Community detection and anomaly prediction in dynamic networks

Hadiseh Safdari[1, *] and Caterina De Bacco[1, †]

[1] *Max Planck Institute for Intelligent Systems, Cyber Valley, Tuebingen 72076, Germany*

Abstract

Anomaly detection is an essential task in the analysis of dynamic networks, as it can provide early warning of potential threats or abnormal behavior. We present a principled approach to detect anomalies in dynamic networks that integrates community structure as a foundational model for regular behavior. Our model identifies anomalies as irregular edges while capturing structural changes. Leveraging a Markovian approach for temporal transitions and incorporating structural information via latent variables for communities and anomaly detection, our model infers these hidden parameters to pinpoint abnormal interactions within the network. Our approach is evaluated on both synthetic and real-world datasets. Real-world network analysis shows strong anomaly detection across diverse scenarios. In a more specific study of transfers of professional male football players, we observe various types of unexpected patterns and investigate how the country and wealth of clubs influence interactions. Additionally, we identify anomalies between clubs with incompatible community memberships, but also instances of anomalous transactions between clubs with similar memberships. The latter is due in particular to the dynamic nature of the transactions, as we find that the frequency of transfers results in anomalous behaviors that are otherwise expected to interact as they belong to similar communities.

I.   Introduction

Dynamic networks are ubiquitous in real-world applications, ranging from social networks to biological systems and transportation networks [1, 2]. As these networks evolve over time, detecting anomalies in the networks becomes increasingly important for identifying critical events, predicting failures, and uncovering hidden patterns [3–5]. Anomaly detection in dynamic networks involves identifying objects (nodes, edges, subgraphs) that deviate significantly from the expected behavior of the network, given its past behavior. This process enables the detection of irregularities that may be indicative of important changes or abnormalities within the network structure or dynamics. For instance, network intrusions can result in confidential personal information being leaked to hackers. By utilising anomaly detection techniques, one can acquire a more profound understanding of the behavior and evolution of dynamic networks, allowing informed decision-making and necessary courses of action [6, 7].

Current approaches to detecting anomalies in dynamic graphs include machine-learning techniques, and graph-based algorithms [3, 8–11]. Although deep learning methods show potential in handling the complexity of graph data, they are not without limitations [3]. A significant obstacle is the models' lack of interpretability, which leads to challenges in understanding the reasoning behind anomaly detection decisions, an essential aspect in applications that require explanations.

Moreover, these models heavily rely on large labeled datasets for effective training, which becomes impractical in dynamic network environments due to the laborious and expensive process of data collection and labeling. Furthermore, the high computational complexity of deep learning models poses a barrier to real-time anomaly detection in dynamic networks, where timely identification is of utmost importance. In addition, many dynamic graph anomaly detection techniques in deep learning rely on models like DeepWalk [12], and GCN [13] that are primarily designed for static graphs. As a result, they do not effectively utilize evolving patterns in attributes for the detection task [3].

In addition to deep learning techniques, graph-based anomaly detection techniques, where nodes and edges represent objects and their relationships, have also been explored for anomaly edge detection in dynamic networks [3]. However, previous works in this area, which rely on statistical models, often face limitations in accurately identifying unknown anomalies and struggle with scalability issues in large networks. As the complexity and scale of dynamic networks continue to grow, there is an increasing need for principled approaches that can improve both the accuracy and efficiency of anomaly detection. Furthermore, the models in this category do not incorporate anomalies in a systematic way, as part of the data structure [14, 15].

---

* hadiseh.safdari@tuebingen.mpg.de

† caterina.debacco@tuebingen.mpg.de

In this paper, we introduce a principled approach that combines community detection and anomaly prediction, to learn patterns of regular behavior in the network structure and identify deviations, i.e., anomalous behavior. In many complex networks, nodes are clustered into communities of similar nodes [16–18]. For example, in transportation networks, communities might be clusters of closely connected airports, showing busy routes or regional hubs. Similarly, in online social networks, communities might be groups of users with shared interests or frequent interactions. These communities reveal how the network is organized and how its members interact. They help us understand the structure and functions of the underlying networks. Here, we exploit these insights to encode regular behavior in a probabilistic generative model for networks by means of community structure that evolve over time. By encoding deviations from this behavior as anomalies, we are able to estimate the probability that a certain edge is anomalous while also learning the hidden community structure.

The incorporation of community membership in this setting presents advantages over traditional anomaly detection methods [3, 8, 19]. For instance, by focusing on specific communities within the network, our approach can provide more targeted and accurate anomaly detection, as anomalies often manifest within localized regions rather than affecting the entire network. On the other hand, an evolving community structure can explain the underlying dynamics of the network effectively [20–22], enabling us to detect anomalies even in the absence of prior knowledge about specific anomaly types.

In this work we develop a probabilistic approach for modelling dynamic networks with anomalous edges, termed DynACD, where nodes remain fixed, and edges appear and disappear. Nodes have community memberships that determine the regular behavior of edge formation over time. The model estimates both community structure and the probability that an edge is anomalous.

We evaluate the effectiveness of our approach on real-world datasets and demonstrate its ability to accurately detect anomalous events in dynamic networks. By studying a dataset of players transfers between professional men football clubs, we find a variety of anomalous behaviors, ranging from anomalies appearing between clubs with compatible community structure but unusual frequency of transfers, or between clubs that have incompatible communities. Finally, we illustrate a potential application relevant for data collection procedures that require an initial preprocessing steps, showing how our model can be used to flag potential errors caused by mislabeling of nodes.

II.   Model: Dynamic DynACD

We observe pairwise interactions between individual nodes over time, which we represent using a time-dependent adjacency matrix $A_{ij}(t)$. In this work, we focus on binary directed temporal networks, meaning that $A_{ij}(t) = 1, 0$ indicates the existence of an edge from node i to j at time step t, or not, respectively.

To model the structure of temporal networks, we adopt a continuous-time Markov process approach similar to [23, 24]. We assume that the appearance and disappearance of edges are continuous events that occur on real-valued timescales. However, since our observations are discrete, we track the network at discrete time steps, $t = 0, 1, ..., T$.

We define a binary variable (or label) $Z_{ij} \in \{0, 1\}$ to classify anomalous edges: if $Z_{ij} = 1$, then the pair of edges $(i, j), (j, i)$ is anomalous, it is regular otherwise. This is a static quantity, as in this work, we focus on the scenarios where the status (or label) of a pair of nodes is either anomalous or not. We assume this not to change in time, while edges can appear and disappear based on their label. We also consider it to be symmetric, i.e., $Z_{ij} = Z_{ji}$, as we assume that an anomaly involves a given pair of nodes, regardless the direction of an edge (but $A_{ij}(t)$ is in general asymmetric, to model more generally directed networks). This label Z_{ij} is not known a priori, it is a latent variable that we need to learn from data, together with the other parameters of the model.

