Identifying Anomalous Edges with Link Sampling and Consensus

Abdullah Karaaslanli, Panagiotis A. Traganitis and Selin Aviyente

Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA

2024 58th Asilomar Conference on Signals, Systems, and Computers | 979-8-3503-5405-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/IEEECONF60004.2024.10942669

***Abstract*—Graphs and graph structured data are ubiquitous in many applications as they readily represent datasets that lie in irregular, yet structured, domains. Due to their popularity, a plethora of methods have been developed to learn from graph-structured data, which have been shown to be effective in many real-world applications including biology, finance, and social sciences, among others. However, these methods generally assume that the observed graph is free of corruption. This assumption does not hold in cases where the graph includes structural contamination, such as anomalous edges, which can degrade learning performance. This paper presents a method to identify anomalous edges that can be employed prior to learning methods to mitigate their effects. The proposed method employs link prediction (LP) to assign likelihood scores to the observed edges. As LP is not anomaly aware, we combine LP with ideas from sampling and consensus algorithms. LP is applied to subgraphs which tend to have fewer anomalies. Edge anomaly scores are then obtained by judiciously combining LP prediction results across subgraphs. Preliminary results indicate the effectiveness of the proposed method.**

I. Introduction

Recent years have seen a surge in machine learning and signal processing on graphs and graph structured data. Such data is encountered in many applications including biology, economics and sociology where graphs represent dependent entities and their associated data. For example, in sociology, people and their relations can be represented as a graph where personal attributes can be modeled as signals defined on the graph. A plethora of methods have been developed to learn from graph data for various problems including clustering, embedding and classification. Although these methods have increased our understanding regarding graph datasets, they mostly assume that the available graph data is free of corruptions, or anomalies. However, graph data can include different types of anomalies resulting from deliberate attacks or observational noise. One such corruption is anomalous interactions, or edges, between entities that are not supposed to be connected. Anomalous edges appear in a range of applications, such as fraudulent transactions in financial networks [1] or spam calls in communication networks [2]. As anomalous edges can degrade the performance of graph learning methods [3], there is a strong need for developing approaches to detect them in order to to mitigate their effects on algorithms.

Various methods have been developed for anomalous edge detection and they can be grouped into three categories [4]: residual-based detection; embedding based detection; and statistical model based detection. Methods in the first category seek a "clean" version of the graph and consider residual edges as anomalies. Such methods, identify "clean" graphs by typically assuming the adjacency matrix is a low-rank matrix, while anomalous edges are modeled as sparse corruptions [5], [6]. Embedding based detection seeks vectorial representations for edges using node or edge embedding approaches [7], [8]. Anomaly detection methods for vectorial data [9] are then used to find anomalous edges. Finally, in statistical model based detection the graph is assumed to be generated by a random graph model [10], [4]. Edges deviating significantly from the presumed model are deemed as anomalies.

Although the aforementioned methods have been used successfully, they have some shortcomings that need to be addressed. First, they typically rely on assumptions that might not hold. For example, the low-rank assumption imposes specific structure on a graph, e.g. a community structure, which is not the case for all observed graphs. Similarly, statistical modeling relies on random graph models that might not be the true model generated the graph. Secondly, existing anomalous edge detection methods use all edges without explicitly distinguishing nominal edges from anomalous ones. For instance, embedding based methods learn edge representations using all edges instead of learning only from the uncorrupted edges, which lowers the reliability of anomalous edge detection. Finally, some methods assume existence of a training set of clean edges, which may not always be available.

To address these shortcomings, this work puts forth a novel link prediction (LP) based method for detecting anomalous edges. In LP, a model that can identify missing edges is learned from the observed graph; the LP model returns a score indicating the likelihood of two nodes being connected. The LP model can also be used to identify anomalous edges by calculating scores for observed edges [11]. However, the direct application of LP to detect anomalous edges is sub-optimal, since existing LP models are agnostic to anomalous edges. This in turn, can reduce the performance of LP for anomaly detection. To overcome this issue, we propose a novel scheme (LinkSAC), inspired by sampling and consensus (SAC) approaches [12], [13], to improve LP for anomalous edge detection. In particular, LinkSAC learns multiple LP models from a set of subgraphs, rather than one LP model for the whole graph. The subgraphs are sampled from the whole graph such that they are "cleaner" than the whole graph in

This work was supported in part by NSF CCF-2312546.

