



HAL
open science

Suspicious: a Resilient Semi-Supervised Framework for Graph Fraud Detection

Bastien Giles, Baptiste Jeudy, Christine Largeron, Damien Saboul

► **To cite this version:**

Bastien Giles, Baptiste Jeudy, Christine Largeron, Damien Saboul. Suspicious: a Resilient Semi-Supervised Framework for Graph Fraud Detection. 35th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Nov 2023, Atlanta, GA, United States. pp.212-220, 10.1109/ictai59109.2023.00039 . hal-04461200

HAL Id: hal-04461200

<https://hal.science/hal-04461200v1>

Submitted on 16 Feb 2024

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.

Suspicious: a Resilient Semi-Supervised Framework for Graph Fraud Detection

Bastien Giles

Université Jean Monnet Saint-Etienne, CNRS,
Lab. H. Curien UMR 5516,F-42023,
SAINT-ETIENNE, France
be-ys Research,
ARGONAY, France
0000-0003-3766-978X

Baptiste Jeudy

Université Jean Monnet Saint-Etienne, CNRS,
Lab. H. Curien UMR 5516,F-42023,
SAINT-ETIENNE, France
baptiste.jeudy@univ-st-etienne.fr

Christine Largeron

Université Jean Monnet Saint-Etienne, CNRS,
Lab. H. Curien UMR 5516,F-42023,
SAINT-ETIENNE, France
christine.largeron@univ-st-etienne.fr

Damien Saboul

be-ys Research,
ARGONAY, France
damien.saboul@be-ys-research.com

Abstract—Graph-based fraud detection is an important task in many real-world domains such as insurance, finance, and cybersecurity. Even if existing semi-supervised models have proven to be efficient in identifying anomalous nodes, they assume that a labeled sample of the nodes is available to train the model, without taking into account the real-world problem of the unreliability of such a sample. In practice, the labeling is often done manually and contains many errors. In this paper, we study fraud detection in attributed networks, and we propose a new framework, based on two graph auto-encoders trained following a suspicion mechanism: the first auto-encoder is trained to better reconstruct the normal nodes while the second one, the fraudulent ones. The final classification is done by coupling the result of both auto-encoders. We demonstrate that our approach obtains at least equivalent performances to state of the art methods in the case of a perfectly labeled sample while being more resilient to the introduction of mistakes in this sample.

Index Terms—Anomaly detection, Fraud identification, Graph, Networks, Mislabeling

I. INTRODUCTION

Anomaly and fraud identification is an important problem that has been studied for a long time [1], [3], [6], [11], [25] and, more recently, in graph data [15], [24], [32]. It is used in different fields such as insurance [23], [41], or finance [34]. Depending on the application, data can be represented in various forms requiring appropriate methods. Attributed graphs are one of these representations. While graphs allow the representation of relationships by links between vertices corresponding to entities, attributed graphs provide, in addition, an attribute matrix that contains characteristics or features of the nodes. For example, in the case of a social network, the interactions between the users could be represented by a graph and the profile of each user (age, gender, center of interests, etc.) is stored in an attribute matrix [13]. We consider that, in an attributed graph, fraudulent nodes, like anomalies, do

not follow the same patterns or distributions as the normal elements. However, contrary to anomalies, frauds result from intentional illegitimate behavior or actions. For this reason, they can exhibit distinct and consistent patterns that distinguish them from the normal cases and could even constitute a small-sized class in the graph.

In practice, supervised approaches are often hard to apply in identifying frauds because they require a fully labeled dataset that is difficult and expensive to obtain. Consequently, most experimental studies implement so-called unsupervised methods that do not require any labeling and use the rarity of abnormal elements to guide the design of classification models. While these methods effectively identify outliers far from normal nodes, they frequently fail at discovering real frauds corresponding to illegitimate and intentionally generated elements that an expert can identify [20]. Thus, the so-called semi-supervised methods are also explored because they require only a subset of the dataset to be labeled. However, current semi-supervised methods do not consider the possibility of human error during the labeling of this subset. In practice, the presence of such mislabeled nodes (*i.e.*, normal nodes labeled as frauds and conversely) within the training dataset and the small number of labeled nodes seriously degrade the quality of the classification between normal and fraudulent nodes [31].

To overcome these limits, we propose a new general framework for fraud detection in attributed graphs based on state-of-the-art models. This semi-supervised framework, called Suspicious, produces an efficient anomaly detector resilient to human error. More specifically, Suspicious uses two auto-encoders similar to those introduced in [8]. In [8], an auto-encoder is trained to compress the attributed network into a lower dimensional graph embedding before expanding that representation into an approximation of the original graph. The

motivation is that the auto-encoder will capture the distribution of normal nodes and thus, they will be correctly reconstructed, while out-of-distribution outliers will not. However, this auto-encoder cannot exploit any supervised information about known frauds. This is why our framework adds a second auto-encoder trained to reconstruct fraudulent nodes. This second auto-encoder cannot be used alone since there are very few labeled fraudulent nodes which, in addition, may have labeling errors. For this reason, the results of this second auto-encoder alone are generally poor. Thus, our framework, Suspicious, combines both auto-encoders, each one trained to reconstruct the normal nodes and the fraudulent nodes respectively. To our knowledge, Suspicious is the only graph-based anomaly detector designed to be resilient to the existence of labeling errors in the training sample.

Our contributions are:

- 1) *Suspicious*¹, a general reconstruction-based weakly semi-supervised framework for fraud detection in attributed graphs which presents the advantage of being robust against mislabeling in the training set.
- 2) Extensive experiments confirming that our framework performs as well as the best current methods on perfectly labeled training sets and is resilient to mislabeling errors.

After a presentation of the state of the art in Section 2, we describe Suspicious in Section 3 and its experimental evaluation in Section 4 before concluding.