With this in mind, we express the likelihood of the existence of the edge (i, j) at time step $t = 0$, given a set of latent variables Θ as:

$$P(A_{ij}(0)|\Theta) = \text{Pois}(A_{ij}(0); \lambda_{ij}(0))^{1-Z_{ij}} \text{Pois}(A_{ij}(0); \pi)^{Z_{ij}} , \tag{1}$$

where π is a constant and the hyper-parameter $\mu \in [0, 1]$ regulates the prior distribution of Z,

$$Z_{ij} \sim \text{Bern}(\mu) . \tag{2}$$

In other words, at an initial time step $t = 0$, a Poisson distribution with mean π governs the probability of an anomalous edge at the initial time step. Instead, the existence of a regular edge between nodes i, j is dictated by a Poisson distribution with mean $\lambda_{ij}(0)$. This parameter depends only on the communities to which the nodes belong, according to a mixed-membership model [25]:

$$\lambda_{ij}(0) = \sum_{k,q} u_{ik} v_{jq} w_{kq}(0) \quad . \tag{3}$$

The K-dimensional vectors u_i and v_i represent the out-going and in-coming communities of nodes i, respectively. The $K \times K$ affinity matrix, with entries $w_{kq}(0)$ plays a crucial role in regulating the community structure of the network. When the diagonal entries exceed the off-diagonal entries, it promotes assortativity and increases the likelihood of edges connecting nodes within the same community. The number of communities K is a hyper-parameter that needs to be selected with model selection criteria. In our experiments here we use 5-fold cross-validation.

Given the probability of the network at the initial time step, and the labels of edges Z_{ij}, we now proceed with the evolution in time of $A_{ij}(t)$, $\forall t = 1, \ldots, T$. We consider a Markov process to approximate the trajectory in time as:

$$P(\{A(t)\}|\Theta) = \prod_{i,j} \left\{ P\left(A_{ij}(0)|\Theta\right) \times \prod_{t=1}^{T} \{P(A_{ij}(t)|A_{ij}(t-1), \Theta)\} \right\} \quad .$$

The dynamic is determined by the transition probability, that we assume factorized as

$$P(A_{ij}(t)|A_{ij}(t-1), \Theta) = [P_a(A_{ij}(t)|A_{ij}(t-1), \Theta)]^{Z_{ij}} \times [P_r(A_{ij}(t)|A_{ij}(t-1), \Theta)]^{1-Z_{ij}} \quad , \tag{4}$$

where the subscripts a, r denote the evolution of an anomalous or regular edge, respectively. In this work, we assume that at each time-step, regular edges appear with rate $\lambda_{ij}(t) = \sum_{k,q} u_{ik} v_{jq} w_{kq}(t)$, and disappear with rate γ, while anomalous edges appear and disappear with the rates ℓ and ζ, respectively. Hence, to fully build the time evolving probability of the node pairs, it becomes essential to estimate the transition rates that govern the appearance and disappearance of both regular and anomalous edges over time.

In line with the methodology employed by Zhang et al. [19, 24], we adopt a similar approach to compute the probability of edge existence. This involves solving a master equation that governs the evolution of the presence of edges, including both regular and anomalous ones (refer to Secs. S1 and S2 for more details).

By solving the master equation for the edge probability outlined in Eq.(S1), the transition rates specified in Eq. (S3) can be derived based on the model parameters denoted by $\Theta \equiv \{u, v, w, \beta, \mu, \pi, \phi, \ell\}$, see Eq. (S4). Hence, we are equipped to finalize the posterior computation for the edges, as articulated in Eq. (S5).

The goal of our study is to learn the hidden parameters of the model, Θ, based on the adjacency matrices observed at each time step $\{A(t)\}_{t=0}^{T}$. To accomplish this objective, we perform an inference task by maximizing the log-posterior of the parameters given the data (refer to Sec. S3 for additional details).

An important quantity that we are able to estimate in closed-form is the the posterior distribution of the labels $P(Z_{ij}|\{A_{ij}(t)\}_{t=0}^{T}, \theta)$, and in particular the expected value $\mathbb{E}[Z_{ij}] = Q_{ij} \in [0, 1]$. Other methods, e.g., approaches based on deep learning and embeddings, do not have a straightforward way to estimate this; hence, they need to make use of ad-hoc choices to classify anomalies a posteriori using proxies for this quantities (see Sec. S9 for a detailed discussion). In our case, the Z_{ij} are encoded explicitly in our model.

We made some main assumptions about the temporal dynamics of the model parameters. We consider the affinity matrix as a variable dependent on time, while keeping the community membership vectors u_i and v_i static over time. It is worth noting that an alternative perspective can be achieved by keeping w fixed and permitting changes in the community membership vectors over time [22]. Either of these approaches allow the community structure to evolve dynamically impacting edge formation in time via the parameter $\lambda_{ij}(t)$. Our model assumes a fixed number of communities K, yet employs a mixed-membership model to assign nodes to multiple communities with varying intensities. This allows to effectively capture the likelihood of the data by adjusting how an entry u_{ik} or v_{ik} impacts the magnitude of $\lambda_{ij}(t)$ via $w(t)$, while maintaining K constant. An overview of the algorithm to estimate the hidden parameters of the model is described in Algorithm 1.

Algorithm 1 DynACD: EM algorithm.

Input: network $A(t) = \{A_{ij}(t)\}_{i,j=1}^N$, $t = 0, \ldots, T$
  number of communities K.

Output: membership $u = [u_{ik}]$, $v = [v_{ik}]$; network affinity matrix $w(t) = [w_{kq}(t)]$; regular edge disappearance rate $\beta(t)$; mean value of Poisson anomaly distribution π; prior on anomaly indicator μ; anomalous edge disappearance rate ϕ, and ℓ.

Initialize $u, v, w(t), \beta, \mu, \pi, \phi, \ell$ at random.

Repeat until $\mathcal{L}$ converges:

