam, B.H., Boo, Y.L.: Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems* **133**, 113303 (2020). <https://doi.org/10.1016/j.dss.2020.113303>
39. Rizzo, I.: Efficiency and integrity issues in public procurement performance. *Journal of Public Finance and Public Choice* **31**(1-3), 111–128 (2013). <https://doi.org/10.1332/251569213x15664519748613>
40. Rozemberczki, B., Kiss, O., Sarkar, R.: Karate Club: An api oriented open-source Python framework for unsupervised learning on graphs. In: 29th ACM International Conference on Information and Knowledge Management. pp. 3125–3132 (2020). <https://doi.org/10.1145/3340531.3412757>
41. Shaul, Z., Naaz, S.: cgSpan: Closed graph-based substructure pattern mining. In: IEEE International Conference on Big Data (2021). <https://doi.org/10.1109/BigData52589.2021.9671995>

42. Shervashidze, N., Schweitzer, P., van Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research* **12**(77), 2539–2561 (2011), <http://jmlr.org/papers/v12/shervashidze11a.html>
43. Siglidis, G., Nikolentzos, G., Limnios, S., Giatsidis, C., Skianis, K., Vazirgiannis, M.: GraKeL: A graph kernel library in python. *Journal of Machine Learning Research* **21**(54), 1–5 (2020), <https://www.jmlr.org/papers/v21/18-370.html>
44. Thoma, M., Cheng, H., Gretton, A., Han, J., Kriegel, H.P., Smola, A., Song, L., Yu, P.S., Yan, S., Borgwardt, K.M.: Discriminative frequent subgraph mining with optimality guarantees. *Statistical Analysis and Data Mining* **3**(5), 302–318 (2010). <https://doi.org/10.1002/sam.10084>
45. Toivonen, H., Srinivasan, A., King, R.D., Kramer, S., Helma, C.: Statistical evaluation of the predictive toxicology challenge 2000-2001. *Bioinformatics* **19**(10), 1183–1193 (2003). <https://doi.org/10.1093/bioinformatics/btg130>
46. Wachs, J., Kertész, J.: A network approach to cartel detection in public auction markets. *Scientific Reports* **9**, 10818 (2019). <https://doi.org/10.1038/s41598-019-47198-1>
47. Wale, N., Karypis, G.: Comparison of descriptor spaces for chemical compound retrieval and classification. In: 6th International Conference on Data Mining. pp. 678–689 (2006). <https://doi.org/10.1109/icdm.2006.39>
48. Yan, X., Cheng, H., Han, J., Yu, P.S.: Mining significant graph patterns by leap search. In: ACM SIGMOD International Conference on Management of Data. pp. 433–444 (2008). <https://doi.org/10.1145/1376616.1376662>
49. Yan, X., Han, J.: gSpan: graph-based substructure pattern mining. In: IEEE International Conference on Data Mining. pp. 721–724 (2002). <https://doi.org/10.1109/ICDM.2002.1184038>
50. Yang, Z., Zhang, G., Wu, J., Yang, J.: A comprehensive survey of graph-level learning. arXiv **cs.LG**, 2301.05860 (2023), <https://arxiv.org/abs/2301.05860>
51. Yuan, H., Yu, H., Gui, S., Ji, S.: Explainability in graph neural networks: A taxonomic survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **in press** (2022). <https://doi.org/10.1109/tpami.2022.3204236>
52. Zhang, M., Cui, Z., Neumann, M., Chen, Y.: An end-to-end deep learning architecture for graph classification. In: AAAI Conference on Artificial Intelligence. vol. 32, pp. 4438–4445 (2018). <https://doi.org/10.1609/aaai.v32i1.11782>

A Classifier Comparison

Table 7 complements the results presented in Section 4.2, by showing the performance obtained by a selection of classifiers on the FOPPA dataset, using the PANG_GenBin representation and all available patterns.

B Ethical Implications

Anomaly detection can have ethical implications, for instance if the methods are used to discriminate against certain individuals. In this respect, however, our PANG methodological framework does not present any more risk than the supervised classification methods developed in machine learning.

Classifier	<i>Anomalous Class</i>	<i>Normal Class</i>
C-SVM	0.93 (0.02)	0.93 (0.02)
Random Forest	0.91 (0.02)	0.91 (0.02)
K Neighbors	0.74 (0.04)	0.71 (0.06)
Gradient Boosting	0.87 (0.02)	0.87 (0.04)

Table 7: *F*-Scores for the *Anomalous* and *Normal* classes, according to all the considered classifiers.

Moreover, this work takes place in the framework of a project aiming, among other things, at proposing ways of automatically red flagging contracts and economic agents depending on fraud risk. Therefore, the method that we propose is meant to be used by public authorities to better regulate public procurement and the management of the related open data.

Finally, the data used in this article are publicly shared, and were collected from a public open data repository handled by the European Union. They do not contain any personal information, and cannot be used directly to infer any personal information, as they only describe the economic transactions of companies and public institutions regarding public procurement.