II. STATE OF THE ART

When considering graph data, an anomalous element can be a subgraph, a link between two nodes, or the node itself. In this article, we focus on node anomalies and, more precisely, frauds which correspond to anomalous elements resulting from intentional illegitimate behaviors or actions². Among the methods designed to identify them, we can first mention the proximity-based approaches [4], [14] which measure the closeness of objects through the structure of the graph and consider “close-by” objects in the graph as likely to belong to the same class (anomaly or normal). Another family of methods, dedicated to networks with a community structure [39] composed of groups of strongly connected nodes, considers anomalous the nodes (or edges) that connect two communities. However, we can remark that the methods belonging to these families only succeed in finding a very specific type of “structural” anomaly. In contrast, a third family of methods, designed for attributed graphs, considers only the attribute matrix and reduces the problem to anomaly detection in tabular data. The literature in this area is quite extensive [1], [7], with various models such as those based on distance, density, clustering of similar elements, depth, and many others. However, these methods ignore all the structural information given by the graph. Recently, new models based on Graph Neural Networks (GNN) have emerged such as Graph Convolutional Network

(GCN) [16], GraphSage [33], Graph Isomorphism Network (GIN) [38], Simple Graph Convolutional network (SGC) [36] or Beta Wavelet Graph Neural Network (BWGNN) [31], especially built for anomaly detection. They have proven to outperform classical methods on most graph-specific tasks, including anomaly detection [20]. Based on the principle of graph embeddings, they produce, thanks to an encoder, a low-dimensional vector representation of the nodes using both sets of information: the attribute matrix and the adjacency matrix. Classical tabular-based anomaly detectors can then be applied to these embeddings. For example, in the semi-supervised model presented in [17], the distance between the vector representation of the nodes in the embedding space and the center of a learned hypersphere is computed and used as an anomaly score. The hypersphere is trained by a GCN on labeled nodes so that it only includes normal nodes. However, the sparsity in available labeled data makes these methods even more sensitive to errors in the training set. Reconstruction-based methods also aim to exploit the graph structure and node attributes to detect anomalies [18], [26]. They use matrix factorization to create an approximation of the original graph, before considering the distance of each node to its reconstruction as an anomaly score. Other current approaches [2], [8], [9] use embeddings as a base for reconstruction. In this case, a decoder is added to reconstruct the original graph and attributes from embeddings before computing an anomaly score from the reconstruction errors. Among these methods, Dominant [8] is probably the most related to our work. It uses an auto-encoder based on GCNs that compute an embedding of the graph and then two distinct decoders that approximate the adjacency and attribute matrices respectively. Finally, poorly reconstructed nodes are supposed to be anomalous. However, the anomalies detected by most of these methods are limited to synthetic anomalies. Synthetic anomalies correspond either to contextual anomalies: a node with an attribute vector different from its neighbors, or structural anomalies: a set of nodes with an atypical structure, generally a high degree clique. Our model Suspicious gets around this weakness by adding semi-supervised learning, allowing the detection various types of anomalies. In addition, it is able to identify the fraudulent nodes in an attributed graph even if the training set contains mislabeled elements. By using two auto-encoders, it addresses the lack of resilience of semi-supervised models to the introduction of errors in this training sample.

III. SUSPICIOUS

After defining the problem to be solved and introducing the notations used in the following sections, this section presents the proposed Suspicious methodological framework for identifying anomalies in an attributed graph in a way that is resilient to the labeling errors of the training sample.

A. Problem

Let $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be an attributed network defined by the set of nodes $\mathcal{V} = \{v_1, \dots, v_n\}$, the set of edges \mathcal{E} represented by a symmetric adjacency matrix \mathbf{A} where $a_{k,j} = 1$ if there is

¹Code and datasets available <https://src.koda.cnrs.fr/labhc/code4publications/2023-ICTAI-suspicious>

²In the following, both terms will nevertheless be used to designate the elements to be identified

an edge between nodes k and j and $a_{k,j} = 0$ otherwise, and the attribute matrix $\mathbf{X} \in \mathbb{R}^{(n \times d)}$ which i th row \mathbf{x}_i represents the attribute vector of v_i . In the same way, \mathbf{a}_i denotes the i th row of \mathbf{A} . It is assumed that there is a labeled subset of nodes, $\mathcal{V}_l \subset \mathcal{V}$. It is itself composed of two disjoint subsets, \mathcal{V}_s and \mathcal{V}_n , containing nodes identified, respectively, as fraudulent and normal, possibly with mislabeling errors: $\mathcal{V}_l = \mathcal{V}_s \cup \mathcal{V}_n$ and $\mathcal{V}_s \cap \mathcal{V}_n = \emptyset$. This mislabeling, relatively frequent in practice, even when the labeling is done by experts, leads to anomalies being present in \mathcal{V}_n and normal nodes in \mathcal{V}_s . We denote $\mathcal{V}_a \subset \mathcal{V}$ the set of the nodes that are truly anomalous, *i.e.*, the ground truth that is not available in real applications.

The problem we seek to solve can be expressed as follows: Given the attributed network $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ and the two labeled subsets \mathcal{V}_s and \mathcal{V}_n , the goal is to compute an anomaly score for the unlabeled nodes such that the true fraudulent nodes have a higher score than the true normal ones.

B. Model

1) *Principle Behind Suspicious*: To solve this task, Suspicious uses two graph auto-encoders, *Susp* and *Norm*, as shown in Figure 2. The auto-encoder *Norm* tries to reconstruct G such that the normal nodes are better reconstructed than the fraudulent ones, while *Susp* does the opposite and tries to reconstruct G in a way that anomalies are better reconstructed than normal nodes. For each unlabeled node, we then obtain a reconstruction error (*i.e.*, score) from *Norm* and another one from *Susp* and, finally, these errors are combined in a ranking score. The formal definition of this score is given in the following section, but, informally, these two scores categorize the nodes as follows:

- A low reconstruction error by *Norm* and a high one by *Susp*: both models agree that this is a normal node which should result in a final score among the lowest.
- A high reconstruction error by *Norm* and a low one in *Susp*: both models agree that the node is fraudulent. This means that the node corresponds to a relevant fraud and, therefore, it should result in a final score among the highest.
- In other cases, the combination of the two scores allows our model to better identify the anomalies belonging to \mathcal{V}_s than each auto-encoder individually, as confirmed by the ablation study presented in Section IV-E.