Authorized licensed use limited to: Michigan State University. Downloaded on May 26,2025 at 14:11:34 UTC from IEEE Xplore. Restrictions apply.

the sense that they include lower number of anomalies. As a result, edge likelihood scores learned from such subgraphs are less affected by anomalies, improving the detection of anomalous edges. The edge likelihood scores estimated by LP for subgraphs are then combined through a consensus step, to yield the final scores for anomaly detection.

The proposed method enjoys unique advantages compared to the state-of-the-art anomalous edge detection approaches. First, it does not make any assumptions about the structure of nominal edges. Rather, it relies on the chosen LP algorithm to decide what constitutes a normal edge. This enables one to utilize different off-the-shelf LP algorithms to model nominal edges, which in turn provides flexibility, in the sense that the proposed algorithm can handle different types of graph data. For instance, anomalous edge detection in attributed graphs can be performed by simply using an LP algorithm that can learn from node attributes [14]. Furthermore, the proposed approach employs sampling to find "clean" subgraphs, hereby mitigating the negative effects of anomalous edges when modeling normal edges. Finally, the proposed approach is unsupervised.

II. Background

A. Graphs and Anomalous Edges

A graph is a mathematical object represented as a tuple $G = (V, E)$ where V is the node, or vertex, set with $|V| = N$ and $E \subseteq V \times V$ is the edge, or link, set with $|E| = M$. An edge between nodes u and v is represented as e_{uv}. If $e_{uv} \in E$ implies $e_{vu} \in E$, then the graph is called undirected. Unless otherwise noted, graphs considered are undirected and extensions to other graph types are discussed when necessary. The neighborhood of a node u is defined as $\mathcal{N}_u = \{v : e_{uv} \in E\}$ and $d_u = |\mathcal{N}_u|$ is its degree. For a graph with anomalous edges, E is partitioned into two, i.e., $E = E_{\rm a} \cup E_{\rm c}$ where $E_{\rm a}$ denotes the set of corrupted or anomalous edges and $E_{\rm c}$ is the set of "clean" or "nominal" edges and $|E_{\rm a}| \ll |E_{\rm c}|$. The anomaly size of G is defined as $\eta(G) := |E_{\rm a}|/|E|$. The next subsections outline LP and SAC methods which are the basis of our proposed method.

B. Link Prediction

In link prediction, one aims to identify missing edges in a graph G by learning a model $\mathcal{M}_{\rm LP} : V \times V \to \{0, 1\}$, that maps a node pair (u, v) to 0 or 1, with 1 indicating the presence of an edge between u and v [15]. State-of-the-art approaches for LP learn $\mathcal{M}_{\rm LP}$ in a "supervised" fashion [16], that is,

$$\mathcal{M}_{\rm LP} \coloneqq \mathcal{C}(\mathcal{F}(u, v; \boldsymbol{\theta}); \boldsymbol{\phi}), \;\; u, v \in V, \tag{1}$$

where $\mathcal{F} : V \times V \to \mathbb{R}^d$ is an embedding model, parametrized by $\boldsymbol{\theta}$, that maps a node pair to a d-dimensional vector; $\mathcal{C} : \mathbb{R}^d \to \{0, 1\}$ is a binary classifier with parameters $\boldsymbol{\phi}$ which assigns $\mathcal{F}(u, v; \boldsymbol{\theta})$ to the final result of 0 or 1. Learning $\mathcal{M}_{\rm LP}$ from the observed G is then equivalent to learning $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. Many LP algorithms have been proposed, each different on the parametrization of $\mathcal{F}$ and $\mathcal{C}$ and the algorithm used for estimating them [15], [17], [18].

