RESEARCH

Open Access

Anomaly detection and community detection in networks

Hadiseh Safdari and Caterina De Bacco*

*Correspondence:
caterina.debacco@tuebingen.mpg.de

Max Planck Institute for Intelligent Systems, Cyber Valley, 72076 Tübingen, Germany

Abstract

Anomaly detection is a relevant problem in the area of data analysis. In networked systems, where individual entities interact in pairs, anomalies are observed when pattern of interactions deviates from patterns considered regular. Properly defining what regular patterns entail relies on developing expressive models for describing the observed interactions. It is crucial to address anomaly detection in networks. Among the many well-known models for networks, latent variable models—a class of probabilistic models—offer promising tools to capture the intrinsic features of the data. In this work, we propose a probabilistic generative approach that incorporates domain knowledge, i.e., community membership, as a fundamental model for regular behavior, and thus flags potential anomalies deviating from this pattern. In fact, community membership serves as the building block of a null model to identify the regular interaction patterns. The structural information is included in the model through latent variables for community membership and anomaly parameter. The algorithm aims at inferring these latent parameters and then output the labels identifying anomalies on the network edges.

Keywords: Probabilistic models, Latent variable models, Community detection, Anomaly detection

Introduction

Anomalies or outliers—deviations in the observed data so extreme as to arouse suspicion [1]—form an unavoidable and problematic obstacle for data scientists. Over the past decades, anomalies have engendered a growing sense of concern in fields as varied as intrusion detection for network systems [2–4], fraud detection in banking industry [5, 6], identifying fake users and events in communication networks, and medical condition monitoring [7], to name a few. Methods and techniques from various fields, such as statistics [8], data mining [9], machine learning [10, 11], and information theory [12], have been employed to address this problem.

Much of this growing body of work focuses on standard tabular datasets [13–16]. However, anomaly detection in network datasets, where many individuals interact in complex ways, has been lagging behind. In fact, in many complex systems the data is made of pairwise interactions between individuals, i.e., the only observed information. For instance, in social networks, we know the nature of interactions between the individuals, i.e., friendship, financial, but we may not have any metadata about the individuals. In this context, anomalies can merely be detected by considering the set of interactions, and measuring which nodes or edges manifest an interaction pattern that is significantly different from that of their peers. In the case of online social networks, for example, advanced detection techniques that are independent of profile information are needed to detect fake profiles and malicious activities.

Our main objective in this work is to investigate the anomaly detection problem in networks. We consider the problem of observing a network that can have two possible, and different, mechanisms for edge formation; one involves the majority of the edges, whereas the other is an anomaly that we aim to detect. In other words, we have a regular pattern of interactions, and an anomaly. The latter belongs to a subset of interactions that deviates from the regular pattern.

In many networked systems, in particular social networks, the interaction pattern is driven by community membership: individuals belong to groups and this determines how they interact [17, 18]. To properly detect anomalies, one should incorporate this insight to build a suitable null model that distinguishes between regular interactions and those instead relatable to malicious activities. Thus, we focus on networks that display community patterns as the regular mechanism.

Efforts have been made in this area and various models have been proposed for applying community detection approaches in anomaly detection. For instance, Prado-Romero et al. [19] proposes an adaptive method to detect anomalies using the most relevant attributes for each community. In general, most of the approaches focus on attributed graphs to predict anomalous behavior [20–23].

Probabilistic generative models are however a powerful approach to tackle community detection, as they allow to incorporate domain knowledge about how interactions arise into rigorous probabilistic models. However, they have been rarely used in the context of anomaly detection. Along these lines, [24] propose a Bayesian model that combines network edges with additional information on nodes to identify anomalous nodes. Here, instead we do not assume any extra information being available a priori beyond the network structure.

In this work, we aim to build our model upon recent developments in community detection [25] to address anomaly detection in networks. Specifically, we aim at incorporating latent variables that measure the extent to which edges are classified as anomalous, together with latent variables for the hidden community structure. More specifically, by starting with an expressive generative model that captures the interaction patterns observed in network datasets, we can improve predictive power in detecting anomalous network interactions as well. The task is to infer both types of latent variables, i.e., communities and anomalies.

We tackle the problem by building the core foundational probabilistic generative model, while considering the existence of individual anomalous edges. Our model outputs labels for edges, identifying them as legitimate or anomaly. We assume that the only data we observe is the set of edges, coded by the adjacency matrix of entries A_{ij}, containing the weight of an edge between nodes i and j. Our goal is to determine which of the two possible mechanisms generated the edge and to label anomalies accordingly. Our approach is applicable to both directed and undirected networks.

We present an efficient and scalable algorithm which could be easily utilized by practitioners on networked datasets, without the need for extra node metadata.

Methods

Modeling anomalous edges

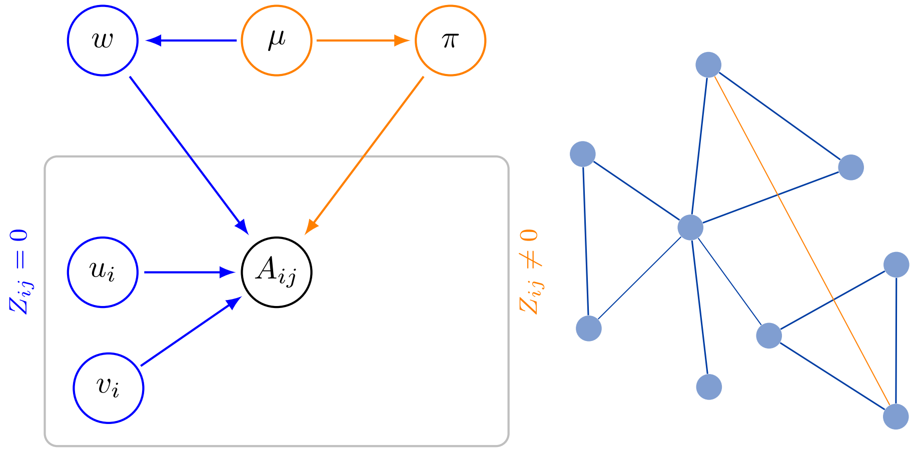
To achieve our goal of identifying anomalous edges and detecting communities in networks simultaneously, we need to explore statistical patterns in the connectivity of the networks dictated by community structure. This can be obtained using the formalism of network probabilistic generative models [25–27], as they are based on a rigorous theoretical framework and have efficient numerical implementations. These approaches assume that nodes are assigned to latent variables representing communities and these community memberships determine the probability that edges exist between the nodes. In particular, to model non-anomalous (or regular) edges, we consider the ideas proposed in [25]; as they flexibly apply to various types of networks with the characteristics needed in our problem: undirected and directed, weighted and unweighted, and it assumes mixed-membership community structure where nodes can belong to multiple communities.

We further assume that individual edges can be identified as anomalies when they deviate from what we consider a regular behavior, as described above. To model this, we introduce a binary random variable $Z_{ij} \in \{0, 1\}$: when $Z_{ij} = 1$ the edge (i, j) is an anomaly. This is a latent variable that is not known in advanced and needs to be learned from data. It determines the probability distribution from where the edge (i, j) is then extracted. From a generative modeling perspective, this setting can be understood as first drawing latent labels on edges, Z_{ij}, that determine which edges are anomalous and which edges are regular. Then drawing interactions A_{ij} between nodes from a specific distribution depending on the edge type, anomalous or regular. A schematic representation of our model is shown in Fig. 1. Formally, the generative model is:

$$Z_{ij} \sim \mathrm{Bern}(\mu) \tag{1}$$

$$A_{ij} \sim \begin{cases} \mathrm{Pois}(A_{ij};\pi) & \text{if} \quad Z_{ij}=1 \quad (\textit{anomalous edge}) \\ \mathrm{Pois}(A_{ij};M_{ij}) & \text{if} \quad Z_{ij}=0 \quad (\textit{regular edge}) \end{cases}, \tag{2}$$

where $\mu \in [0, 1]$ is an hyper-parameter controlling the prior distribution of Z. Here, we assume a Poisson distribution with mean value of M_{ij} for the formation of regular edges, and a Poisson distribution with mean value of π for the anomalous edge formation. The parameter $M_{ij} = \sum_k u_{ik} v_{jk} w_k$ is controlled by community structure as u_i, v_i are community membership vectors; $u_i = [u_{ik}]$ determines how much i belongs to community k considering the amount of *out-going* edges; $v_i = [v_{ik}]$ only considers *incoming* edges. An affinity matrix w of positive real-valued entries and dimension $K \times K$, where K is the number of communities, encodes the density of edges in different communities, i.e., it shapes assortative or disassortative structures of the communities. Here we assume an assortative structure where nodes are more likely to exist within rather than between communities. This implies that $w = [w_k]$ is diagonal. However, similar derivations can be found for other types of structures. Collectively, we indicate with $\Theta = \left(\{u_i\}, \{v_i\}, w, \pi, \left\{Z_{ij}\right\}\right)$ the latent variables of the model.