2) *Architecture of Suspicious*: The architecture of both auto-encoders used in Suspicious is illustrated in Figure 1 in which the colors correspond to the types of nodes: green for normal nodes, red for anomalies, orange for poorly reconstructed nodes, and blue for well-reconstructed nodes. The encoder (ENC in Equation 1) is a GCN [16] that computes an embedding \mathbf{Z} of the nodes in G . Then the attribute decoder DEC (another GCN) and the adjacency matrix decoder are used to recreate, from \mathbf{Z} , the approximations $\hat{\mathbf{X}}$ and $\hat{\mathbf{A}}$ of \mathbf{X} and \mathbf{A} , respectively:

$$\mathbf{Z} = \text{ENC}(\mathbf{A}, \mathbf{X}, \Theta_1); \hat{\mathbf{X}} = \text{DEC}(\mathbf{A}, \mathbf{Z}, \Theta_2); \hat{\mathbf{A}} = (\mathbf{Z}^t \mathbf{Z}). \quad (1)$$

Both auto-encoders, *Susp* and *Norm*, use this architecture with different learnable parameters sets: Θ_1^s and Θ_2^s for *Susp* and Θ_1^n and Θ_2^n for *Norm*.

We can notice that each auto-encoder has the same architecture as Dominant [8], but our model uses two of them and does not use the same loss function as Dominant, as explained in the next section. Furthermore, our framework is not limited to GCNs and can be used with other graph embedding models: we also implemented versions³ of our framework in which ENC and DEC are GraphSAGE [12] or SGC [36] instead of the GCN.

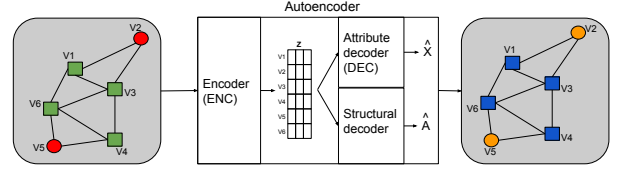


Fig. 1. Architecture of an auto-encoder. Green nodes are normal, red nodes are anomalies, orange nodes are poorly reconstructed, and blue nodes are well-reconstructed.

3) *Calculation of Reconstruction Errors*: For each node v_i and each auto-encoder AE, which can be either *Susp* or *Norm*, we define the node reconstruction error as:

$$\text{err}_{AE}(v_i) = (1 - \alpha) \|\mathbf{a}_i - \hat{\mathbf{a}}_i\|_F^2 + \alpha \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_F^2. \quad (2)$$

where α is a parameter allowing to weight the importance given to each type of reconstruction error, on the structure and on the attributes.

During training, Suspicious tries to minimize the reconstruction error of each auto-encoder on one of the samples (\mathcal{V}_s or \mathcal{V}_n) while maximizing the error on the other. This leads to the definition of two loss functions, one for *Norm* and one for *Susp*:

$$L_{Norm} = \frac{\sum_{v_i \in \mathcal{V}_n} \text{err}_{Norm}(v_i)}{\sum_{v_i \in \mathcal{V}_s} \text{err}_{Norm}(v_i)}, \quad L_{Susp} = \frac{\sum_{v_i \in \mathcal{V}_s} \text{err}_{Susp}(v_i)}{\sum_{v_i \in \mathcal{V}_n} \text{err}_{Susp}(v_i)}. \quad (3)$$

Thus, both auto-encoders *Susp* and *Norm* use \mathcal{V}_n and \mathcal{V}_s but in a different way: nodes that are similar to the majority of the nodes in their respective sample get lower scores, while nodes that are less represented in the sample get higher scores. The loss functions are minimized during training using gradient descent, as explained in the next section. As previously stated, the main difference between Suspicious and Dominant lies not only in its architecture, Suspicious is based on two auto-encoders, as shown in Figure 2, while Dominant has only one, but also in the calculation of the loss function. While Dominant tries to reconstruct the whole graph with minimal error, in our model, each encoder tries to minimize its own reconstruction error.

³Codes available on <https://src.koda.cnrs.fr/labhc/code4publications/2023-ICTAI-suspicious>

4) *Final Score and Decision Criterion*: The reconstruction errors of the nodes for each auto-encoder are normalized in $[0, 1]$, where AE is either *Susp* or *Norm*:

$$En_{AE}(v_i) = \frac{err_{AE}(v_i) - \min_{v_j \in \mathcal{V}}(err_{AE}(v_j))}{\max_{v_j \in \mathcal{V}}(err_{AE}(v_j)) - \min_{v_j \in \mathcal{V}}(err_{AE}(v_j))} \quad (4)$$

The final ranking score is then:

$$Ranking_{score}(v_i) = \frac{En_{Norm}(v_i)}{En_{Susp}(v_i)}. \quad (5)$$

Therefore, high scores in En_{Norm} with low scores in En_{Susp} produce a high $Ranking_{score}$ corresponding to nodes classified as anomalous, which is in line with the principle behind Suspicious, described previously.

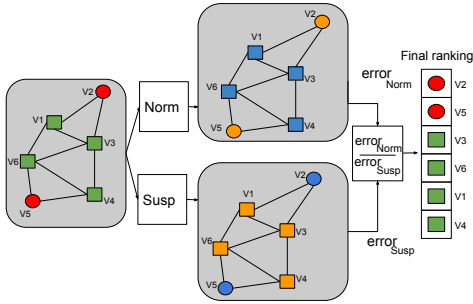


Fig. 2. Architecture of Suspicious. *Norm* and *Susp* are auto-encoders similar to those shown in Figure 1. Green nodes are normal, red nodes are anomalous, orange nodes are poorly reconstructed and blue nodes are well-reconstructed.

IV. EXPERIMENTS

This section details the experimental evaluation of our framework, Suspicious, and its comparison with state of the art methods. It also includes an ablation study showing the interest of combining the two auto-encoders, *Norm* and *Susp*, as well as an analysis of the impact of labeling errors in the training sample, confirming the resilience of our method.

A. Datasets

The experiments are conducted on six real-world datasets:

- Cora and PubMed are popular public network datasets [30]. In these graphs, each node is a scientific publication, and the edge represents the citation of another publication. The attributes correspond to the content of the publications represented as a bag-of-words vector.
- In Photo and Computers, each node is a product, and an edge exists if two products are often purchased together. The attributes are also bag-of-words vectors.
- Reddit is a social media platform. Our dataset is an extracted graph consisting of both subreddits and user nodes, each with bag-of-words attribute vectors extracted from their respective posts. The user label “Banned” is used as our anomaly label. Since banned users are users that broke the rules of Reddit, they can be considered frauds.