Once $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are learned, one can also calculate an LP score l_{uv} for a node pair (u, v) as follows:

$$l_{uv} = \Pr[\mathcal{C}(\mathcal{F}(u, v; \boldsymbol{\theta}); \boldsymbol{\phi}) = 1 \mid G], \tag{2}$$

where $\Pr[\cdot]$ indicates the probability of an event. For an unconnected node pair u and v, $e_{uv} \notin E$, l_{uv} indicates the likelihood of u and v being connected with an edge, given G. Next, we describe the basic principles of random sampling and consensus methods.

C. Sampling and Consensus (SAC)

Consider a dataset $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ where $\mathbf{x}_i \in \mathbb{R}^d$ are data points with associated labels $y_i \in \mathbb{R}$. Suppose that $\mathcal{X}$ includes a subset of anomalous, or outlying, data points, $\mathcal{X}_{\rm a}$, that deviate from the true data distribution of $\mathcal{X}$. SAC is an approach developed for robust learning from such datasets, while mitigating the effects of $\mathcal{X}_{\rm a}$ on the estimated model. As the name suggests, it consists of two steps: the sampling step, and; the consensus step. During the sampling step, one learns multiple models $\{\mathcal{M}^t\}_{t=1}^T$; each model is estimated from a subsample $\mathcal{X}^t \subset \mathcal{X}$ with $|\mathcal{X}^t| < N$. If $\mathcal{X}^t \cap \mathcal{X}_{\rm a} = \varnothing$, the model $\mathcal{M}^t$ is not affected by $\mathcal{X}_{\rm a}$. On the other hand, if $\mathcal{X}^t \cap \mathcal{X}_{\rm a} \neq \varnothing$, $\mathcal{M}^t$ is learned from a contaminated dataset, leading to a degraded model. During the consensus step, one finds the best performing $\mathcal{M}^t$:

$$\mathcal{M}^* = \operatorname*{argmin}_{\mathcal{M}^t} \mathcal{L}(\mathcal{M}^t, \mathcal{X}), \tag{3}$$

where $\mathcal{L}$ is a loss function measuring the quality of $\mathcal{M}^t$ when it is applied to the whole dataset $\mathcal{X}$. $\mathcal{M}^*$ is assumed to be learned from a clean subsample and used as the final model. The following section outlines how we use LP and SAC to identify anomalous edges in a graph.

III. Detecting Anomalous Edges with LP and SAC

Consider a graph $G = (V, E)$ whose edge set includes a subset of anomalous edges represented as $E_{\rm a}$. The goal of *unsupervised* anomalous edge detection is to identify $E_{\rm a}$ using only the information provided by G without any supervision. In particular, our aim is to estimate an *anomaly score*, s_{uv}, for each edge $e_{uv} \in E$ such that the scores for edges in $E_{\rm a}$ are higher than those in $E_{\rm c}$.

An LP model $\mathcal{M}_{\rm LP}$ can be utilized for this problem as:

$$s_{uv} = 1 - l_{uv}, \;\; \forall e_{uv} \in E, \tag{4}$$

where l_{uv} is the likelihood score defined in (2). Different than LP, where l_{uv}'s are calculated for unconnected node pairs to identify missing edges, (4) employs $\mathcal{M}_{\rm LP}$ to calculate scores for observed edges E. In this case, l_{uv} indicates how well e_{uv} fits to the overall topology of the graph. Small values of l_{uv} indicate that e_{uv} is not likely to be an edge, i.e. an anomaly. However, off-the-shelf LP algorithms assume all edges in E to be true edges when learning the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ of (1). This assumption does not hold when anomalous edges

Authorized licensed use limited to: Michigan State University. Downloaded on May 26,2025 at 14:11:34 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 LinkSAC