  1. Calculate ρ and $\boldsymbol{Q}$ (E-step):

$$\rho_{ij}(t) = \frac{u_{ik}v_{jq}w_{kq}(t)}{\sum_{k,q} u_{ik}v_{jq}w_{kq}(t)} \ ,$$

$$Q_{ij} \sim \text{as in Eq.}(S30) \ .$$

  2. Update parameters Θ (M-step):

    i) for each node i and community k update memberships:

$$u_{ik} = \frac{\sum_{j,q} [1 - Q_{ij}] \sum_{t=0}^T \rho_{ijkq}(t) \, \hat{A}_{ij}(t)}{\sum_{j,q} v_{jq} [1 - Q_{ij}] \sum_{t=0}^T \hat{\beta}(t) \, w_{kq}(t)} \ ,$$

$$v_{ik} = \frac{\sum_{j,q} [1 - Q_{ij}] \sum_{t=0}^T \rho_{ijkq}(t) \, \hat{A}_{ji}(t)}{\sum_{j,q} u_{jq} [1 - Q_{ij}] \sum_{t=0}^T \hat{\beta}(t) \, w_{kq}(t)} \ ,$$

    ii) for each pair (k, q) update affinity matrix:

$$w_{kq}(t) = \frac{\sum_{i,j} [1 - Q_{ij}] \rho_{ijkq} \, \hat{A}_{ij}(t)}{\sum_{i,j} [1 - Q_{ij}]\hat{\beta}(t) \, u_{ik} \, v_{jq}} \ ,$$

    iii) update prior on anomaly indicator:

$$\mu = \frac{1}{N(N-1)/2} \sum_{i<j} Q_{ij} \ , \tag{5}$$

    iv) update anomaly parameters:

$$\pi = \frac{\sum_{i,j} Q_{ij} \hat{A}_{ij}(0)}{\sum_{i,j} Q_{ij}} \ , \tag{6}$$
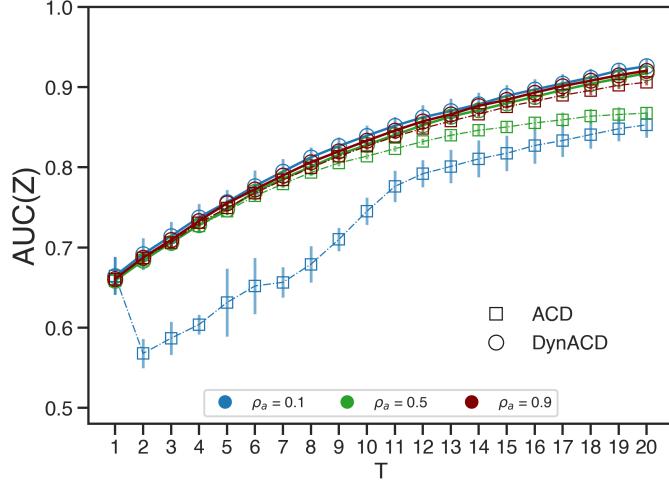
$$\ell = \frac{\sum_{i,j} \sum_{t=1}^T Q_{ij} \, \hat{A}_{ij}(t)}{T \sum_{i,j} \phi \, Q_{ij}} \ , \tag{7}$$

    v) The disappearance rate of regular edges β has no closed-form update as in Eq. (S25) ,
  vi) The disappearance rate of anomalous edges ϕ has no closed-form update as in Eq. (S26) .

III.   Results

A.   Synthetic Networks

As the primary objective of this study is to develop an algorithm for anomaly detection, it is essential to evaluate the predictive capability of the DynACD algorithm in identifying anomalous edges over time. To this end, we generate synthetic networks, using our model, and evaluate its performance in detecting anomalies in a situation where we know the ground truth anomalies. These are non-zero entries of the matrix Z used to generate the data. We utilize the matrix Q with entries Q_{ij} (refer to Sec. S3), representing the model's estimations of anomaly score for each edge.

FIG. 1: **Anomaly detection in synthetic networks.** The AUC(Z) metric quantifies the model's ability to distinguish between regular and anomalous edges. The synthetic network has $N = 300$ nodes, average degree $\langle k \rangle = 8$, and $K = 8$ communities of equal-size unmixed group membership generated with our generative model. Here, $\beta = 0.2$, $\ell = 0.2$, $\phi = 0.2$. Lines are averages and standard deviations over 10 sampled networks. DynACD ($\circ$), and ACD on aggregated dataset ($\square$).

We consider the area under the receiver operating curve (AUC) as prediction performance evaluation metric. A higher AUC score signifies enhanced performance in accurately identifying true anomalies. We are interested in particular in assessing how performance is enhanced by exploiting the dynamic character of the data. To this end, we compare the performance of DynACD against an algorithm developed for anomaly detection in static networks, ACD [19]. ACD does not accept dynamical data as input. Instead, at each time step, we apply ACD to the aggregated dataset, i.e., the sum over the time dimension. As a consequence, ACD predicts anomalies in a dataset where the temporal information is lost.

In both cases, we observe a discernible increase in the model's anomaly detection capabilities as we increase the number of time steps, T, which is given as input to the algorithm, as shown in Fig. 1. However, DynACD achieves significantly higher AUC values when the density of anomalies is low ($\rho_a = 0.1$); hence there are few positive examples of anomalies for a model to detect. For a balanced density ($\rho_a = 0.5$), performance is similar for a low number of time steps, however, subsequently the static model tends to plateau, while DynACD keeps increasing AUC, as more time steps are given in the input. This behavior is further observed, although to a lesser extent, for more extreme density of anomalies ($\rho_a = 0.9$). These results highlight the advantage of using the temporal information. When density of anomalies is low, the dynamical model can better recover them. When density is higher, the dynamical model better benefits of an increasing number of time steps given in input.

Further quantitative examination of DynACD and comparison with DynCD for synthetic networks in additional settings can be found in Sec. S6.

B. Detecting anomalies in real world dynamical networks

We study temporal real-world networked datasets characterized by a wide spectrum of nodes and edges, originating from diverse contexts such as trading (Transfermarkt), transportation (US-Air), and social networks (UCI Messages), see Sec. S7 for a detailed overview.

Our goal is to detect anomalous edges in these different datasets. As in these cases, the true labels for anomalous or regular edges are unavailable, we utilize the common strategy of injecting random edges into the dataset [8, 26] and label them as anomalous. Then, we run our anomaly detection algorithm and evaluate the model's ability to identify anomalies. Specifically, we inject n random edges between nodes within a given anomaly density range, ρ_a. Afterwards, we apply our algorithm to the modified dataset, calculating the expected value $\mathbb{E}[Z_{ij}] = Q_{ij}$ for the edge labels, i.e., the probability that the edges between two nodes i, j are anomalous. Following this step, we assign labels to the edges. In this particular experiment, we identify the initial n pairs (i, j) with the highest Q_{ij} values as anomalous edges. As a performance metric, we calculate the recall, which measures the accuracy of the algorithm in correctly identifying True Positives; and the AUC, the ability to rank True anomalous edges higher than False anomalous ones. Higher values mean better classification performance in both cases. Notice that with this procedure, as we

fix the number of detected anomalous edges equal to the number n of injected, we have that recall is equal to precision.