TABLE I
CHARACTERISTICS OF THE DATASETS.

Dataset	Nodes	Edges	Attributes	Anomalies		Anomaly Rate	
				dataR	dataS	dataR	dataS
Cora	2708	5278	1433	180	280	0.066	0.1
PubMed	19717	44338	500	4141	1971	0.208	0.1
Photo	7487	119043	745	331	748	0.043	0.1
Computers	13381	245778	767	291	1338	0.021	0.1
Books	1418	3695	21		28		0.020
Reddit	10984	168016	64		366		0.033

- Books is also an Amazon co-purchase network, like Photo and Computers, but in this dataset, the labels are obtained through an “amazon fail” tag which indicates that the product was massively flagged by users. In Reddit and Books, the label information (anomalous/normal) has been obtained through human expertise and requires no additional modification. However, as it is the result of human decisions, it can contain mislabeling. Thus, this ground truth is not fully reliable for an experimental evaluation.

The evaluation of Suspicious requires datasets including anomalous elements. We consider three types of anomalies. In addition to organic anomalies provided by humans, we follow the experimental protocols introduced in Kumagai et al. [17] and Liu et al. [20] to define, respectively, rare class anomalies and synthetic anomalies for the first four previous datasets (Cora, PubMed, Photo and Computer). Thus, two versions have been generated for each dataset *data* denoted respectively by *dataR* and *dataS*. The characteristics of our datasets are summarized in Table I: number of nodes ($|\mathcal{V}|$), edges ($|\mathcal{E}|$), attributes (d), anomalies ($|\mathcal{V}_a|$) and anomaly rate ($\frac{|\mathcal{V}_a|}{|\mathcal{V}|}$).

Rare class anomalies: Each of the datasets, Cora, PubMed, Photo and Computer, contains several classes of nodes. Following the protocol of [17], [35] rare class anomalies can be defined by relabeling the elements of the smallest class as anomalies, while all the nodes of the other classes are considered normal. In this way, a dataset named *dataR*, suitable for binary classification with class imbalance, is created.

Synthetic anomalies: The same datasets, Cora, PubMed, Photo, and Computers, are also modified to obtain contextual and structural anomalies following the methodology proposed in [20]. Specifically, contextual anomalies are created by replacing the attribute vector of a node by the most different node attribute vector out of a range of randomly selected nodes. Structural anomalies are created by adding edges to the graph to create k cliques of k nodes with $k = \sqrt{0.05 \times |\mathcal{V}|}$. The ratio of created anomalous nodes is 5% for each type of anomaly, resulting in a total of 10% of anomalous nodes of both types, contextual and structural, as indicated in Table I (Anomaly rate) and resulting in the dataset called *dataS*.

Organic anomalies: Books and Reddit contain organic anomalies [20]: the label (anomalous/normal) has been obtained through human expertise and requires no additional modification.

1) *Protocol*: In a realistic fraud detection application, the number of available labeled nodes would be low and the labels could be erroneous. To reflect these hypotheses, we propose the following protocol: A training sample \mathcal{V}_l is randomly sampled from \mathcal{V} according to a rate R equal to 10% by taking $R \times |\mathcal{V}_a|$ anomalous nodes and $R \times |\mathcal{V} \setminus \mathcal{V}_a|$ normal nodes. The corresponding test set is equal to $\mathcal{V} \setminus \mathcal{V}_l$. To evaluate the resilience of the methods, we then introduce errors in labels: the label of some nodes in \mathcal{V}_l is swapped. The proportion of normal nodes wrongly labeled fraudulent is denoted Mis_s , whereas the proportion of anomalous nodes labeled normal is Mis_n :

$$Mis_n = \frac{|\mathcal{V}_a \cap \mathcal{V}_n|}{|\mathcal{V}_n|}, \text{ and } Mis_s = \frac{|\mathcal{V}_s \setminus \mathcal{V}_a|}{|\mathcal{V}_s|}. \quad (6)$$

Due to the application, the data is imbalanced: the number of normal nodes $|\mathcal{V}_n|$ is much larger than the number of frauds $|\mathcal{V}_a|$. Thus, Mis_n (which is upper bounded by $|\mathcal{V}_a|/|\mathcal{V}_n|$) is always small and can be neglected. Therefore, in the experiments, it is taken as equal to 0, which also decreases the number of parameters to test. The proportion Mis_s takes values varying between 0 and 0.5 in the experiments. Finally, 10 training/test samples are produced for each dataset and each mislabeling rate Mis_s . All reported results in the following sections are averages (with standard deviation SD) over these 10 samples.

B. Settings for Suspicious and Baselines

Our methodological framework, Suspicious, is evaluated in three settings depending on the version of the auto-encoder (AE) used: ENC and DEC in Equation 1 can be: GCN [16], GraphSage [33] or SGC [36]. This leads to three variants respectively denoted Ours-GCN, Ours-Sage, and Ours-SGC in the following sections. They are compared with the following unsupervised (u) and semi-supervised (s) state-of-the-art methods:

- **Kumagai et al.** (s): a semi-supervised graph embedding method that uses the AUC regularizer as part of the loss to minimize the volume of a hypersphere that encompasses labeled normal nodes [17]. As Kumagai has been shown to outperform OCGNN [35], OSVM [29], Doc-N [28], Deep Walk [27], and ImVerde [37] on these datasets, we do not consider these methods as additional baselines since by outperforming Kumagai et al., Suspicious will also outperform them.
- **GNN classifiers** (s): Three graph neural networks, GCN [16], GIN [38] and GAT [33], are used as node classifiers.
- **BWGNN** (s): BWGNN [31] is a GNN designed for anomaly detection.
- **Dominant** (u): an unsupervised method of graph reconstruction based on an auto-encoder which calculates anomaly scores as the sum of the reconstruction errors made on the attributes and the graph structure [8].
- **AnomalyDAE** (u): an improved version of Dominant [9]. It has been empirically demonstrated in [20] that AnomalyDAE consistently outperforms LOF [5] and IF

[19]. For this reason, we do not integrate them in our evaluation.