Input: G, Number of Subgraphs T, Subgraph Size S
Output: Final Scores s_{uv}'s

for $(t \in \{1, \dots, T\}$ **do**
  $G^t \leftarrow \mathrm{SubgraphSampler}(G, S)$
  $\mathcal{M}^t_{\mathrm{LP}} \leftarrow \mathrm{TrainLP}(G^t)$
  $\mathcal{S}^t \leftarrow \mathrm{ApplyLP}(\mathcal{M}^{\mathrm{t}}_{\mathrm{LP}}, \mathrm{G^t})$
end for
Calculate s_{uv} with (5) for all $e_{uv} \in E$

exist, which results in a drop in the quality of $\mathcal{M}_{\mathrm{LP}}$, and subsequently leading to a suboptimal anomaly score s_{uv}.

To overcome this issue, we propose to integrate LP with SAC to mitigate the effects of E_{a} on s_{uv}. Similar to SAC, our proposed approach first samples T subgraphs $\mathcal{G}^t = (V^t, E^t)$ from G, with $|V^t| = S < |V|$ indicating the subgraph size. Subgraphs are sampled such that $\bigcup_{t=1}^{T} E^t = E$ and $E^t \cap E^{t'}$ is not necessarily an empty set for $t \neq t'$. From each G^t, an LP model $\mathcal{M}^t_{\mathrm{LP}}$ is learned and used to obtain a score set $\mathcal{S}^t = \{s^t_{uv} : e^t_{uv} \in E^t\}$ as described in (4). During the consensus step, we aggregate these score sets per $e_{uv} \in E$ to obtain the final anomaly scores:

$$s_{uv} = \mathrm{agg}(\{s^t_{uv} : e_{uv} \in E \cap E^t\}), \tag{5}$$

where $\mathrm{agg}(\cdot)$ is an aggregation function, such as mean or median. The proposed method is referred to as LinkSAC and is summarized in Algorithm 1. There are two subprocesses $\mathrm{SubgraphSampler}$ and $\mathrm{TrainLP}$ (with corresponding $\mathrm{ApplyLP}$) in Alg. 1 that need to be selected and are discussed below.

A. LP Model Selection

LP model selection is driven by two main considerations: graph type; and presumed dynamics of nominal and anomalous behavior. Different graph types benefit from tailored algorithms, e.g., attributed graphs benefit from attribute-aware LP models [14], while directed graphs, call for directional LP models [19]. LP should also be selected based on the assumptions related to the normal behavior in the graph. For instance, if edges in E_{c} are modeled to be homophilic, i.e. occuring between similar nodes, then the chosen LP algorithm should preserve this property [20]. In this case, anomalous edges would correspond to the edges between dissimilar nodes. Another example is community structured graphs where anomalous edges occur between nodes from different communities. In such a case, community-aware LP models would be useful [21].

B. Subgraph Sampling

The key idea behind SAC approaches is that subsamples should be either free from contamination or have minimal levels of it. Extending this idea to Alg. 1, we want to find subgraphs that are less contaminated compared to the whole graph, i.e. $\eta(G^t) < \eta(G)$. To find such subgraphs, exploration-based graph sampling is used; subgraphs are sampled by traversing over the graph topology with a stochastic process, such as a random walk [22]. In particular, the sampler starts from a seed node u_0 and iteratively moves over the graph to obtain a sequence $\{u_i \in V : i = 0, 1, 2 \dots\}$ where u_i is the node visited by the sampler at the i-th iteration. Node u_i is randomly chosen based on the following probability distribution defined over the node set,

$$\mathcal{P}_i(v) = \Pr[u_i = v \mid u_{i-1}, u_{i-2}, \dots, u_0],\ v \in V. \tag{6}$$

The sampler traverses the graph until it has visited S unique nodes, collected in V^t. Then the induced subgraph, $G^t = (V^t, E^t)$, is deemed as one graph sample. In general, graph exploration-based samplers differ based on the definition of $\mathcal{P}_i$'s, but typically lead to subgraphs which include topologically close nodes [23]. Since an anomalous edge is defined as an edge between nodes that are not expected to be connected, such as nodes that are topologically distant, it can be said that exploration-based graph sampling is biased toward returning clean subgraphs.