(a) Graphical model representation. (b) Network example.

Fig. 1 Model visualization. **a** Graphical model: the entry of the adjacency matrix A_{ij} is determined by the community-related latent variables u, v, w (blue), and by the anomaly parameters π (orange), depending on the values taken by the hyper-prior μ. **b** Example of possible realization of the model: blue edges display interactions mainly based on the community, the orange edge exhibits an anomalous edges

Based on this, the probability of an edge A_{ij} given the latent variables Θ can be written as:

$$P(A_{ij}|\Theta) = \text{Pois}(A_{ij}; M_{ij})^{1-Z_{ij}}\, \text{Pois}(A_{ij}; \pi)^{Z_{ij}}. \tag{3}$$

We assume a non-informative prior for w and sparsity-enforcing priors for the membership vectors u_i, v_i, thus encouraging the model to limit the number of non-zero entries.

Our goal is to estimate the latent variables, Θ, given the adjacency matrix A_{ij}. To this end, we perform the inference task by maximizing the log-likelihood, $L(\Theta) = \log P(A|\Theta)$ with respect to Θ. Given network data as the input, the desired output is inferring the probability that an edge is anomalous, as well as the underlying community structure, i.e., clustering nodes in communities. Our approach is capable of both learning how nodes are divided into groups and identifying those edges that are likely to be anomalous. We implement the inference task using an Expectation–Maximization (EM) scheme as detailed in "Convergence criteria" section. A pseudo-code of the algorithm is provided in Algorithm 1. We refer to our model for anomaly detection in networks with community structure as ACD.

The computational complexity of the algorithm scales as $O(EK + N^2)$, where E is the total number of edges. In most of the applications, K is usually much smaller than E and for sparse networks, as is often the case for real datasets, $E \propto N$. Hence, the complexity is dominated by $O(N^2)$. This contribution comes from terms containing Q_{ij} that are not also multiplied by A_{ij}, i.e. terms in the denominators of the updates in Algorithm 1. The matrix Q is a dense object that is necessary for classifying edges as anomalies. This may make it prohibitive to run our model on large systems. Exploring possible approximation to it to allow scaling to larger sizes is an interesting direction for future work.

```
Algorithm 1 ACD : EM algorithm.
Input: network $A = \{A_{ij}\}_{i,j=1}^{N}$,
       number of communities $K$.
Output: membership $u = [u_{ik}]$, $v = [v_{ik}]$; network affinity matrix $w = [w_k]$; mean
        value of Poisson anomaly distribution $\pi$; prior on anomaly indicator $\mu$.
Initialize $\Theta : (u, v, w, \pi, \mu)$ at random.
Repeat until $\mathcal{L}$ convergence:
  1. Calculate $\rho$ and $Q$ (E-step):
       $\rho_{ijk} = u_{ik} v_{jk} w_k / \sum_{k} u_{ik} v_{jk} w_k \ ,$
       $Q_{ij} = \frac{\mathrm{Pois}(A_{ij};\pi)\mathrm{Pois}(A_{ji};\pi)\,\mu}{\mathrm{Pois}(A_{ij};\pi)\mathrm{Pois}(A_{ji};\pi)\,\mu + \mathrm{Pois}(A_{ij};M_{ij})\,\mathrm{Pois}(A_{ji};M_{ji})\,(1-\mu)} \ .$
  2. Update parameters $\Theta$ (M-step):
       i) for each node $i$ and community $k$ update memberships:
          $u_{ik} = \frac{a - 1 + \sum_{j}(1-Q_{ij})\,A_{ij}\rho_{ijk}}{b + \sum_{j}(1-Q_{ij})\,v_{jk}w_k}$
          $v_{jk} = \frac{a - 1 + \sum_{i}(1-Q_{ij})\,A_{ij}\rho_{ijk}}{b + \sum_{i}(1-Q_{ij})\,u_{ik}w_k}$
       ii) for each community $k$ update affinity matrix:
          $w_k = \frac{\sum_{i,j}(1-Q_{ij})\,A_{ij}\rho_{ijk}}{\sum_{i,j}(1-Q_{ij})\,u_{ik}v_{jk}} \quad .$
       iii) update anomaly parameter:
          $\pi = \frac{\sum_{i,j} Q_{ij} A_{ij}}{\sum_{i,j} Q_{ij}} . \qquad (4)$
       iv) update prior on anomaly indicator:
          $\mu = \frac{1}{N(N-1)/2} \sum_{i<j} Q_{ij} \quad . \qquad (5)$
```

Results on synthetic data

In order to validate the performance of our model and investigate its applicability, we apply it to synthetic datasets sampled with our generative model, see Appendices for details ("Appendix 2: Generative model" section). These have known ground truth community memberships and anomalous edges. Hence, we assess the ability of our algorithm to identify anomalous edges and in detecting communities. Once parameters are inferred, we use point estimates of u_i, v_j to assign nodes to group and of Z_{ij} to classify edges. We compare these estimates with their respective ground truth values. As performance metrics we consider the F1 score and cosine-similarity (CS), respectively. We are interested in particular in assessing how these quantities vary with ρ_a, the fraction of anomalous edges over the whole set of edges.

Specifically, we generate synthetic data sets with $N = 500$ nodes, average degree of $\langle k \rangle = 20$, $K = 3$ hard communities of equal size with assortative structure and a range of $\rho_a \in [0, 1]$.

As a baseline model for comparison, we consider a version of our model that reduces to standard community detection (CD) as in [25]. This is obtained by setting $\mu, \pi = 0$ which are kept fixed as hyper-parameters in inference.

We observe that ACD significantly outperforms CD in detecting communities robustly across different values of ρ_a, as shown in Fig. 2. In particular, its performance is stronger within an intermediate region where $\rho \in [0.4, 0.6]$, i.e. when the majority of edges switches from being regular to being anomalous and CD's performance decays much faster. In terms of anomaly detection, we observe that the performance improves by the increase of the anomaly density, with the largest improvement achieved for small values of $\rho_a < 0.2$, before reaching a steady increase towards the maximum value of 1 for larger ρ_a.

Results on real world datasets

In order to verify the validity of the algorithm and evaluate its performance on real-world datasets, we carry out three experiments. We study three real-world datasets with node attributes available as potential ground truth for comparison for community membership of nodes. More details on the studied datasets are available in "Real data: dataset description" section.