- **Anomalous** (u): a method that uses CUR decomposition and residual analysis for computing a final score defined by the norm of its reconstruction residual [26].
- **CoLA** (u): a method that produces an embedding of the graph through a GNN and then uses contrastive learning to compute an anomaly score [21]. Moreover, as CoLA is based on the same approach as [42] and provides equivalent results, this last one is not included.

C. Evaluation Parameters and Metrics

An implementation of the method from Kumagai⁴ was used with the parameters published in the paper: an embedding of dimension 4 for Books since it has fewer attributes, and 32 for the other datasets, a maximum of 500 repetitions with an early stopping mechanism, and the center C defined as the average of the embeddings of the nodes labeled as normal after the first layer of the model.

For Dominant, AnomalyDAE, CoLA and Anomalous, the pygod implementation from BOND [20] was used with the publication settings, a two-layer GCN as an encoder and a one-layer GCN as an attribute decoder, an embedding dimension of 32, 100 repetitions, and $\alpha = 0.5$ for AnomalyDAE and Dominant. For Anomalous, the gradient descent is used instead of closed-form optimization provided in the official implementation due to fairness and efficiency concerns. For the GNN classifiers, the implementation benchmark [22] was used and modified to include mislabeling errors and a training set with the same size as ours. Our framework was implemented using parts of the Dominant implementation and models from PyTorch Geometric [10]; the code and datasets are publicly available⁵. The implementations of Suspicious use the same parameters as our baselines: an embedding dimension of 4 for Books and 32 for the others, a dropout rate of 0.5 for both auto-encoders, 100 repetitions, and a learning rate of 0.005. As in the literature in the field, the results are evaluated with AUC (Area Under the Roc curve) scores computed on the unlabeled nodes: for each dataset and each error rate $Mis_s \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$, we randomly select 10 train samples \mathcal{V}_l and report the average AUC, with standard deviation, computed on the corresponding 10 test samples $\mathcal{V} \setminus \mathcal{V}_l$.

D. Experimental Results

The results obtained by the different methods for the datasets described in Table I are presented in Tables II-III.

Table II shows the results obtained for the rare class datasets in the case of a perfectly labeled training sample *i.e.* when Mis_s equals 0. Our framework obtains the best results across all datasets. However, most semi-supervised methods accurately detect the anomalies and obtain relatively equivalent results to ours, while the unsupervised methods designed for

⁴https://github.com/tuananh0305/GCN_ANOMALY_DETECTION

⁵<https://src.koda.cnrs.fr/labhc/code4publications/2023-ICTAI-suspicious>

TABLE II
AVERAGE AUC \pm SD(%) ON RARE CLASS DATASETS.

	PhotoR	ComputersR	CoraR	PubMedR
Ours-GCN(s)	93.9 \pm 0.2	99.6 \pm 0.1	98.1 \pm 0.5	95.2 \pm 0.2
Ours-Sage(s)	97.1 \pm 0.9	99.5 \pm 0.1	95.6 \pm 0.9	95.3 \pm 0.2
Ours-SGC(s)	97.3 \pm 0.6	99.6 \pm 0.2	95.7 \pm 1.4	95.5 \pm 0.3
BWGNN(s)	92.1 \pm 4.6	72.5 \pm 10.5	48.4 \pm 12.4	88.8 \pm 0.3
GIN(s)	68.0 \pm 18.9	93.1 \pm 1.2	88.3 \pm 0.1	89.0 \pm 0.1
GAT(s)	96.8 \pm 0.2	98.4 \pm 0.2	97.0 \pm 0.5	91.7 \pm 0.4
GCN(s)	95.6 \pm 0.2	98.5 \pm 0.0	98.0 \pm 0.0	92.5 \pm 0.0
Kumagai(s)	94.4 \pm 1.8	97.2 \pm 5.5	94.5 \pm 1.8	94.3 \pm 0.3
AnomalyDAE(u)	51.8 \pm 0.1	56.4 \pm 0.1	47.9 \pm 0.4	61.8 \pm 0.2
Dominant(u)	51.5 \pm 0.0	46.3 \pm 0.0	48.7 \pm 0.4	51.0 \pm 0.0
Anomalous(u)	54.1 \pm 6.8	49.1 \pm 3.4	53.0 \pm 1.2	58.9 \pm 1.9
CoLA(u)	55.8 \pm 3.1	66.2 \pm 1.2	38.3 \pm 7.3	39.2 \pm 3.3

synthetic anomalies completely fail to detect the rare class anomalies.

The results obtained on the synthetic class datasets with perfectly labeled training samples ($Mis_s = 0$) are summarized in Table III. AnomalyDAE gets the best results on PhotoS, and only our framework, Suspicious, reaches equivalent results. Dominant provides the best result on CoraS while our framework arrives second, before AnomalyDAE, with significantly better performances than all the other methods. Finally, Ours-GCN obtains the best results on PubmedS and ComputerS. We can observe that semi-supervised methods struggle to reach performances equivalent to those of the unsupervised methods, specifically designed for these type of anomalies. However, Suspicious, which does not belong to this family of methods, still manages to achieve performances equivalent to the best-performing method, AnomalyDAE.

Finally, the results obtained on the organic datasets are also presented in Table III. All of our methods obtain the best results on both Books and Reddit. While the GCN reaches equivalent performances on the Reddit dataset, all the other methods fail to find the anomalies in Books. The globally poor results on these datasets can be explained by the nature of those anomalies, as they are the result of human decisions. Indeed the ground truth itself can contain mislabeling [40] and, the labels can be inconsistent due to the diversity of the annotators. However, since our method is able to identify a portion of those anomalies it shows to be more appropriate to reproduce human reasoning than the other state-of-the-art methods, supervised or not. Nevertheless, we exclude those datasets from the complementary studies related to the impact of the parameter α and mislabeling rate Mis_s , since it is not pertinent to conduct them on datasets with poor results.