To make this discussion concrete, consider simple random walk (SRW) based sampling, where $\mathcal{P}_i(v)$ is defined as [24]

$$\mathcal{P}_i(v) = \begin{cases} 1/d_{u_{i-1}} & v \in \mathcal{N}_{u_{i-1}} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Let $\mathcal{W}$ represent the simple random walker. Consider a node r that is visited by $\mathcal{W}$ during its sampling process. Assume there are two edges incident to r: e_{rs} and e_{rv} where $e_{rs} \in E_{\mathrm{c}}$ and $e_{rv} \in E_{\mathrm{a}}$. Let $\Pr[e_{rs} \in E^t]$ be the probability of e_{rs} being in the sampled subgraph, which is equal to the probability of both r and s being visited by $\mathcal{W}$, i.e. $\Pr(\{r, s\} \subset V^t)$. Similarly, define these probabilities for r and v. For our case, it is desirable to have $\Pr(\{r, s\} \subset V^t) > \Pr(\{r, v\} \subset V^t)$, indicating clean edges are more probable to be in the sampled graph than the anomalous ones. This occurs if it easier for $\mathcal{W}$ to go to s from r than it is to go to v, which happens if there are more and shorter paths between r and s than there are between r and v.

In order to study this condition, we can consider the commute distance of SRWs. For nodes u and v, the commute distance c_{uv} is the average number of steps a random walk $\mathcal{W}$ starting at u needs to take to go to v for the first time and come back to u [25]. c_{uv} is a valid distance measure and it reduces as the number of paths between two nodes increases or the length of these paths decreases [26]. Based on commute distance, we then have $\Pr(\{r, s\} \subset V^t) > \Pr(\{r, v\} \subset V^t)$ if $c_{rs} < c_{rv}$.

As a valid distance measure, commute distance defines a type of similarity between nodes such that it is smaller for node pairs connected with a larger number of shorter paths. Since by definition anomalous edges occur between dissimilar node pairs, $\mathcal{W}$ is less likely to sample pairs connected with an anomalous edge. This implies the proposed method learns from subgraphs that are cleaner than overall graph.

Authorized licensed use limited to: Michigan State University. Downloaded on May 26,2025 at 14:11:34 UTC from IEEE Xplore. Restrictions apply.