Experiment 1: injection of anomalous edges

In the first experiment, we inject anomalous edges in a given input network. These edges are selected uniformly at random among all the possible pairs of nodes that are not already connected by an edge. Then, we apply our method on this altered network and measure the algorithm's performance in (i) detecting the injected edges, i.e., anomalous

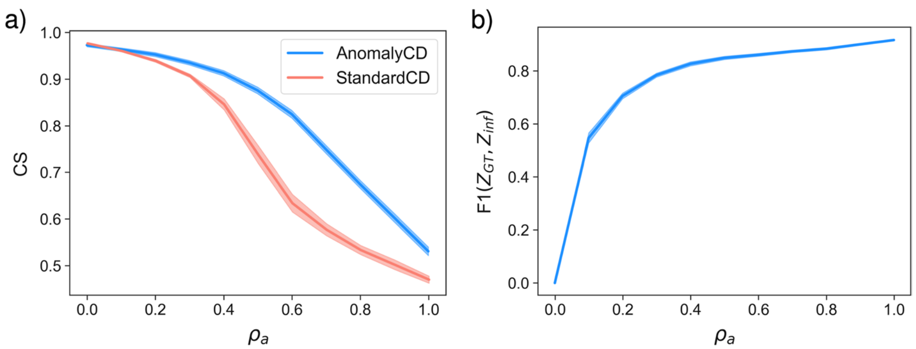


Fig. 2 Community detection and anomaly detection in synthetic networks. **a** Cosine similarity (CS) between ground truth and inferred communities and **b** the F1 score between the ground truth and the inferred anomalies. The model's ability to detect anomalous edges increases with increasing the ratio of anomalous edges in the network. Synthetic networks with $N = 500$ nodes, $\pi = 0.6$, and $K = 3$ communities of equal-size unmixed group membership generated with ACD. Lines and shades around them are averages and standard deviations over 20 sampled networks

edges, and (ii) detecting how communities are correlated with the node attributes available with the dataset. As performance metrics we measure precision, recall, and the Area Under the Curve (AUC) for anomaly detection and CS for community detection. We vary the fraction ρ_a of injected anomalous edges to assess how performance is impacted by this number.

Books about US politics The network we study in this experiment contains 105 books about US politics which were published around the 2004 presidential election [28] (POLBOOKS). In this network, nodes are books and an undirected, binary edge between two books indicates that those were co-purchased by the same costumer, for a total of 441 edges. Injected anomalies here represent books that are either mistakenly co-purchased or mistakenly accounted in the dataset.

The results of this experiment is presented in Fig. 3. While AUC and community detection are both robust against the number of edges injected in the network, precision and recall are more nuanced. This is due to the possibility of tuning the prior of Z_{ij} via μ in order to obtain different regimes in retrieving anomalous edges. As can be seen in Table 1 and Fig. 4, for a given level of injected anomalies, we can have high precision or high recall, depending on the initialized value of the prior. Hence, classification performance can be tuned towards either high precision or high recall by calibrating μ accordingly, depending on the practitioner's goal.

For instance, when a practitioner wants to be strict in the criteria of labeling an edge as anomalous, thus avoiding labeling as "anomalous" edges that are not, then one should

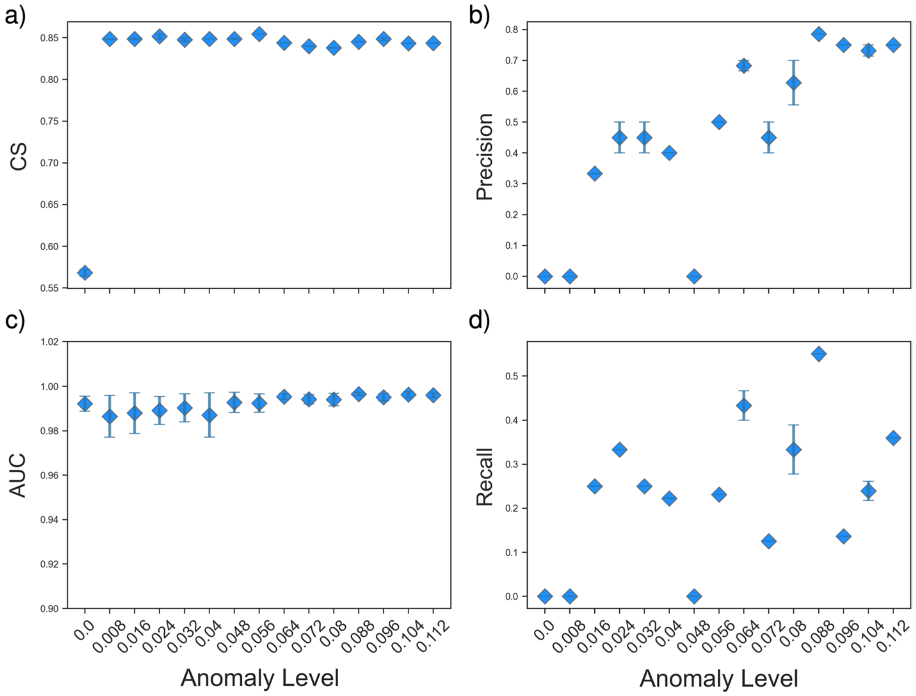


Fig. 3 The network of POLBOOKS with injected edges (Experiment 1). We compare the performance of the ACD in terms of community detection and the detection of the anomalous edges injected in the network. We tune the percentage of injected edges ρ_a. Here $\pi = 0.25$. Markers and errors denote means and standard deviations over 20 samples

Table 1 The confusion matrix for the network of POLBOOKS with injected edges (Experiment 1)

	μ	Precision	Recall	F1
High precision	0.5	0.78	0.32	0.45
High recall	0.75	0.27	0.64	0.38

We show how our model performs in terms of identifying anomalies—edges that have been injected in the dataset—as we vary the prior on Z, tuned by the parameter μ. Here $\pi = 0.25$ and $\rho_a = 0.087$

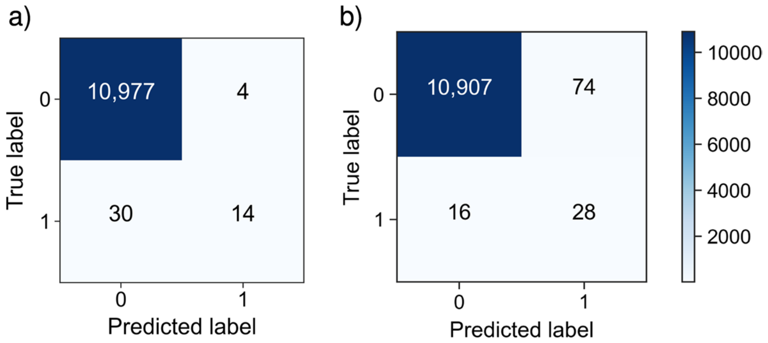


Fig. 4 The network of POLBOOKS with injected edges (Experiment 1). We compare the performance of ACD in detecting the anomalous injected edges by estimating the confusion matrix for tow different values to initialize μ. By adjusting the value of μ, we can tune the recall and precision. **a** $\mu = 0.5, \pi = 0.25$, **b** $\mu = 0.75, \pi = 0.25$

be more conservative with the prior, i.e. select a smaller μ. Instead, when the priority is to detect as many anomalies as possible (at the risk of mislabeling true edges) one should increase μ and thus increase recall. This choice should depend on the application at hand, in particular one should reflect on the potential cost of classifying an edge as anomaly when it is not and compare it with the potential cost of missing anomalous edges.

Experiment 2: 2-step inference of communities

In the second experiment we are interested in exploiting the information learned about anomalous edges to enhance performance in community detection. The hypothesis is that the presence of anomalies may corrupt community detection, for instance when anomalous edges connect two nodes that should not be part of the same communities. Using our model, we can act on the dataset by removing those edges that have higher probability of being anomalous, thus reducing noise in favor of better community detection. In practice, this is executed using a 2-step routine where we first run ACD on the input dataset to estimate Z_{ij}. Then, we remove those edges with higher probability of being anomalous. Finally, we perform regular CD (running ACD with $\pi = \mu = 0$) on the "cleaned" network to extract communities. We observe enhanced results in the community detection task after removing the anomalous edges.

Zachary's Karate Club We first test this second experiment on the dataset of Zachary's Karate Club to convey in more qualitative terms how model improves upon the community detection task. The network's small size of 34 nodes allows to better visualize the problem and how the 2-step routine works. This social and undirected network shows the interactions between members of a university karate club for a period of 3 years from

1970 to 1972. The members are the nodes and an edge between a pair of members indicates social interactions between them. During this period, due to administrative issues, the club splintered into two. We exploits the membership of the nodes in the new clubs as possible meaningful ground truth communities.