E. Ablation Study

The aim of this ablation study is to check the interest of combining both auto-encoders *Norm* and *Susp* to identify the fraudulent elements. To that aim, we calculate a gain defined as the difference of performances, in terms of AUC score, obtained with Suspicious versus only one auto-encoder *AE* that can be *Norm* or *Susp* (Equation 7). To compute the

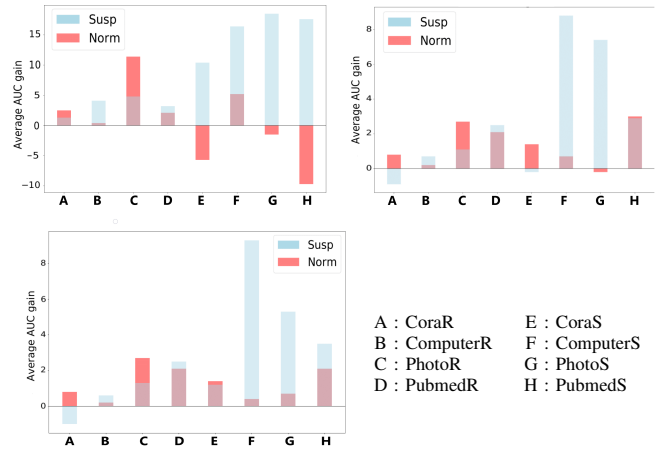


Fig. 3. Average gain using Suspicious versus using only *Norm* (red) or *Susp* (blue) on the different datasets for the settings Ours-GCN (top left), Ours-Sage (top right) and Ours-SGC(bottom).

performances of an AE, we calculate its AUC score using its reconstruction error err_{AE} .

$$Gain = AUC(Suspicious) - AUC(AE) \quad (7)$$

A positive difference corresponds to a gain and confirms the interest of using two auto-encoders instead of only one. Figure 3 shows these differences between the average AUC score calculated with *Norm* (in red) and *Susp* (in blue) on the different datasets for the three settings (Ours-GCN top left, Ours-Sage top right and Ours-SGC bottom).

With Ours-GCN, the use of both auto-encoders improves the detection of rare class anomalies, comparatively to *Susp* and *Norm*, but it degrades the performances of *Norm* for synthetic anomalies on PubMedS, CoraS and, to a lesser extent, on PhotoS. This can be explained by the GCN’s tendency to discard node information that is too different from its neighbors, creating a bias that makes it unable to reconstruct contextual anomalies that are part of synthetic anomalies. By incorporating two auto-encoders, this bias can be corrected,

TABLE III
AVERAGE AUC \pm SD(%) ON SYNTHETIC AND ORGANIC DATASETS.

	PhotoS	ComputersS	CoraS	PubmedS	Reddit	Books
Ours-GCN(s)	76.5 \pm 2.3	80.1 \pm 4.0	78.85 \pm 3.7	84.3 \pm 2.6	64.3 \pm 1.7	58.0 \pm 8.0
Ours-Sage(s)	72.3 \pm 1.0	73.7 \pm 0.4	75.8 \pm 2.5	71.1 \pm 1.0	63.5 \pm 2.8	60.9 \pm 5.1
Ours-SGC(s)	70.7 \pm 1.8	71.8 \pm 1.1	76.4 \pm 2.7	69.1 \pm 0.6	62.5 \pm 4.8	58.1 \pm 7.3
BWGNN(s)	55.3 \pm 8.3	64.0 \pm 1.9	66.3 \pm 1.2	69.5 \pm 0.5	55.8 \pm 2.1	54.2 \pm 3.6
GIN(s)	51.2 \pm 3.9	59.1 \pm 3.6	32.3 \pm 0.0	70.9 \pm 6.4	48.9 \pm 7.3	50.0 \pm 0.0
GAT(s)	53.6 \pm 0.8	70.3 \pm 3.3	66.7 \pm 0.9	72.0 \pm 4.5	59.7 \pm 3.7	50.0 \pm 0.0
GCN(s)	57.0 \pm 0.3	65.3 \pm 0.3	40.0 \pm 0.2	73.9 \pm 0.7	62.0 \pm 0.1	50.0 \pm 0.0
Kumagai(s)	50.2 \pm 2.3	53.2 \pm 1.9	67.9 \pm 2.8	57.0 \pm 1.3	52.7 \pm 0.7	43.4 \pm 1.8
AnomalyDAE(u)	77.5 \pm 0.0	75.5 \pm 0.0	77.5 \pm 0.0	75.7 \pm 0.1	48.6 \pm 3.7	55.2 \pm 6.7
Dominant(u)	63.6 \pm 0.0	65.3 \pm 0.0	83.9 \pm 0.0	81.2 \pm 0.0	56.1 \pm 0.0	38.9 \pm 1.2
Anomalous(u)	48.5 \pm 2.7	49.3 \pm 0.08	33.6 \pm 0.9	37.0 \pm 0.02	53.0 \pm 0.0	47.3 \pm 3.1
CoLA(u)	58.5 \pm 1.6	56.6 \pm 0.08	67.0 \pm 1.8	73.6 \pm 2.3	53.0 \pm 1.3	50.0 \pm 0.0

allowing for better detection of rare class anomalies at the expense of reduced performance for contextual anomalies. It should be noted, however, that this results in an efficient and versatile model with good performance in both scenarios. Concerning Ours-Sage and Ours-SGC, the results show that using both auto-encoders provides almost always a consistent advantage, and when it is not the case, for CoraR and PhotoS, the loss remains low while the gain can be very significant, such as for ComputerS. In conclusion, with an improvement in 42 of 48 cases (Figure 3), this experiment confirms the interest of combining both auto-encoders as in Suspicious. This allows all our models to gain versatility and detect both kinds of anomalies on every dataset.

F. Impact of the Mislabeling Errors

This set of experiments aims to evaluate the resilience of Suspicious to mislabeling errors, frequent in practice due to the difficulty for experts to identify frauds. It consists of studying the variation of the AUC score in function of Mis_s , defined in Equation 6. The results obtained on rare anomaly datasets are presented in Figure 4, while the results obtained on the synthetic anomaly datasets are presented in Figure 5. Since unsupervised methods cannot be affected by labeling errors, their results are included only for information. The Kumagai method has been excluded from this study due to its extremely long execution time as well as CoLA due to its low performances. In Figure 4, we can observe that our models remain among the best-performing methods, no matter the value of Mis_s on rare type anomaly datasets. In Figure 5, we notice that, except for Ours-GCN, our framework remains stable no matter the value of Mis_s . It is important to note that Ours-GCN only seems to be affected by Mis_s when it outperforms the other models and never truly falls below their already satisfying results. These figures show that our model is not impacted by errors occurring in the subset \mathcal{V}_s of \mathcal{V}_l used for its training as long as the labeling error rate remains below 40%, and even for higher rates, it outperforms most semi-supervised methods.