Table I
PROPERTIES OF THE WIKIPEDIA AND FACEBOOK GRAPHS

	Number of Nodes	Number of Edges
Wikipedia	2,277	31,421
Facebook	5,908	41,729

IV. NUMERICAL TESTS

In this section, the proposed algorithm is tested on real world graphs with synthetically added anomalous edges. We consider two undirected graphs: the Wikipedia graph [27] and the Facebook graph [28]. The former dataset is a graph where nodes represent English Wikipedia pages on chameleons and edges reflect mutual links between them. The second graph is the social network of verified Facebook politician pages where edges correspond to mutual likes between the pages. Basic properties of both graphs are shown in Table I. Two different types of synthetic anomalies are added to the datasets: 1) ρM unconnected node pairs are randomly selected and connected with an edge where M is the number of edges in the original graph and $\rho > 0$ determines the anomaly size of the graph; and 2) adversarial attack anomalies, where ρM anomalous edges are added using the algorithm of [29], which connects node pairs such that random walk based node embedding methods are maximally harmed. This technique has two hyperparameters: embedding dimension and random walk length, which we set to 32 and 5, respectively. The performance of the proposed *LinkSAC*, i.e. Alg. 1, is compared to LP without SAC (referred to as *noSAC*). To evaluate the effect of aggregation on the results of the proposed method, two different aggregation functions are used: mean and median, denoted as *LinkSAC - Mean* and *LinkSAC - Median*, respectively[1]. The LP algorithm in *LinkSAC* and *noSAC* is variant of the algorithm proposed in [30]. In particular, we use 5 topological features commonly used in link prediction for $\mathcal{F} : V \times V \rightarrow \mathbb{R}^5$: the number of common neighbors, preferential attachment, Jaccard similarity, Adamic Adar and resource allocation [31]. For $\mathcal{C}$, we employed a random forest classifier with 100 trees, optimizing the Gini index. Subgraphs are sampled using random walk with restart, which is a random walk where at each iteration it can teleport back to the initial node with probability 0.1. The sampled subgraphs have $S = \lfloor \alpha N \rfloor$ nodes where $0 < \alpha < 1$ determines the subgraph size. Number of subgraphs T is set to a value such that each edge appears at least in 10 subgraphs. For all experiments, the figure of merit is the area under precision and recall curve (AUPRC). Let $\mathbf{y} \in \mathbb{R}^M$ be a binary vector with $y_i = 1$ if i-th edge in E is anomalous. Also define $\widehat{\mathbf{y}} \in \mathbb{R}^M$ where $\widehat{y}_i$ is the anomaly score of ith edge in E returned by Alg. 1, i.e., if the i-th edge is e_{uv}, then $\widehat{y}_i = s_{uv}$. AUPRC is then calculated by constructing precision-recall curve by comparing $\mathbf{y}$ and $\widehat{\mathbf{y}}$ and calculating the area beneath this curve. In the following two experiments, average performance over 10 Monte Carlo simulations is reported.

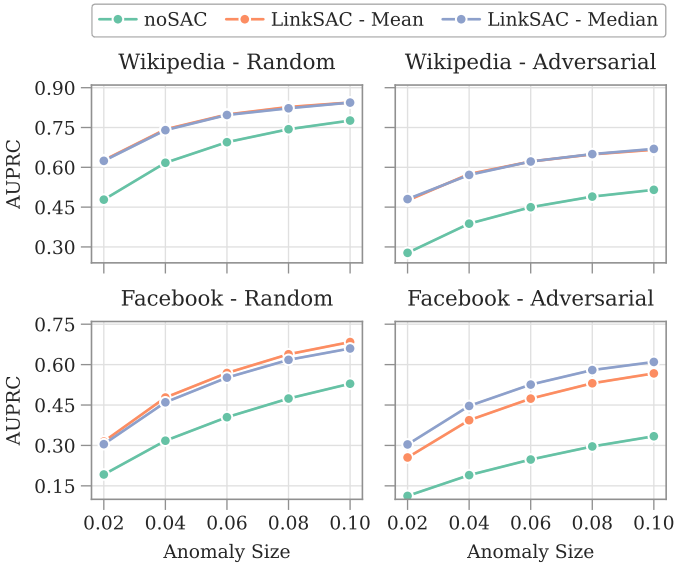

Figure 1. Performance of anomalous edge detection as a function of ρ.

Experiment 1: We first study how the fraction of anomalous edges (ρ) affects performance while fixing α to 0.05. Results are reported in Figure 1. The proposed approach shows superior performance than LP without SAC irrespective of the aggregation function for both datasets and anomaly types. This indicates that SAC improves LP algorithm as it mitigates the effect of anomalous edges on LP. We observe that AUPRC values are higher for random anomalies than adversarial ones. As the latter is designed to cause more harm on the graph structure while being less apparent, it is expected that their detection is more challenging.

Experiment 2: In this experiment, we study the effect of α on the proposed approach while fixing ρ to 0.05. The subgraphs employed in *LinkSAC* increase in size as α increases. Furthermore, the anomaly size of each subgraph increases as they get larger, since one needs to traverse larger parts of the graph, resulting in distant node pairs being sampled together. Thus, we expect the performance of the proposed method to drop with increasing α. Figure 2 shows the AUPRC values of all three methods. As expected the proposed approach degrades with increasing α values. However, its performance is still better than LP without SAC.