It is should be noted that what we refer to as the ground truth communities are in fact metadata of nodes that could be utilized to compare the resulting communities. The intention is merely to have a criterion for a quantitative comparison. In other words, we examine how the communities inferred by the algorithm are consistent with existing metadata. In all the real data studied in this work, this interpretation of the metadata as the ground truth is applied.

Figure 5 provides a visual representation of how the 2-step routine works. Communities inferred before and after removing anomalous edges are compared against those obtained using node attributes as ground truth. We remove two edges classified with the highest probability as anomalous, these are shown in red in Fig. 5a. Removing the red edge connecting two more central nodes has the effect of changing the community assigned to one of the two nodes, which is now aligned with the ground truth after running CD the second time. Instead, the other node in the removed edge keeps its community as detected in the first step, which was already aligned with the ground truth. As a result, the 2-step routine improves community detection performance. Notice also that removing the other anomalous edge does not impact performance, as the nodes involved in that edge do not switch communities and are already aligned with the ground truth. Hence, not all the removed edges may necessarily impact community detection the same.

We can now proceed analogously to apply the experiment on a larger dataset and present quantitative results on the POLBOOKS dataset. Figure 6 demonstrates the communities inferred by ACD using the 2-step routine and the inferred anomalous edges are shown in red in Fig. 6a as in the example before. We notice that the majority of

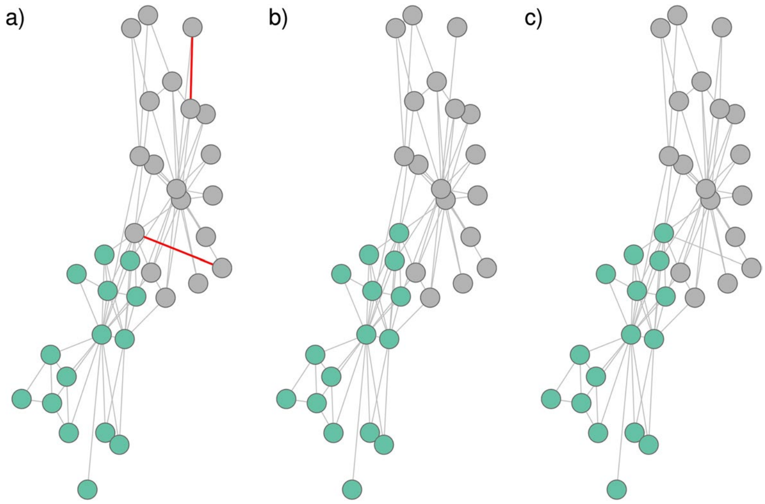


Fig. 5 The network of Zachary karate Club (Experiment 2). **a** The communities inferred by ACD. The edges inferred by our algorithm as the anomalous edges are shown in red. Here, $\pi = 10^{-4}$, $\mu = 0.1$. **b** The communities inferred by CD (i.e. ACD with $\pi = \mu = 0$). **c** Ground truth communities

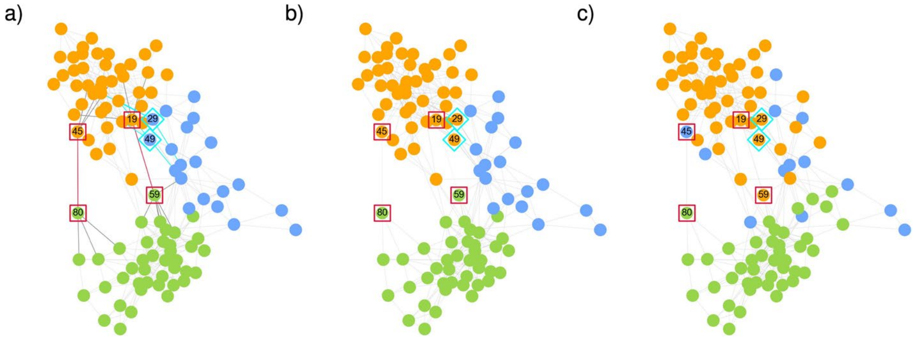


Fig. 6 The network of POLBOOKS (Experiment 2). We show an example output of running the 2-step algorithm where we first detect anomalous edges, then remove them and finally run regular CD on the pruned dataset.) The communities inferred by ACD before removal. The edges inferred by ACD as anomalous edges are shown in red. Here, $\pi = 0.0001$, $\mu = 0.5$. **b** The communities inferred by CD after removal of anomalies. **c** Ground truth communities. Red rectangles denote pairs of nodes that are connected by the edges inferred as anomalous. Cyan rectangles demonstrate the nodes that changed their community membership, after removing the aforementioned anomalous edges. Cyan edges in panel (**a**) present the edges connected to these

Table 2 The network of POLBOOKS (Experiment 2)

Measure	Removed anomalous edges	μ_0		
		0.01	0.05	0.5
CS	Yes	0.837 ± 0.015	0.834 ± 0.020	0.838 ± 0.014
	No	0.767 ± 0.039	0.766 ± 0.040	0.768 ± 0.040
AUC	Yes	0.882 ± 0.018	0.883 ± 0.019	0.888 ± 0.020
	No	0.953 ± 0.058	0.936 ± 0.056	0.922 ± 0.045

We present the ability of ACD in community detection, represented by cosine similarity (CS). Moreover, we validate the model by measuring the AUC in link prediction tasks. The results are robust with respect to the initial values of μ. By removing the edges detected as anomalies, the community detection task is improved. Here, $\pi = 10^{-3}$. The CS errors are calculated by averaging over 20 runs of the edge removal routine. To estimate AUC we perform 5-fold cross validation and report the averages and standard deviations over the fivefolds

the detected edges are between different communities. Comparing the communities detected in panels (a) and (b) with the ground truth communities in panel (c) we notice how the 2-step routine infers communities more aligned with the ground truth, as several nodes at the center of the figure switch communities from blue to green after removal (For more details on the soft community membership of nodes, see Fig. 10). In other words, ACD is capable of improving community detection by uncovering edges that interfere with the community detection process.

Quantitatively, this is shown in Table 2 by an increase of CS value from 0.77 to 0.84 after anomaly removal, consistently over different values of prior's parameter μ. Table 2 emphasizes the robustness of the model in detecting communities that are aligned with metadata information with respect to changing the initial value of μ. In order to compare the performance of ACD in community detection task, we applied Bayesian Poisson matrix factorization (BPMF) [29] on the POLBOOKS dataset, which results in $\text{CS} = 0.778 \pm 0.065$. In addition, in the table we report also the results of link prediction tests for model validation in the absence of ground truth (metadata are only a candidate for ground truth, true model parameters are unknown in real data). Specifically, we run 5-fold cross-validation and measure the AUC on the test dataset. Higher values indicate better performance in predicting missing edges, and better model's expressiveness. We see that the AUC only slightly drops when removing the anomalous edges, thus suggesting that removing information in an informed way (i.e. anomalous edges as detected by our model) enhances community detection without drastically affecting the AUC.

Experiment 3: Adding anomalous non-edges

The first two sets of analyses examined the ability of the algorithm in detecting the anomalous edges and the impact of removing those edges from the dataset in community detection and link prediction tasks. However, our model is also able to estimate the probability of a non-edge to be anomalous. This can be used for instance in cases where we expect certain connections between nodes to happen, and if they are not observed we can use our algorithm to detect potential missing edges. Hence, we design a new experiment to assess the possibility of improving the community detection task by adding edges between disconnected nodes. In other words, the algorithm detects unseen edges which may improve community detection if we were to add them to the network, in a similar 2-step routines as before, this time by adding, instead of removing edges.

As in the previous experiments, we apply ACD on the dataset to estimate Z. However, in this case, we select the entries corresponding to non-edges (i.e. $A_{ij} = 0$) which have highest probability of being anomalous and then add them to the network. Then, we apply regular CD on the final dataset and compare communities inferred before and after adding these edges.