We find a robust performance of DynACD in detecting anomalies across the three datasets, as shown in Fig. 2. We notice how recall increases with the fraction of injected anomalies ρ_a, reaching a plateau at around 0.8, depending on the dataset. Similarly, we find high values of AUC in all datasets, observing both increasing and decreasing trends with ρ_a, but even in the decreasing cases (UCI Messages) the performance stays above 0.95. These results confirm the robustness of DynACD in detecting anomalous edges in different scenarios.
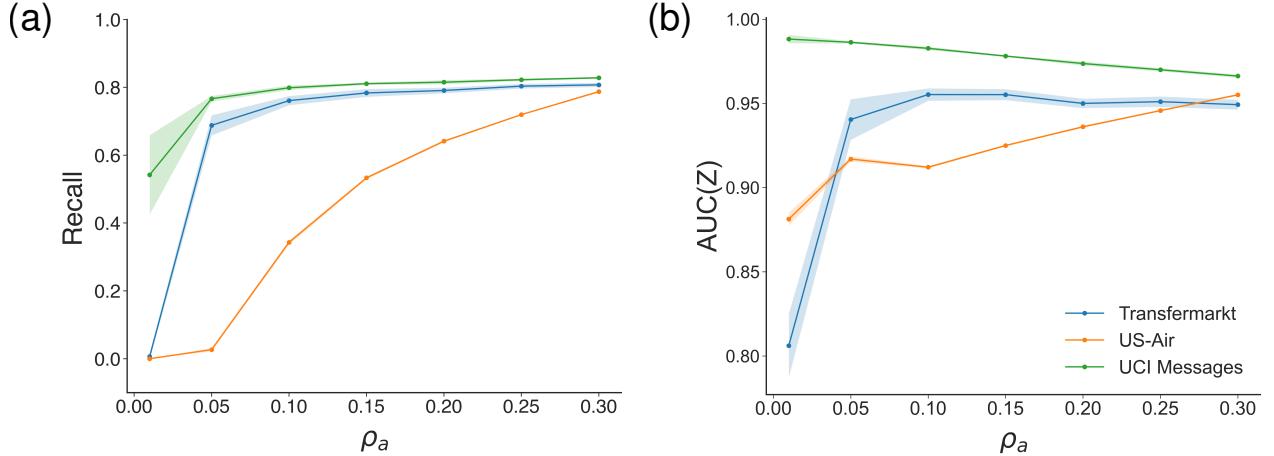

FIG. 2: **Anomaly Detection in the real-world datasets.** Performance of DynACD in detecting anomalies across real-world datasets. The graph illustrates increasing recall with the fraction of injected anomalies ρ_a, reaching a plateau at around 0.8, and consistently high AUC values above 0.95, confirming the algorithm's robustness in detecting anomalous edges across diverse scenarios. Lines and shades around them are averages and standard deviations over 10 sampled networks.

C. Analysis of football player transfers from Transfermarkt

To illustrate the applicability of our model, we run the algorithm on the Transfermarkt dataset, which contains information on player transfers among European men's football clubs. The dataset contains transfers involving eight different professional men's football leagues in Europe, see Sec. S7. Here nodes are clubs and directed edges $i \to j$ represent transfers from club i to club j. We focus on transfers made during the period from season 2008/2009 to 2022/2023. We have divided the dataset into five distinct time snapshots, each encompassing a four-year duration. We consider a network where $A_{ij}^t = 1, 0$ if there is at least a player transferred from i to j in period t or not.

To enrich the qualitative interpretation of our results, we utilize metadata for club categorization. This includes the transaction volume associated with player transfers into and out of clubs, representing a proxy for their financial wealth. Overall, we have a total of $N = 261$ clubs across eight leagues and $E = \sum_{i,j,t} A_{ij}^t = 7613$ transactions.

1. Community analysis

Our algorithm assumes that community membership is a primary driver of player transfers between football clubs, with the idea that one can identify main patterns of behaviors and thus cluster clubs in groups. These patterns can be influenced by different factors. For instance, we expect that the country of a club plays an important role, as this shapes the main market in which the club is embedded in. By analyzing the distribution of community membership we see that indeed country correlates well with the inferred partitions, see Fig. 5 for an example of the in-coming groups based on v. Indeed 4 out of 8 inferred groups are made mainly by clubs in the same countries (Italy, Netherlands,

England, and Russia). Other groups are more mixed, with a group made mainly of German and Spanish clubs. French and Portuguese clubs are split into two main communities and, to a lesser extent, into other groups. In one community, the country does not seem to play a role. It is a small community of 17 clubs that have a low magnitude in all the entries of v_i. These are small clubs that typically classify in the bottom part of the league table, have a low wealth category, and do not trade much in general. The main exception is RB Leipzig, which is a top European club. We argue that it is classified in this group because this dataset does not fully capture the pattern of player transfers that concerns this particular club. The Transfermarkt dataset contains only deals that involve clubs in either of the eight considered leagues. However, RB Leipzig is very active in markets outside these eight countries, hence our data may be limited in capturing the behavior of this club. Similar patterns are observed for the out-going communities based on u, see Fig. S3.

Finally, one can notice how top clubs tend to have a more mixed-membership as they trade more broadly across different markets, and their in-coming and out-going membership differs. For instance, Manchester City and Chelsea have in-coming communities that differ from those of the majority of English clubs, while their out-going ones are more aligned with their country. This signals that, while they tend to sell players more often to other English clubs, they tend to instead acquire players from other markets.

2. Club wealth and anomalous edges

In addition to communities, our algorithms naturally output estimates of anomalous edges. We analyze how these edges are distributed between clubs based on their wealth. We extract three tiers based on the total transfer fees spent in the years covered by the dataset by each club. Specifically, we label a club's wealth as "high" when this number is higher than 800 M€; "low" if is lower than 150 M€; and "average" if it is in between these two extremes. Then, we measure how regular and anomalous edges are distributed based on the wealth of the in-coming and out-going club. In Fig. 4, we notice a significantly higher fraction of anomalous edges involving wealthier clubs ("high") at both ends. Specifically, compared to the clubs engaged in regular edges, there is a notable increase in the number of transfers from wealthier clubs to both wealthier (0.119 for anomalous edges vs. 0.044 for regular edges), and average wealth (0.127 for anomalous edges vs. 0.094 for regular edges) clubs. Conversely, we notice a higher fraction of transfers between low wealth clubs in regular edges, suggesting that is less common to observe an anomalous edge between clubs of lower wealth.

3. Interplay between anomalous edges and communities

In the previous paragraphs, we focused on describing communities and anomalies separately. However, our model combines them to explain the observed transactions in time between clubs. Hence, here we investigate how these two factors together contribute to explaining edge formation, and illustrate various concrete scenarios where their interplay can guide practitioners in their analysis.