V. CONCLUSIONS

In this paper, we presented LinkSAC, a method to detect anomalous edges using LP and SAC. Existing work on anomalous edge detection assumes specific structures for the graph, which may not hold in practice. LinkSAC address this shortcoming by relying on LP, which enables one to employ different LP algorithms suitable for the studied graph. By integrating LP with SAC, our algorithm reduces effect

[1]Code for the proposed method can be found at https://github.com/abdkarr/LinkSAC

Authorized licensed use limited to: Michigan State University. Downloaded on May 26,2025 at 14:11:34 UTC from IEEE Xplore. Restrictions apply.

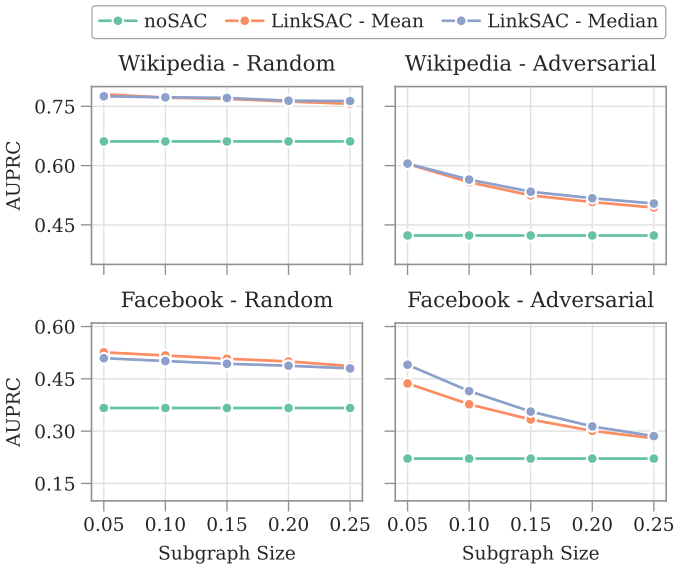

Figure 2. Performance of anomalous edge detection as a function of α.

of anomalous edges on LP. Our future work will focus on developing new subgraph sampling methods that minimize the amount of anomalies in subgraphs based on the discussion in Section III-B. Consensus step will also be investigated from the perspective of crowdsourcing [32] to obtain better aggregation functions. Finally, more extensive numerical tests will be performed and comparison against existing anomalous edge detection algorithms will be conducted.

REFERENCES

[1] R. J. Bolton, D. J. Hand *et al.*, "Unsupervised profiling methods for fraud detection," *Credit scoring and credit control VII*, pp. 235–255, 2001.

[2] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri, "Know your neighbors: Web spam detection using the web topology," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 423–430.

[3] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 2847–2856.

[4] R. Luo, B. Nettasinghe, and V. Krishnamurthy, "Anomalous edge detection in edge exchangeable social network models," in *Conformal and Probabilistic Prediction with Applications*. PMLR, 2023, pp. 287–310.

[5] H. Tong and C.-Y. Lin, "Non-negative residual matrix factorization with application to graph anomaly detection," in *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, 2011, pp. 143–153.

[6] K. D. Polyzos, C. Mavromatis, V. N. Ioannidis, and G. B. Giannakis, "Unveiling anomalous edges and nominal connectivity of attributed networks," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2020, pp. 726–730.

[7] L. Ouyang, Y. Zhang, and Y. Wang, "Unified graph embedding-based anomalous edge detection," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[8] D. Duan, C. Zhang, L. Tong, J. Lu, C. Lv, W. Hou, Y. Li, and X. Zhao, "An anomaly aware network embedding framework for unsupervised anomalous link detection," *Data Mining and Knowledge Discovery*, pp. 1–34, 2023.

[9] C. C. Aggarwal and C. C. Aggarwal, *Outlier ensembles*. Springer, 2017.

[10] R. Guimerà and M. Sales-Pardo, "Missing and spurious interactions and the reconstruction of complex networks," *Proceedings of the National Academy of Sciences*, vol. 106, no. 52, pp. 22 073–22 078, 2009.