G. Impact of a Varying R

These experiments aim to understand the impact of R , the size of the labeled sample used for training, on the

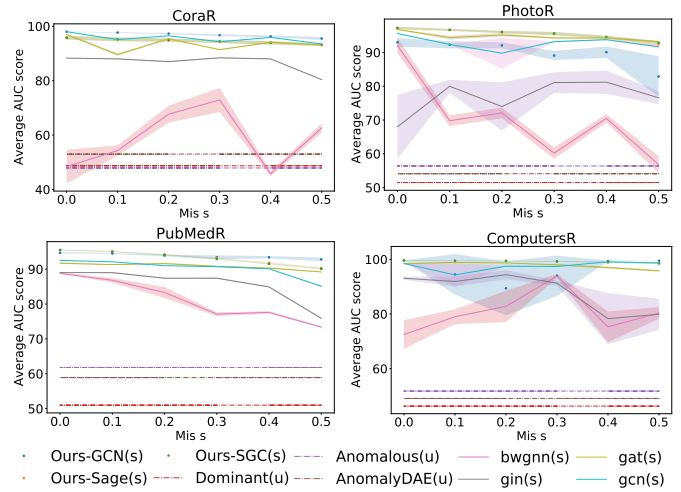


Fig. 4. Evolution of average AUC score in function of Mis_s on rare anomaly datasets.

performances of our framework. Figure 6 shows the variation in the average AUC-score, with standard deviation, in function of R . A lower R generally results in a lower performance but, most datasets reach a competitive average AUC-score with a fairly low value for R . However a higher value is needed to stabilize the standard deviation. Thus, it seems that R mainly impacts the stability of the results but, it should be noted that a value of 3% for R is sufficient to make our framework one of the best-performing method on all datasets.

H. Impact of a Varying α

Finally, we also studied the impact of α on the results. This parameter allows the weighting of the importance given to each type of reconstruction error done either on the structure or on the attributes. Figure 7 presents the average AUC score of Ours-GCN for all non-organic datasets in function of α . For rare class anomalies (DataR), α has a very low impact unless it is equal to 0 or 1. For synthetic anomalies (DataS), there are lower AUC and more variations both in value and in SD (especially ComputersS and PhotoS). All in all, a value around 0.5 generally achieves good results and low SD. The stability of the results on DataR is another advantage of Suspicious over

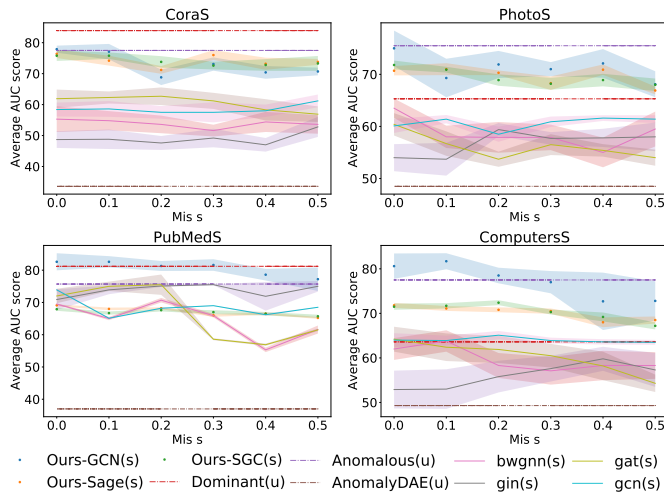


Fig. 5. Evolution of average AUC score in function of Mis_s on synthetic anomaly datasets.

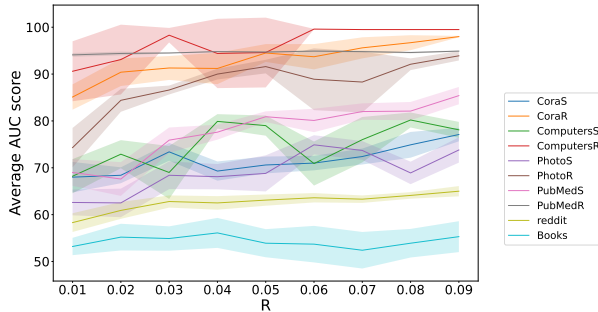


Fig. 6. Evolution of average AUC score of Ours-GCN in function of R .

other methods, as this parameter α has an important impact on the final results of Dominant [8] and AnomalyDAE [9].

V. CONCLUSION

We proposed a semi-supervised general framework for fraud detection in attributed graphs, compatible with all types of GNN, that is versatile while still being resilient to human error in the labeling of the training set. When there is no mislabeling, our model obtains better or equivalent results compared to the current state of the art on all types of anomalies, whereas the other methods are mainly efficient for a certain type of anomaly. Moreover, our models still provide good results with a small sized training set even if it contains labeling errors. As a next step, we plan to add a module to provide an explanation of the final score that can be understood by the experts who provided the original information.

REFERENCES

[1] Aggarwal, C.C.: Outlier Analysis. Springer Int. Publishing (2017)
 [2] Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: Semi-supervised anomaly detection via adversarial training. In: Computer Vision – ACCV 2018. pp. 622–637 (2019)
 [3] Akoglu, L.: Anomaly mining - past, present and future. In: IJCAI. pp. 4932–4936 (2021)

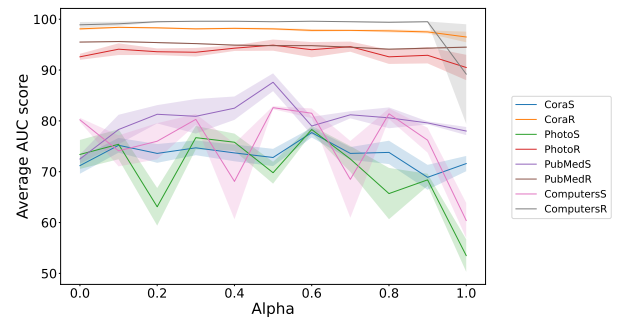


Fig. 7. Evolution of average AUC score of Ours-GCN in function of α .