Anomalies between clubs with compatible communities

The previous analysis studied coarse-grained patterns of anomalies based on club wealth. Having access to individual estimates of Q_{ij}, the expected value of an edge to be anomalous, we can also isolate fine-grain patterns described by the behavior of individual clubs. One such pattern is that involving clubs in the same community. While we would usually expect more edges to exist between nodes with higher λ_{ij}^t, as dictated by their community memberships, we can still observe high Q_{ij}. This is because anomalies may also stem from unusual dynamical patterns, i.e., unusual distribution in time of transfers than we would otherwise expect between clubs with high λ_{ij}^t. One example of such a case is that of transfers involving pairs of Italian clubs, Genoa and Juventus and Genoa and Inter Milan, as shown in Table I. Comparing the distribution of transfers in time between these clubs and those involving other pairs with similar community memberships and λ_{ij}^t, we can attribute anomalies to the frequent appearance and disappearance of transfers between these specific clubs, impacting the expected value of the anomalous parameter, as detailed in Eqs. (S30-S31). We notice in fact that, while we expect transfers between Genoa and Juventus or Inter Milan to happen, we observe a higher number than expected, in both directions over time. In other words, it is not the presence of transfers between these clubs that is anomalous, but their frequency. This can also be seen in Fig. 3, where we isolate all the transfers involving Genoa and notice how Juventus and Inter Milan are the only Italian clubs that exchange players with them consistently across time. This pattern is inherent to the dynamic nature of the data and

could not be naturally captured by static models for anomaly detection.

Anomalies between clubs with incompatible communities

A different pattern of anomaly is that involving clubs with low λ_{ij}^t. When several edges exist between pairs of clubs in a way that is not expected by their community memberships (as measured by a low λ_{ij}^t), our model flags also these types of transfers as anomalous. A notable example is a set of edges involving Udinese and Watford, an Italian and an English club, respectively. Both clubs have memberships u_i and v_i peaked in one community (the one of mainly Italian and mainly English clubs, respectively). From this estimate, we expect the two clubs to transfer players mainly with other clubs in the same country. However, we observe an unusual high volume of transfers between these two clubs, something that is flagged as anomalous by our model, see Table S3. As the total number of transfers ($A_{ij}(t) = 7$) is not an anomaly in general (it is observed in several other pairs of clubs that are estimated as regular), we can attribute the high estimated Q_{ij} to the fact that it is a high number for clubs in these two communities specifically. We should remark that Udinese and Watford shared ownership in the years tracked by the dataset, which could explain the unusually high number of transactions. Note that this information was not given in input to the algorithm. Instead, observing a high value of Q_{ij} prompted further investigation into the relationship between these two clubs, which illustrates a practical usage of this model.

A more extreme case under this scenario is when some edges are observed between clubs with $\lambda_{ij}^t = 0$. Depending on what time step the edges are observed, this could imply the extreme value $Q_{ij} = 1$, see Eq. S30. In this case, it may be enough to observe one player transferred between two clubs in very incompatible communities to flag the edge as anomalous, for instance transfers between NEC Nijmegen and CA Osasuna in Table S3.

Case study: mislabeling in data collection.

Having discussed different types of anomalies captured by our model, we now illustrate relevant case scenarios that apply more broadly when collecting a dataset. For this, we manipulate the dataset by merging the clubs AC Milan and Inter Milan, two distinct Italian clubs with similar names. The high similarity in their name can cause confusion in data collection processes that start with a preprocessing step to clean node names to identify them uniquely. For instance, it is common to have several repetitions of the same node with slightly different names, due to errors or imprecisions in the data collection, particularly when node names come as string types (as opposed to integer numbers). This was a necessary step also in the Transfermarkt data, where many clubs appeared multiple times with slightly different string names. In our example, "Milan", "A.C. Milan", "AC Milan", and "Milan AC" refer all to the same club, whereas "Inter Milan", "FC Internazionale", "Inter" refer to another club; but "Inter Milan" can cause confusion. In our experiment, we merge the two clubs simulating a possible error caused by mislabeling clubs. As a result, we obtain one node to represent two clubs, and players transferred involving either of these two clubs are all assigned to that unique node "Milan". All the rest of the dataset is kept the same as before. We run the model on this manipulated data and obtain several anomalous edges as in the first case described above, i.e., high λ_{ij}^t and high Q_{ij}, as shown in Table S4. All these edges involve the "Milan" node and several other clubs (all Italian but Manchester City). The anomalies stem from the high frequency of transfers (in several cases $A_{ij}^t = T$, $\forall t = 1, \ldots, T$), which is higher than what is expected by communities alone. This high frequency of anomalies involving one particular club prompts practitioners to investigate further, and likely to identify the data collection problem.

TABLE I: **Transfermarkt dataset: frequency of transfers between Genoa and Juventus or Inter Milan**. Illustrative instances of player transfers between the clubs, characterized by a high expected value for tie formation based on community membership, here measured by a high average $\langle \lambda_{ij}^t \rangle = \sum_t \lambda_{ij}^t / T$. However, these interactions were flagged as anomalous (high Q_{ij}) due to higher rates of appearance.