[11] M. J. Rattigan and D. Jensen, "The case for anomalous link discovery," *Acm Sigkdd Explorations Newsletter*, vol. 7, no. 2, pp. 41–47, 2005.

[12] V. N. Ioannidis, D. Berberidis, and G. B. Giannakis, "Unveiling anomalous nodes via random sampling and consensus on graphs," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5499–5503.

[13] R. C. Bolles and M. A. Fischler, "A ransac-based approach to model fitting and its application to finding cylinders in range data." in *IJCAI*, vol. 1981, 1981, pp. 637–643.

[14] B. Taskar, M.-F. Wong, P. Abbeel, and D. Koller, "Link prediction in relational data," *Advances in neural information processing systems*, vol. 16, 2003.

[15] T. Zhou, "Progresses and challenges in link prediction," *Iscience*, vol. 24, no. 11, 2021.

[16] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *Proceedings of the twelfth international conference on Information and knowledge management*, 2003, pp. 556–559.

[17] V. Martínez, F. Berzal, and J.-C. Cubero, "A survey of link prediction in complex networks," *ACM computing surveys (CSUR)*, vol. 49, no. 4, pp. 1–33, 2016.

[18] J. Li, H. Shomer, H. Mao, S. Zeng, Y. Ma, N. Shah, J. Tang, and D. Yin, "Evaluating graph neural networks for link prediction: Current pitfalls and new benchmarking," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[19] Q.-M. Zhang, L. Lü, W.-Q. Wang, Yu-Xiao, and T. Zhou, "Potential theory for directed networks," *PloS one*, vol. 8, no. 2, p. e55437, 2013.

[20] L. M. Aiello, A. Barrat, R. Schifanella, C. Cattuto, B. Markines, and F. Menczer, "Friendship prediction and homophily in social media," *ACM Transactions on the Web (TWEB)*, vol. 6, no. 2, pp. 1–33, 2012.

[21] Z. Wang, Y. Wu, Q. Li, F. Jin, and W. Xiong, "Link prediction based on hyperbolic mapping with community structure for complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 450, pp. 609–623, 2016.

[22] J. Leskovec and C. Faloutsos, "Sampling from large graphs," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 631–636.

[23] B. Rozemberczki, O. Kiss, and R. Sarkar, "Little ball of fur: a python library for graph sampling," in *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020, pp. 3133–3140.

[24] L. Lovász, "Random walks on graphs," *Combinatorics, Paul erdos is eighty*, vol. 2, no. 1-46, p. 4, 1993.

[25] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Transactions on knowledge and data engineering*, vol. 19, no. 3, pp. 355–369, 2007.

[26] P. G. Doyle and J. L. Snell, *Random walks and electric networks*. American Mathematical Soc., 1984, vol. 22.

[27] B. Rozemberczki, C. Allen, and R. Sarkar, "Multi-scale attributed node embedding," *Journal of Complex Networks*, vol. 9, no. 2, p. cnab014, 2021.

[28] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton, "Gemsec: Graph embedding with self clustering," in *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*, 2019, pp. 65–72.

[29] A. Bojchevski and S. Günnemann, "Adversarial attacks on node embeddings via graph poisoning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 695–704.

[30] A. Ghasemian, H. Hosseinmardi, A. Galstyan, E. M. Airoldi, and A. Clauset, "Stacking models for nearly optimal link prediction in complex networks," *Proceedings of the National Academy of Sciences*, vol. 117, no. 38, pp. 23 393–23 400, 2020.

[31] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, pp. 623–630, 2009.

[32] S. Ibrahim, P. A. Traganitis, X. Fu, and G. B. Giannakis, "Learning from crowdsourced noisy labels: A signal processing perspective," *arXiv preprint arXiv:2407.06902*, 2024.

Authorized licensed use limited to: Michigan State University. Downloaded on May 26,2025 at 14:11:34 UTC from IEEE Xplore. Restrictions apply.