American college football The experiment was tested on a network of football games between American colleges in the fall of 2000 [17]. Nodes in the network indicate the college teams and the undirected, and binary edges connecting them represent the games between the teams. The teams are divided into 12 conferences where members of each conference have more frequent games with each others compared to the games with members of other conferences. We use the conference membership as candidate ground truth community memberships to compare against. An anomalous non-edge corresponds to a game that has not been planned by the league's organizers but could have been played (for instance by adding more games to the fixtures or substituting with other games currently in the fixture), as it aligns with the pattern of existing games. In this context, selecting the fixtures is an important task for the organizers, as the set of matches that a team has to play can significantly impact its chance to go to the final National Championship.

In Fig. 7 we show a qualitative example of how communities change before and after addition of 6 non-existing edges classified with the highest probability of being anomalous (they correspond to 1% of the total number of edges). It is clear from this figure that addition of few edges impacts the community membership of several nodes, and not only those directly connected to the newly added edges. In particular, it strengthen the memberships of nodes in the communities of the nodes directly connected to the added edge (see e.g. the green and pink, whose nodes become less overlapping). In addition, it softens the membership of several nodes that are "Independent" (brown nodes in Fig. 7c), they do not belong to any conference. These are nodes that play several games against nodes in various conferences. In general both approaches achieve strong results in detecting communities that align with conference membership. However, the 2-step routine with edge

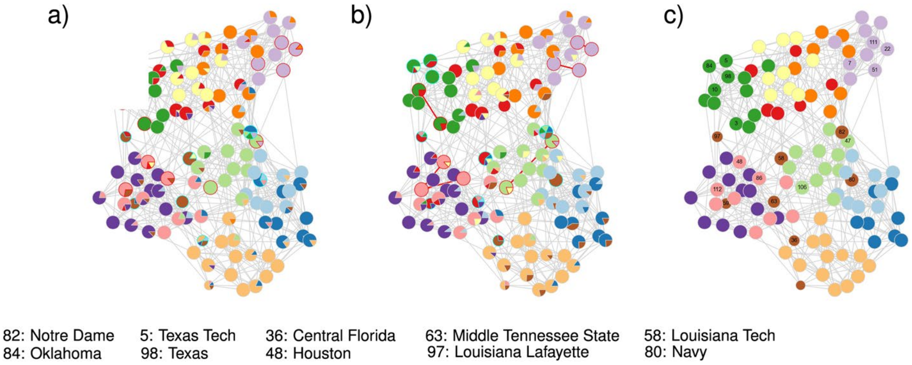


Fig. 7 The network of American college football after addition of edges (Experiment 3). **a** The communities inferred by CD on the original input data. **b** Community detection by CD algorithm on the dataset after addition of the potential inferred edges inferred by ACD (these edges are shown in red). **c** Ground truth communities. The nodes that changed their communities between **a** and **b** are shown with cyan border. Those with red border are nodes connecting to added edges. The name of some of the nodes impacted by the edge addition is added to the figure as the footnote

Table 3 The network of American college football (Experiment 3)

	CD on dataset	**ACD on dataset**	**CD on dataset with added edges**
CS	0.955 ± 0.003	0.957 ± 0.001	0.957 ± 0.003
F1	0.959 ± 0.005	0.957 ± 0.003	0.962 ± 0.006

We report CS and F1-score between the inferred communities and those given by metadata. Averages and standard deviations are over 20 runs of the edge addition routine

addition improves performance further, as detected by both CS and F1-score, see Table 3. These results show the flexibility of our model in detecting various types of anomalies and acting on them by suitably modifying the network to enhance community detection.

Discussion

We have proposed a probabilistic model for detecting anomalies in networks. It relies on the assumption that regular patterns of interactions are determined by community structure and exploits this insight to detect pairs of nodes, existing or non-existing ties, that deviate from regular behavior.

The algorithmic implementation uses an expectation–maximization routine that outputs both community membership of nodes and probability estimates for pairs of nodes of being anomalous or not. We find that in synthetic data it improves community detection while also showing robust performance in identifying anomalies.

In addition, in the case of real-world datasets, we have performed various experiments that show an increase in the model's ability in community detection tasks. Specifically, both in the experiment where the inferred anomalous edges were removed from the network, and in the case where non-existing but potential ties were identified and added to the network, there was an improvement in detecting communities that are aligned with metadata. Also, in another experiment, in which anomalous edges were injected into the system, our model showed high capability in detecting these ties.

We have focused here in anomalies on pairs of nodes, edges or non-edges but similar ideas and methods can be used to extend this model to anomalies on nodes. In this context, it may be interesting to explore future extensions that incorporate extra information, e.g. node attributes, along with community structure, as done for instance in [24, 30–32]. As accurately identifying anomalies is deeply connected with the chosen null model determining what regular patterns are, it is important to consider other possible mechanism for tie formation, beyond community structure. In recent works [33–35], we found that modeling community patterns together with reciprocity effects, leads to higher predictive performance, thus more expressive generative models. This could significantly change the performance of our foundational model as well. Hence, a natural next step is to include reciprocity in our model and measure how, by varying the intensity of these effects, anomaly detection improves (or decreases).

As a final remark, it is worth mentioning that what is referred to as an anomalous edge in this work should not necessarily be interpreted as an undesirable interaction or malicious activity. Indeed, an anomaly here reflects an unusual pattern, as compared to that of other edges, which can not be explained by the core structural pattern of the dataset, in this case driven by community structure. We encourage practitioners to carefully assess its qualitative interpretation based on the application at hand and preferably guided by domain expertise and knowledge.

Appendices

Appendix 1: Inference with expectation–maximization

Our goal is, given two mechanisms responsible for edge formation, first to determine the values of the parameters $\Theta : (u_{ik}, v_{ik}, w_k, \pi, \mu)$, which determine the relationship between the anomaly indicator Z_{ij} and the data, and then, given those values, to estimate the indicator Z_{ij} itself. We have the posterior:

$$P(Z,\Theta|A) = \frac{P(A|Z,\Theta)P(Z|\mu)P(\Theta)P(\mu)}{P(A)}. \tag{6}$$

Summing over all the possible indicators we have:

$$P(\Theta|A) = \sum_{Z} P(Z,\Theta|A), \tag{7}$$

which is the quantity that we need to maximize to extract the optimal Θ. It is more convenient to maximize its logarithm, log-likelihood, as the two maxima coincide. We use the Jensen's inequality:

$$L(\Theta) = \log P(\Theta|A) = \log \sum_{Z} P(Z,\Theta|A) \geq \sum_{Z} q(Z)\, \log \frac{P(Z,\Theta|A)}{q(Z)}, \tag{8}$$

where $q(Z)$ is a variational distribution that must sum to 1. In fact, the exact equality happens when,

$$q(Z) = \frac{P(Z, \Theta|A)}{\sum_Z P(Z, \Theta|A)}, \tag{9}$$

this definition is also maximizing the right-hand-side of Eq. (8) w.r.t. q.

Finally, we need to maximize the log-likelihood with respect to Θ to get the latent variables. This can be done in an iterative way using Expectation–Maximization algorithm (EM), alternating between maximizing w.r.t. q using Eq. (9) and then maximzing Eq. (35) w.r.t. Θ.