Source Club (i)	Target Club (j)	$A_{ij}(0)$	$A_{ij}(1)$	$A_{ij}(2)$	$A_{ij}(3)$	$A_{ij}(4)$	$\sum_t A_{ij}(t)$	Q_{ij}	$\langle \lambda_{ij}^t \rangle$
Genoa	Juventus	1	1	1	1	1	5	0.425	0.376
Juventus	Genoa	1	1	0	1	1	4	0.425	0.483
Genoa	Inter Milan	1	1	1	1	1	5	0.357	0.340
Inter Milan	Genoa	1	0	1	1	1	4	0.357	0.427
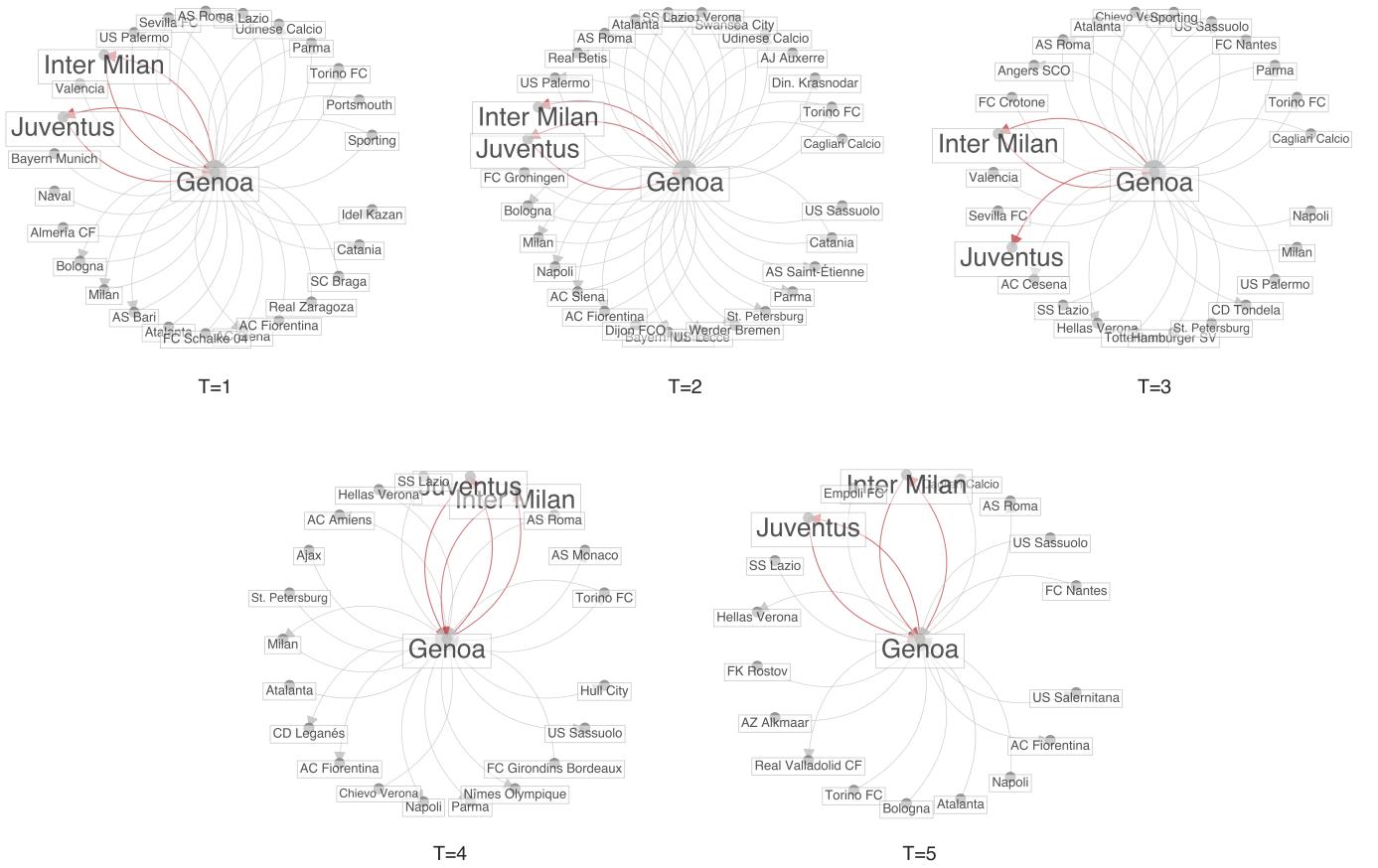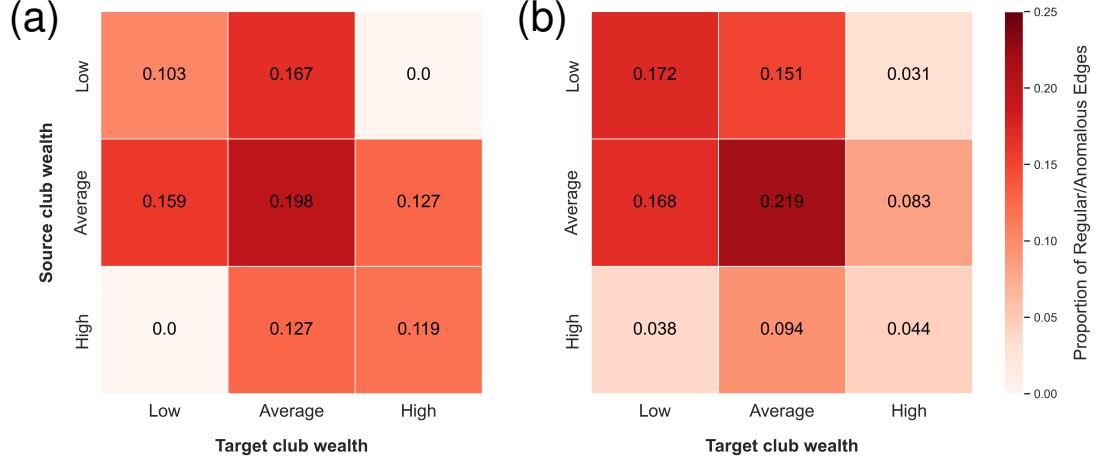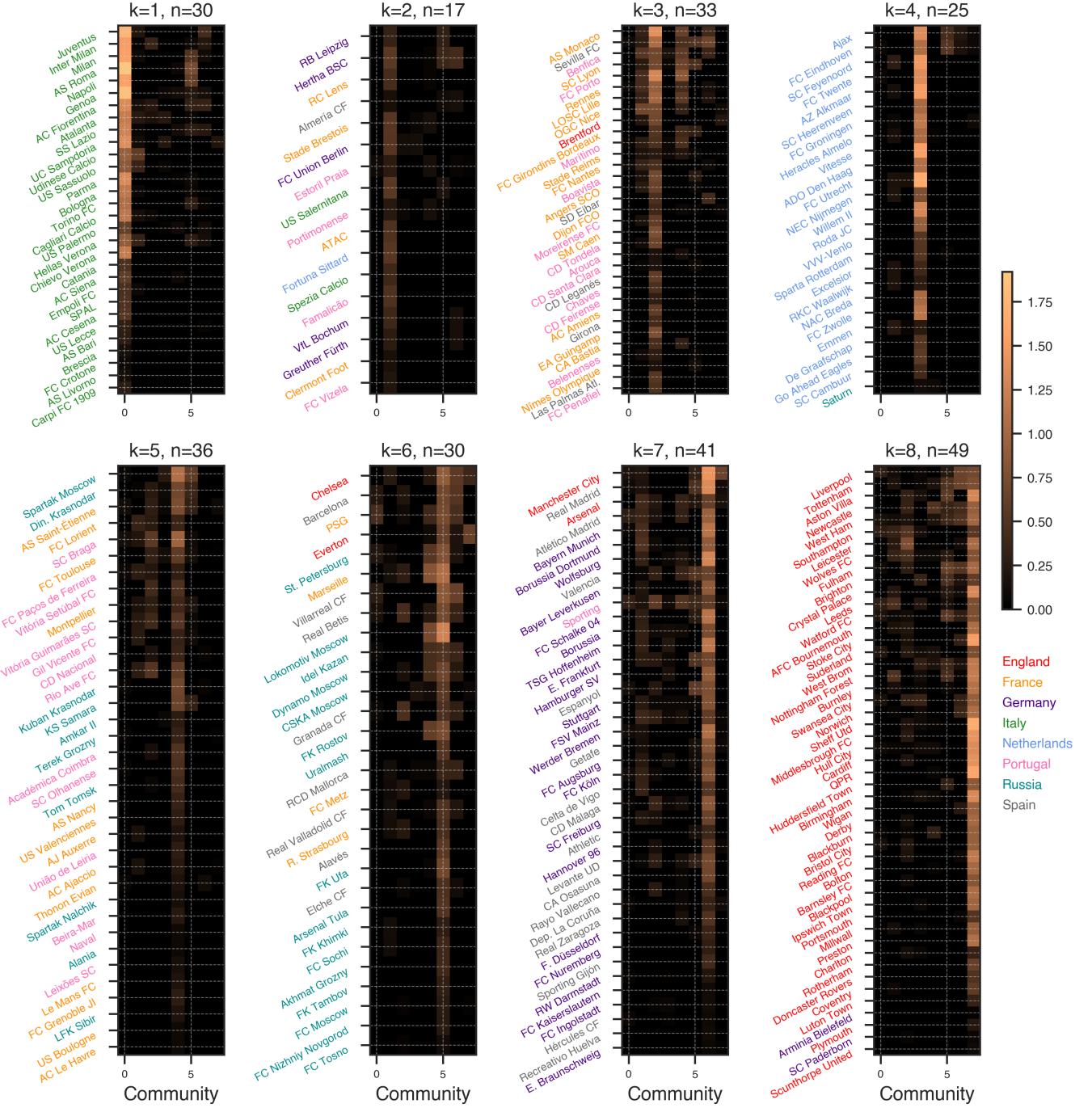
FIG. 3: **Transfermarkt datasets: Genoa transfer network**: Visualization of player transfers to and from Genoa involving various clubs at different time steps. Notably, there is a consistent presence of transfers with Juventus and Inter Milan at most time steps.