[4] Antonellis, I., Molina, H.G., Chang, C.C.: Simrank++: query rewriting through link analysis of the click graph. Proc. of the VLDB Endowment **1**, 408–421 (2008)
 [5] Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: ACM. pp. 93–104 (2000)
 [6] Chalapathy, R., Menon, A.K., Chawla, S.: Anomaly detection using one-class neural networks. ArXiv **1802.06360** (2018)
 [7] Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Comput. Surv. **41**(3), 1–58 (2009)
 [8] Ding, K., Li, J., Bhanushali, R., Liu, H.: Deep anomaly detection on attributed networks. In: SIAM ICDM. pp. 594–602 (2019)
 [9] Fan, H., Zhang, F., Li, Z.: Anomalydae: Dual Autoencoder for Anomaly Detection on Attributed Networks. In: ICASSP. pp. 5685–5689 (2020)
 [10] Fey, M., Lenssen, J.E.: Fast graph representation learning with pytorch geometric. ICLR (Workshop) (2019)
 [11] Grubbs, F.E.: Procedures for detecting outlying observations in samples. Technometrics **11**(1), 1–21 (1969)
 [12] Hamilton, W., Ying, Z., Leskovec, J.: Inductive Representation Learning on Large Graphs. In: NeurIPS. p. 1025–1035 (2017)
 [13] Interdonato, R., Atzmueller, M., Gaito, S., Kanawati, R., Langeron, C., Sala, A.: Feature-rich networks: going beyond complex network topologies. Appl. Netw. Sci. pp. 4:1–4:13 (2019)
 [14] Jeh, G., Widom, J.: Simrank: A measure of structural-context similarity. In: SIGKDD. p. 538–543 (2002)
 [15] 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)
 [16] Kipf, T.N., Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. In: ICLR (2017)
 [17] Kumagai, A., Iwata, T., Fujiwara, Y.: Semi-supervised Anomaly Detection on Attributed Graphs. In: IJCNN. pp. 1–8 (2021)
 [18] Li, J., Dani, H., Hu, X., Liu, H.: Radar: Residual Analysis for Anomaly Detection in Attributed Networks. In: IJCAI. pp. 2152–2158 (2017)
 [19] Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining. pp. 413–422 (2008)
 [20] Liu, K., Dou, Y., Zhao, Y., Ding, X., Hu, X., Zhang, R., Ding, K., Chen, C., Peng, H., Shu, K., Sun, L., Li, J., Chen, G.H., Jia, Z., Yu, P.S.: Benchmarking node outlier detection on graphs. In: NeurIPS (2022)
 [21] Liu, Y., Li, Z., Pan, S., Gong, C., Zhou, C., Karypis, G.: Anomaly detection on attributed networks via contrastive self-supervised learning. IEEE TNNLS **33**(6), 2378–2392 (2022)
 [22] Liu, Z., Cao, C., Sun, J.: Mul-gad: a semi-supervised graph anomaly detection framework via aggregating multi-view information. arXiv **2212.05478** (2022)
 [23] Lu, Y.J., Li, C.T.: GCAN: Graph-aware co-attention networks for explainable fake news detection on social media. In: ACL. pp. 505–514 (2020)
 [24] 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 TKDE (2021)
 [25] Pang, G., Shen, C., Cao, L., Hengel, A.v.d.: Deep Learning for Anomaly Detection: A Review. ACM Computing Surveys p. 1–38 (2021)
 [26] Peng, Z., Luo, M., Li, J., Liu, H., Zheng, Q.: ANOMALOUS: A Joint Modeling Approach for Anomaly Detection on Attributed Networks. In: IJCAI. pp. 3513–3519 (2018)

- [27] Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: SIGKDD. p. 701–710 (2014)
- [28] Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: ICML. vol. 80, pp. 4393–4402 (Jul 2018)
- [29] Schölkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. *Neural Comput.* **13**(7), 1443–1471 (2001)
- [30] Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classification in network data articles. *AI Magazine* **29**, 93–106 (09 2008)
- [31] Tang, J., Li, J., Gao, Z., Li, J.: Rethinking graph neural networks for anomaly detection. In: ICML. vol. 162, pp. 21076–21089 (2022)
- [32] Vaudaine, R., Jeudy, B., Langeron, C.: Detection of contextual anomalies in attributed graphs. In: IDA. pp. 338–349 (2021)
- [33] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph Attention Networks. *ICLR (Poster)* (2018)
- [34] Wang, D., Lin, J., Cui, P., Jia, Q., Wang, Z., Fang, Y., Yu, Q., Zhou, J., Yang, S., Qi, Y.: A Semi-supervised Graph Attentive Network for Financial Fraud Detection. In: ICDM. pp. 598–607 (2019)
- [35] Wang, X., Jin, B., Du, Y., Cui, P., Tan, Y., Yang, Y.: One-class graph neural networks for anomaly detection in attributed networks. *Neural Computing and Applications* **33**(18), 12073–12085 (2021)
- [36] Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: International Conference on Machine Learning. pp. 6861–6871. PMLR (2019)
- [37] Wu, J., He, J., Liu, Y.: Imverde: Vertex-diminished random walk for learning imbalanced network representation. In: IEEE International Conference on Big Data (2018)
- [38] Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: ICLR (2019)
- [39] Xu, X., Yuruk, N., Feng, Z., Schweiger, T.A.J.: SCAN: a structural clustering algorithm for networks. In: SIGKDD. pp. 824–833 (2007)
- [40] Zeni, M., Zhang, W., Bignotti, E., Passerini, A., Giunchiglia, F.: Fixing mislabeling by human annotators leveraging conflict resolution and prior knowledge. *Association for Computing Machinery* **3**(1) (2019)
- [41] Zhang, C., Song, D., Huang, C., Swami, A., Chawla, N.V.: Heterogeneous graph neural network. In: SIGKDD. p. 793–803 (2019)
- [42] Zheng, Y., Jin, M., Liu, Y., Chi, L., Phan, K.T., Pan, S., Chen, Y.P.P.: From unsupervised to few-shot graph anomaly detection: A multi-scale contrastive learning approach. *arXiv* **2202.05525** (2022)