We start by derivation of Eq. (8) with respect to the individual parameters, for example we start by considering u_{ik}. We assume uniform prior w.r.t. Θ, but we can easily incorporate more complex choices if needed.

$$\sum_Z q(Z) \frac{\partial}{\partial u_{ik}} \left[\log \frac{P(Z, \Theta|A)}{q(Z)} \right] = \sum_Z q(Z) \frac{\partial}{\partial u_{ik}} \log P(Z, \Theta|A) \tag{10}$$

$$= \sum_Z q(Z) \frac{\partial}{\partial u_{ik}} \sum_{i,j} (1 - Z_{ij}) \, \log \mathrm{Pois}(A_{ij}; M_{ij}) \tag{11}$$

$$= \sum_{Z,j} q(Z) \, (1 - Z_{ij}) \, \frac{\partial}{\partial u_{ik}} \left[-u_{ik} v_{jk} w_k + A_{ij} \log \sum_k u_{ik} v_{jk} w_k \right] \tag{12}$$

$$= \sum_{Z,j} q(Z) \, (1 - Z_{ij}) \left[-v_{jk} w_k + A_{ij} \frac{\rho_{ijk}}{u_{ik}} \right] = 0, \tag{13}$$

where in the last equation we used once again Jensen's inequality with:

$$\rho_{ijk} = u_{ik} v_{jk} w_k / \sum_k u_{ik} v_{jk} w_k. \tag{14}$$

Defining $Q_{ij} = \sum_Z q(Z) \, Z_{ij}$, the expected value of Z_{ij} over the variational distribution, we finally obtain:

$$u_{ik} = \frac{\sum_j (1 - Q_{ij}) \, A_{ij} \rho_{ijk}}{\sum_j (1 - Q_{ij}) \, v_{jk} w_k}. \tag{15}$$

We find similar expression for v_{ik} and w_k,

$$v_{jk} = \frac{\sum_i (1 - Q_{ij}) \, A_{ij} \rho_{ijk}}{\sum_i (1 - Q_{ij}) \, u_{ik} w_k}, \tag{16}$$

$$w_k = \frac{\sum_{i,j} (1 - Q_{ij}) \, A_{ij} \rho_{ijk}}{\sum_{i,j} (1 - Q_{ij}) \, u_{ik} v_{jk}}. \tag{17}$$

For π:

$$\sum_{Z} q(Z)\frac{\partial}{\partial \pi}\left[\log \frac{P(Z,\Theta|A)}{q(Z)}\right] = \sum_{Z} q(Z)\frac{\partial}{\partial \pi} \log P(Z,\Theta|A) \tag{18}$$

$$= \sum_{Z} q(Z)\frac{\partial}{\partial \pi} \sum_{i,j} \left[Z_{ij} \log \text{Pois}(A_{ij};\pi)\right] \tag{19}$$

$$= \sum_{Z} q(Z)\frac{\partial}{\partial \pi} \sum_{i,j} \left[Z_{ij}\,(-\pi + A_{ij}\log \pi)\right] \tag{20}$$

$$= \sum_{Z,i,j} q(Z)\left[Z_{ij}\,(-1 + A_{ij}\,\frac{1}{\pi})\right] = 0, \tag{21}$$

yielding

$$\pi = \frac{\sum_{i,j} Q_{ij}A_{ij}}{\sum_{i,j} Q_{ij}}. \tag{22}$$

Similarly for μ:

$$\sum_{Z} q(Z)\frac{\partial}{\partial \mu}\left[\log \frac{P(Z,\Theta|A)}{q(Z)}\right] = \sum_{Z} q(Z)\sum_{i<j}\frac{\partial}{\partial \mu}\left[Z_{ij}\log \mu + (1-Z_{ij})\log(1-\mu)\right] \tag{23}$$

$$= \frac{1}{\mu}\sum_{i<j} Q_{ij} - \frac{1}{1-\mu}\sum_{i<j}(1-Q_{ij}), \tag{24}$$

yielding:

$$\mu = \frac{1}{N(N-1)/2}\sum_{i<j} Q_{ij}. \tag{25}$$

To evaluate $q(Z)$, we substitute the estimated parameters inside Eq. (9):

$$q(Z) = \frac{\prod_{i,j} \text{Pois}(A_{ij};\pi)^{Z_{ij}}\,\text{Pois}(A_{ij};M_{ij})^{1-Z_{ij}}\,\prod_{i<j}\mu^{Z_{ij}}\,(1-\mu)^{(1-Z_{ij})}}{\sum_{Z}\prod_{i,j} \text{Pois}(A_{ij};\pi)^{Z_{ij}}\,\text{Pois}(A_{ij};M_{ij})^{1-Z_{ij}}\,\prod_{i<j}\mu^{Z_{ij}}\,(1-\mu)^{(1-Z_{ij})}} \tag{26}$$

$$= \prod_{i<j} \frac{\left[\text{Pois}(A_{ij};\pi)\text{Pois}(A_{ji};\pi)\,\mu\right]^{Z_{ij}}\left[\text{Pois}(A_{ij};M_{ij})\,\text{Pois}(A_{ji};M_{ji})\,(1-\mu)\right]^{(1-Z_{ij})}}{\sum_{Z_{ij}=0,1}\left[\text{Pois}(A_{ij};\pi)\text{Pois}(A_{ji};\pi)\,\mu\right]^{Z_{ij}}\left[\text{Pois}(A_{ij};M_{ij})\,\text{Pois}(A_{ji};M_{ji})\,(1-\mu)\right]^{(1-Z_{ij})}} \tag{27}$$

$$= \prod_{i<j} Q_{ij}^{Z_{ij}}\,(1-Q_{ij})^{(1-Z_{ij})}, \tag{28}$$

where

$$Q_{ij} = \frac{\mathrm{Pois}(A_{ij};\, \pi)\mathrm{Pois}(A_{ji};\, \pi)\, \mu}{\mathrm{Pois}(A_{ij};\, \pi)\mathrm{Pois}(A_{ji};\, \pi)\, \mu + \mathrm{Pois}(A_{ij};\, M_{ij})\, \mathrm{Pois}(A_{ji};\, M_{ji})\, (1-\mu)}. \tag{29}$$

Convergence criteria

The EM algorithm consists of randomly initializing π, μ, u, v, w, then iteration of Eqs. 14–17, 22, 25, and 28, until the convergence of the following log-posterior,

$$L(\Theta) = \log P(\Theta|A) \geq \sum_{Z} q(Z) \log \frac{P(Z, \Theta|A)}{q(Z)} \tag{30}$$

$$= -\sum_{Z} q(Z) \log q(Z) + \sum_{Z} q(Z) \big\{ \log P(A|Z;\, \Theta) + \log P(Z|\mu) + \log P(\Theta) + \log P(\mu) \big\} \tag{31}$$

$$= -\sum_{Z} q(Z) \log q(Z) + \log P(\Theta) + \log P(\mu) \tag{32}$$

$$\begin{aligned} &+ \sum_{Z} q(Z) \left\{ \sum_{i,j} \big[Z_{i,j} \log \mathrm{Pois}(A_{i,j};\, \pi) + (1 - Z_{i,j}) \log \mathrm{Pois}(A_{i,j};\, M_{i,j}) + Z_{i,j} \log \mu + (1 - Z_{i,j}) \log(1-\mu) \big] \right\} \\ &= -\sum_{i<j} \big[Q_{i,j} \log Q_{i,j} + (1 - Q_{i,j}) \log(1 - Q_{i,j}) \big] \\ &\quad + \sum_{i,j} \big\{ Q_{i,j} \big(-\pi + A_{i,j} \log \pi \big) + (1 - Q_{i,j}) \big(-M_{i,j} + A_{i,j} \log M_{i,j} \big) \\ &\quad + Q_{i,j} \log \mu + (1 - Q_{i,j}) \log(1-\mu) \big\} + \mathit{const}, \end{aligned} \tag{33}$$

where we neglect *const*, constant term due to the uniform priors. To calculate $\log q(Z)$, we used Eq. (28), i.e., a Bernoulli distribution.

One can further add parameters' regularization, for instance by assuming Gamma-distributed priors for the membership vectors,

$$P(u_{ik};\, a, b) \propto u_{ik}^{a-1} e^{-b u_{ik}}, \tag{34}$$

where $a \geq 1$, to ensure the maximization of the log-likelihood (the second derivative must be negative), similarly for the v_{ik}. This would add new terms to the log-likelihood:

$$\begin{aligned} \mathcal{L}(\Theta) = L(\Theta) &+ (a-1) \sum_{i,k} \log u_{ik} - b \sum_{ik} u_{ik} \\ &+ (a-1) \sum_{i,k} \log v_{ik} - b \sum_{ik} v_{ik}. \end{aligned} \tag{35}$$

Alternatively, one can add constraints to the parameters, e.g. $\sum_k u_{ik} = 1$ (and similarly for v_i). This would modify the likelihood by adding the corresponding Lagrange multipliers.