FIG. 4: **Transfermarkt datasets: Edges Distribution by wealth category of clubs.** Each cell in the heatmap represents the proportion of edges between clubs, distinguished by wealth categories (High, Average, and Low). We separate the set of edges considered in each heatmap between a) anomalous and b) regular, as estimated by DynACD. We normalize the heatmaps by the total number of edges considered in each plot, so that the sum over all entries in each heatmap equals 1. Darker color means higher proportion of edges exchanged between clubs in the given wealth categories.

FIG. 5: **Communities in the Transfermarkt dataset:** In-coming (soft) community membership v of clubs. The colors of the y-labels indicate the country to which the receiving clubs belong. This plot reveals an alignment between the community membership of clubs and their respective nationalities. The corresponding country of each league is shown in the legend on the right with the color assigned to that league.

IV.   Discussion

We have developed a principled probabilistic generative model that integrates mechanisms for the emergence of anomalous edges within a dynamic network. The model's core is rooted in the community memberships of nodes as the structural parameters that determine regular patterns. The network interactions governed by this mechanism are identified as the regular edges in the network. Instead, deviations for regularity are detected as anomalous.

One of the key advantages of our approach lies in its ability to provide targeted anomaly detection within specific communities of the network. Traditional anomaly detection methods often struggle to localize anomalies, treating the entire network homogeneously. In contrast, DynACD exploits the localized nature of anomalies, recognizing that deviations from regular behavior often occur within specific subsets of the network. This targeted approach enhances the accuracy and efficiency of anomaly detection, particularly in scenarios where anomalies manifest in localized regions rather than affecting the entire network.

Our approach models evolving network structures, allowing us to detect anomalies without prior knowledge. Empirical tests show the effectiveness of DynACD in spotting anomalous events in dynamic networks, particularly when the temporal aspect of the data is essential for understanding network structure.

Furthermore, the application of DynACD to the analysis of football player transfers illustrates its practical utility in real-world scenarios. By uncovering anomalous transfer patterns between football clubs, DynACD highlights potential irregularities that may signify mislabeling in data collection processes or unusual dynamics in player interactions. This case study highlights the importance of integrating domain knowledge with data-driven anomaly detection methods, which allows identifying and interpreting anomalies in complex networks.

Notably, our model is formulated on multiple assumptions. First, we assume community structure to capture regular patterns. Hence, our model may not be appropriate to model datasets where a community structure is not expected. Second, we assume that anomalies are static: although edges can appear and disappear in time their regular or anomalous nature does not change. Also in this case, if we expect an edge to change in nature from regular to anomalous, or vice versa, our model may not be appropriate. One can relax this assumption, for instance by making the variables Z_{ij} time-dependent. This would require further specifying a mechanism for their transition in time and an increased number of parameters, potentially leading to over-fitting and higher algorithmic complexity. Third, we assume a Markovian transition probability for edges to appear and disappear. If we expect significant time-correlations between past history beyond that one-step away, then we would need a model that is able to incorporate memory into the system.

While we have promising results in confirming the reliability and applicability of the proposed algorithm in real world scenarios, there are still several directions to extend the model. Integrating topological properties as reciprocity [27, 28] or triadic closure [29] could enhance its expressiveness in representation of real-world interactions. Exploring how to integrating both node and edge anomalies is a natural future direction, considering their coexistence in datasets. This could be potentially addressed by also incorporating extra information as node attributes [30–32]. In general, main challenges remain addressing the scalability of these method to handle large datasets and the ability to incorporate the intuition of experts and provide them interpretable results.

Acknowledgements

Supporting Information (SI)

S1.   Model Development

We follow an approach similar to that of Zhang et al. [19, 24] and calculate the probability of the existence of edges by solving a master equation. Defining $p_{ij,k}^{(s)}(t)$ as the probability of having k edges of type s, where s represents regular or anomalous connections, between nodes i and j at time t, we can express the evolution of this probability using the following master equation:

$$\frac{dp_{ij,k}^{(s)}(t)}{dt} = \Lambda_{ij}^{(s)}(t)\, p_{ij,k-1}^{(s)}(t) + (k+1)\Gamma^{(s)}\, p_{ij,k+1}^{(s)}(t) - \left(\Lambda_{ij}^{(s)}(t) + k\,\Gamma^{(s)}\right) p_{ij,k}^{(s)}(t), \tag{S1}$$

where $\Lambda_{ij}^{(s)}(t) = \lambda_{ij}(t)$ and $\Gamma^{(s)} = \gamma$ stand for regular edges, $(s = r)$, while $\Lambda_{ij}^{(s)}(t) = \ell$, and $\Gamma^{(s)} = \zeta$ for anomalous edges $(s = a)$. To solve this equation, we use a generating function approach [33] by defining $g^{(s)}(z,t) = \sum_{k=0}^{\infty} p_{ij,k}^{(s)}(t)z^k$. The solution for the generating function is given by [23]:

$$g^{(s)}(z,t) = f\left[(z-1)e^{-t\Gamma^{(s)}}\right] e^{\frac{(z-1)\Lambda_{ij}^{(s)}(t)}{\Gamma^{(s)}}}, \tag{S2}$$

which can be expanded in terms of z to give us $p_{ij,k}^{(s)}(t)$ (more details in Sec. S2). For each type of edges, anomalous or regular edge, there are four possible transitions from time $t-1$ to t: 1) there is no edge neither at time $t-1$, nor at t; 2) the appearance of an edge from non-edge, 3) disappearance of an existing edge, and 4) an existing edge remains; with the following probabilities, respectively,