Appendix 2: Generative model

Being generative, our model can be used to generate synthetic networks that include both anomalous edges and community structure. To this end, we sample the parameters (u, v, w, μ, π) and then, given these latent variables, we sample Z. Finally, given the Z and the latent variables, we can sample the adjacency matrix A.

For a given set of community parameters as the input [25, 30], we sample anomalous edges from a Poisson distribution as in Eq. (2), with a Bernoulli prior as in Eq. (1). The mean value of the Poisson distribution, π, is constant for all edges, however, its value can be chosen in order to control the ratio ρ_a of edges being anomalous over the total number of edges. The average number of anomalous and non-anomalous edges are $N^2\mu\,(1 - e^{-\pi})$, and $(1 - \mu)\,\sum_{i,j}(1 - e^{-M_{ij}})$, respectively. Assuming a desired total number of edges E, we can multiply π, μ and M_{ij} by suitable sparsity constants that tune: (i) the ratio

$$\rho_a = \frac{N^2\mu\,(1 - e^{-\pi})}{N^2\mu\,(1 - e^{-\pi}) + (1 - \mu)\,\sum_{i,j}(1 - e^{-M_{ij}})}, \quad \in [0, 1];$$

(ii) the success rate of anomalous edges π. Once these two quantities are fixed, the remaining sparsity parameter for the matrix M, is estimated as:

$$E\,(1 - \rho_a) = (1 - \mu)\,\sum_{i,j}(1 - e^{-cM_{ij}}), \tag{36}$$

which can be solved with root-finding methods.

Appendix 3: Performance in real-world datasets

Real data: dataset description

We tested our algorithm on three real-world datasets with the available ground truth communities. A brief overview of features of the studied datasets is presented in Table 4.

Real data: performance

Figure 8 shows how ACD can capture the anomalous edges with a satisfying accuracy, by tuning the parameters of our model, i.e., μ and π.

We apply ACD on a network with injected anomaly to estimate Q_{ij}, as the the expected value of Z_{ij} over the variational distribution (see "Appendix 1: Inference with expectation–maximization" section). The entries with the highest values are detected as anomalous edges, Fig. 9.

Table 4 Datasets description

Network	Abbreviation	N	E	Refs.
American college football	College football	115	613	[17]
Political blogs network	POLBLOGS	1490	19,090	[37]
Books about US politics	POLBOOKS	105	441	[28]
Zachary karate club	Zachary	34	78	[36]

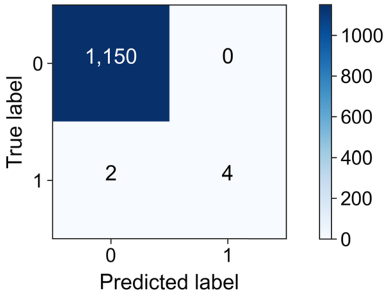


Fig. 8 The network of Zachary karate Club with injected edges (Experiment 1). Precision $= 1.0$, recall $= 0.667$, and F1 score $= 0.8$. Here $\pi = 0.25, \mu_0 = 0.5$

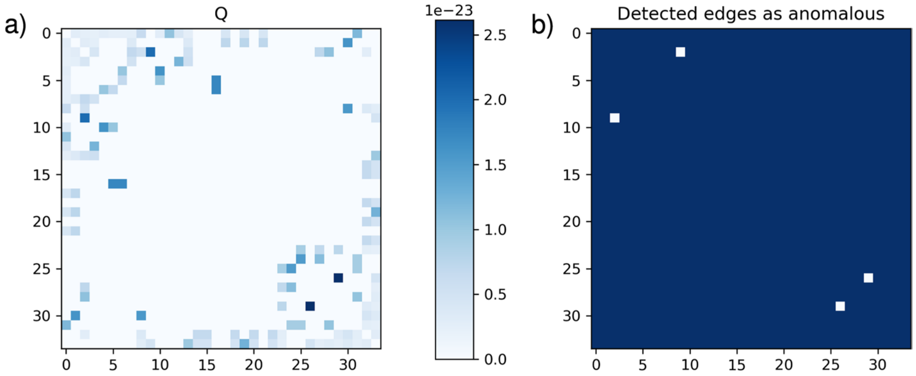


Fig. 9 The network of Zachary karate Club (Experiment 1). **a** Q matrix, estimated by ACD, as the expected value of Z_{ij} over the variational distribution. **b** The inferred anomalous edges, i.e., the entries of Q with the values above the threshold, here, the threshold $= 0.7 \times max(Q)$. Here, $\mu = 0.1, \pi = 10^{-4}$

Table 5 The confusion matrix for the network of POLBLOGS with injected edges (Experiment 1)

	π	Precision	Recall	F1
High recall/precision	0.3	0.49	0.99	0.65
High precision	0.5	0.45	0.27	0.34
High recall	0.8	0.40	0.69	0.51

We show how our model performs in terms of identifying anomalies—edges that have been injected in the dataset—as we vary the anomaly parameters, π. Here $\mu = 0.5$ and $\rho_a = 0.33$

Political blogs: To evaluate the efficiency and ability of our model when appiled to larger datasets, we used our model on the network of hyperlinks between weblogs on US politics [37]. This is a directed network with 1490 nodes, each belonging to one of two categories: conservative, or liberal. We run the Experiment 1 on this dataset by injecting some random anomalous edges, then applying the algorithm to detect those injected edges. Depending on the aim of the study, we can tune the value of the priors to have a higher precision or recall. Table 5 displays different regimes of performance with respect to the value of anomaly parameter, π (Fig. 10).

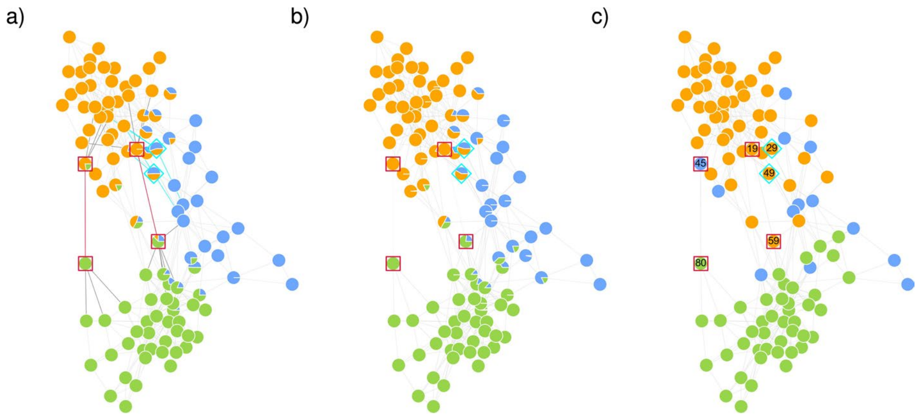


Fig. 10 The network of POLBOOKS (Experiment 2). **a** Soft community membership of nodes inferred by ACD. The edges inferred by ACD as anomalous edges are shown in red. Here, $\pi = 0.0001$, $\mu = 0.5$. **b** The communities inferred by CD, after removing the edges inferred as anomalous. **c** Ground truth communities. Red rectangles denote pairs of nodes that are connected by the edges inferred as anomalous. Cyan rectangles demonstrate the nodes that changed their community membership, after removing the aforementioned anomalous edges. Cyan edges in panel (**a**) present the edges connected to these

Acknowledgements
All the authors were supported by the Cyber Valley Research Fund.

Author contributions
HS and CDB designed the project, developed the code and analyzed the results. HS performed the experiments. HS and CDB wrote the final manuscript. Both authors read and approved the final manuscript.

Funding
Open Access funding enabled and organized by Projekt DEAL. All the authors were supported by the Cyber Valley Research Fund.

Availability of data and materials
The data analyzed in this study are available at [17, 28, 36]. The code to run the model is available in https://github.com/hds-safdari/Anomaly_Community_Detection.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 11 May 2022 Accepted: 29 November 2022
Published online: 22 December 2022

References

1. Hawkins DM. Identification of outliers (monographs on Statistics and applied probability), vol. 11. London: Chapman & Hall, Springer; 1980.
2. Hodge VJ, Austin J. A survey of outlier detection methodologies. Artif Intell Rev. 2004;22(2):85–126. https://doi.org/10.1007/s10462-004-4304-y.
3. Iliofotou M, Pappu P, Faloutsos M, Mitzenmacher M, Singh S, Varghese G. Network monitoring using traffic dispersion graphs (tdgs). In: Proceedings of the 7th ACM SIGCOMM conference on internet measurement. IMC '07. Association for computing machinery, New York, NY, USA. 2007. pp. 315–20. https://doi.org/10.1145/1298306.1298349.
4. Ding Q, Katenka NV, Barford P, Kolaczyk E, Crovella M. Intrusion as (anti)social communication: characterization and detection. In: KDD. 2012.
5. Ghosh S, Reilly D. Credit card fraud detection with a neural-network. In: 1994 Proceedings of the twenty-seventh Hawaii international conference on system sciences. 1994;3:621–30.

6. Agarwal D. An empirical bayes approach to detect anomalies in dynamic multidimensional arrays. In: Proceedings of the fifth IEEE international conference on data mining. ICDM '05. IEEE Computer Society, USA. 2005. pp. 26–33. https://doi.org/10.1109/ICDM.2005.22.
7. Solberg HE, Lahti A. Detection of outliers in reference distributions: performance of horn's algorithm. Clin Chem. 2005;51(12):2326–32.
8. Thottan M, Liu G, Ji C. In: Cormode G, Thottan M, editors. Anomaly detection approaches for communication networks. London: Springer; 2010. p. 239–61. https://doi.org/10.1007/978-1-84882-765-3_11.
9. Caruso C, Malerba D. A data mining methodology for anomaly detection in network data. In: Apolloni B, Howlett RJ, Jain L, editors. Knowledge-based intelligent information and engineering systems. Berlin: Springer; 2007. p. 109–16.
10. Catania CA, Bromberg F, Garino CG. An autonomous labeling approach to support vector machines algorithms for network traffic anomaly detection. Expert Syst Appl. 2012;39(2):1822–9. https://doi.org/10.1016/j.eswa.2011.08.068.
11. Subba B, Biswas S, Karmakar S. A neural network based system for intrusion detection and attack classification. In: 2016 twenty second National Conference on Communication (NCC), 2016;1–6 . https://doi.org/10.1109/NCC.2016.7561088.
12. Amaral AA, de Souza Mendes L, Zarpelão BB, Junior MLP. Deep ip flow inspection to detect beyond network anomalies. Comput Commun. 2017;98:80–96. https://doi.org/10.1016/j.comcom.2016.12.007.
13. Pang G, Shen C, Cao L, Hengel AVD. Deep learning for anomaly detection: a review. ACM Comput Surv. 2021. https://doi.org/10.1145/3439950.
14. Chen J, Sathe S, Aggarwal C, Turaga D. Outlier detection with autoencoder ensembles. 2017;90–8. https://doi.org/10.1137/1.9781611974973.11.
15. Hawkins S, He H, Williams G, Baxter R. Outlier detection using replicator neural networks. In: Kambayashi Y, Winiwarter W, Arikawa M, editors. Data warehousing and knowledge discovery. Berlin: Springer; 2002. p. 170–80.
16. Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. ACM Comput Surv. 2009;41(3):1–58.
17. Girvan M, Newman MEJ. Community structure in social and biological networks. Proc Nat Acad Sci. 2002;99(12):7821–6. https://doi.org/10.1073/pnas.122653799.
18. Fortunato S. Community detection in graphs. Phys Rep. 2010;486(3–5):75–174.
19. Prado-Romero MA, Gago-Alonso A. Community feature selection for anomaly detection in attributed graphs. In: Beltrán-Castañón C, Nyström I, Famili F, editors. Progress in pattern recognition, image analysis, computer vision, and applications. Cham: Springer; 2017. p. 109–16.
20. Gao J, Liang F, Fan W, Wang C, Sun Y, Han J. On community outliers and their efficient detection in information networks. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining. KDD 10. Association for computing machinery, New York, NY, USA. 2010. pp. 813–22.https://doi.org/10.1145/1835804.1835907.
21. Múller E, Sánchez PI, Mülle Y, Böhm K. Ranking outlier nodes in subspaces of attributed graphs. In: 2013 IEEE 29th International conference on data engineering workshops (ICDEW). 2013:216–22. https://doi.org/10.1109/ICDEW.2013.6547453.
22. Sultana N, Palaniappan S. A survey on online social network anomaly detection. Int J Innov Sci Res Technol. 2018;3(3):243–57.
23. Savage D, Zhang X, Yu X, Chou P, Wang Q. Anomaly detection in online social networks. Soc Netw. 2014;39:62–70. https://doi.org/10.1016/j.socnet.2014.05.002.
24. Bojchevski A, Günnemann S. Bayesian robust attributed graph clustering: joint learning of partial anomalies and group structure. In: Thirty-Second AAAI conference on artificial intelligence. 2018.
25. De Bacco C, Power EA, Larremore DB, Moore C. Community detection, link prediction, and layer interdependence in multilayer networks. Phys Rev E. 2017;95(4): 042317. https://doi.org/10.1103/PhysRevE.95.042317.
26. Goldenberg A, Zheng AX, Fienberg SE, Airoldi EM. A survey of statistical network models. Found Trends Mach Learn. 2010;2(2):129–233. https://doi.org/10.1561/2200000005.
27. Ball B, Karrer B, Newman ME. Efficient and principled method for detecting communities in networks. Phys Rev E. 2011;84(3): 036103.
28. Kunegis J. Konect: The koblenz network collection. In: Proceedings of the 22nd international conference on World Wide Web. WWW '13 Companion. Association for computing machinery, New York, NY, USA 2013. pp. 1343–50. https://doi.org/10.1145/2487788.2488173.
29. Gopalan P, Hofman JM, Blei DM. Scalable recommendation with hierarchical poisson factorization. In: UAI, 2015;326–35.
30. Contisciani M, Power EA, De Bacco C. Community detection with node attributes in multilayer networks. Sci Rep. 2020;10:15736. https://doi.org/10.1038/s41598-020-72626-y.
31. Newman ME, Clauset A. Structure and inference in annotated networks. Nat Commun. 2016;7(1):1–11.
32. Hric D, Peixoto TP, Fortunato S. Network structure, metadata, and the prediction of missing nodes and annotations. Phys Rev X. 2016;6(3): 031038.
33. Safdari H, Contisciani M, De Bacco C. Generative model for reciprocity and community detection in networks. Phys Rev Res. 2021;3: 023209. https://doi.org/10.1103/PhysRevResearch.3.023209.
34. Safdari H, Contisciani M, De Bacco C. Reciprocity, community detection, and link prediction in dynamic networks. J Phys Complex. 2022;3(1): 015010. https://doi.org/10.1088/2632-072X/ac52e6.
35. Contisciani M, Safdari H, De Bacco C. Community detection and reciprocity in networks by jointly modeling pairs of edges. Journal of Complex Networks 2022; 10(4):cnac034. https://doi.org/10.1093/comnet/cnac034
36. Zachary WW. An information flow model for conflict and fission in small groups. J Anthropol Res. 1977;33:452–73.
37. Adamic LA, Glance N. The political blogosphere and the 2004 u.s. election: divided they blog. In: Proceedings of the 3rd international workshop on link discovery. LinkKDD '05. Association for computing machinery, New York, NY, USA 2005. pp. 36–43. https://doi.org/10.1145/1134271.1134277.

Publisher's Note