$$
\begin{aligned}
p_{0\to0}^r &= e^{-\beta\times\lambda_{ij}(t)}, & p_{0\to0}^a &= e^{-\phi\times\ell} \\
p_{0\to1}^r &= \beta\times\lambda_{ij}(t)e^{-\beta\times\lambda_{ij}(t)}, & p_{0\to1}^a &= \phi\times\ell e^{-\phi\times\ell} \\
p_{1\to0}^r &= \beta\, e^{-\beta\times\lambda_{ij}(t)}, & p_{1\to0}^a &= \phi\, e^{-\phi\times\ell} \\
p_{1\to1}^r &= (1-\beta)e^{-\beta\times\lambda_{ij}(t)}, & p_{1\to1}^a &= (1-\phi)e^{-\phi\times\ell}
\end{aligned} \tag{S3}
$$

where $\beta = 1 - e^{-\gamma}$, and $\phi = 1 - e^{-\zeta}$. Hence, we can find the transition probability for regular and anomalous edge as follows,

$$
\begin{aligned}
P(A_{ij}(t)|A_{ij}(t-1),\Theta) &= [P_a(A_{ij}(t)|A_{ij}(t-1),\Theta)]^{Z_{ij}} \times [P_r(A_{ij}(t)|A_{ij}(t-1),\Theta)]^{1-Z_{ij}} \\
&= \left[[p_{0\to0}^a]^{(1-A(t-1))(1-A(t))}\, [p_{0\to1}^a]^{(1-A(t-1))A(t)}\, [p_{1\to0}^a]^{A(t-1)(1-A(t))}\, [p_{1\to1}^a]^{A(t-1)A(t)}\right]^{Z_{ij}} \\
&\times \left[[p_{0\to0}^r]^{(1-A(t-1))(1-A(t))}\, [p_{0\to1}^r]^{(1-A(t-1))A(t)}\, [p_{1\to0}^r]^{A(t-1)(1-A(t))}\, [p_{1\to1}^r]^{A(t-1)A(t)}\right]^{1-Z_{ij}} \\
&= \left[e^{-\phi\times\ell}\, [\phi\times\ell]^{(1-A(t-1))A(t)}\, [\phi]^{A(t-1)(1-A(t))}\, [(1-\phi)]^{A(t-1)A(t)}\right]^{Z_{ij}} \\
&\times \left[e^{-\beta\times\lambda_{ij}(t)}\, [\beta\times\lambda_{ij}(t)]^{(1-A(t-1))A(t)}\, [\beta]^{A(t-1)(1-A(t))}\, [(1-\beta)]^{A(t-1)A(t)}\right]^{1-Z_{ij}}.
\end{aligned} \tag{S4}
$$

Given the definition of the joint distribution in Eq. (4), this leads to the following time-dependent, log-posterior:

$$
\begin{aligned}
\mathcal{L}(T,\Theta) \quad &= \log[P(\{A(t)\}|\{A(t-1)\},\Theta)] \\[6pt]
&= -\sum_Z q(Z)\log q(Z) + \sum_{i,j}\sum_Z q(Z) \left\{ \log\left[[\mathrm{Pois}(A_{ij};\lambda_{ij}(0))\,(1-\mu)]^{(1-Z_{ij})}\,[\mathrm{Pois}(A_{ij};\pi)\,\mu]^{Z_{ij}} \right] \right. \\[6pt]
&\quad + \sum_{t=1}^{T}\log\left[[e^{-\beta\times\lambda_{ij}(t)}]^{\left(1-A_{ij}(t-1)\right)(1-A_{ij}(t))} \times [\beta\times\lambda_{ij}(t)]^{\left(1-A_{ij}(t-1)\right)A_{ij}(t)} \times \beta^{A_{ij}(t-1)\left(1-A_{ij}(t)\right)} \times (1-\beta)^{A_{ij}(t-1)A_{ij}(t)} \right]^{(1-Z_{ij})} \\[6pt]
&\quad + \left. \sum_{t=1}^{T}\log\left[[e^{-\phi\times\ell}]^{\left(1-A_{ij}(t-1)\right)(1-A_{ij}(t))} \times [\phi\times\ell]^{\left(1-A_{ij}(t-1)\right)A_{ij}(t)} \times \phi^{A_{ij}(t-1)\left(1-A_{ij}(t)\right)} \times (1-\phi)^{A_{ij}(t-1)A_{ij}(t)} \right]^{Z_{ij}} \right\}. \quad (\text{S5})
\end{aligned}
$$

S2. Master Equation

To solve the master equation in Eq. S1, we multiply both sides by z^k and sum over k; then defining the generating function $g^{(s)}(z,t)=\sum_{k=0}^{\infty}p_k^{(s)}(t)z^k$, we get,

$$
\begin{aligned}
\frac{\partial g^{(s)}(z,t)}{\partial t} \quad &= \Lambda_{ij}^{(s)}(t)\,z\,g^{(s)}(z,t) + \Gamma^{(s)}\frac{\partial g^{(s)}(z,t)}{\partial z} - \Lambda_{ij}^{(s)}(t)\,g^{(s)}(z,t) - z\,\Gamma^{(s)}\frac{\partial g^{(s)}(z,t)}{\partial z} \\[6pt]
&= (z-1)\left[\Lambda_{ij}^{(s)}(t)\,g^{(s)}(z,t) - \Gamma^{(s)}\frac{\partial g^{(s)}(z,t)}{\partial z} \right]. \quad (\text{S6})
\end{aligned}
$$

By replacing time dependent $\Lambda_{ij}^{(s)}(t)$ in Eq. S6, by its expected value, we reach the following relation for $g^{(s)}(z,t)$,

$$
g^{(s)}(z,t) = f\left[(z-1)e^{-t\,\Gamma^{(s)}} \right] e^{\frac{(z-1)\Lambda_{ij}^{(s)}}{\Gamma^{(s)}}}, \quad (\text{S7})
$$

where $f(x)$ is any once-differentiable function of its argument satisfying $f(0)=1$, [23, 24]. We can assume that at $t=0$, there were no edges between two specific nodes, hence by setting $t=0$ in Eq. S7, we find $f(x)=e^{-\frac{x}{\Gamma^{(s)}}\Lambda_{ij}^{(s)}}$. For the regular edges, we assume the mean as $\frac{\Lambda_{ij}^{(s)}}{\Gamma^{(s)}}=\lambda_{ij}$. Therefore, at $t=1$,

$$
\begin{aligned}
g^{(r)}(z,1) \quad &= \exp\left\{ (z-1)(\lambda_{ij})(1-e^{-\gamma}) \right\} \\
&= \exp\left\{ \beta\,(z-1)(\lambda_{ij}) \right\}. \quad (\text{S8})
\end{aligned}
$$

S3. Inference

Our goal is, given these two mechanisms, i.e., edge community membership and anomalous edges, first to determine the values of the latent parameters, $\Theta \equiv \{u,v,w(t),\beta,\mu,\pi,\phi,\ell\}$, which determine the relationship between the hidden label Z_{ij} and the data, and then, given those values, to estimate the label Z_{ij} itself.

We have the posterior:

$$
P(Z,\Theta|A) = \frac{P(A|Z,\Theta)P(Z|\mu)P(\Theta)P(\mu)}{P(A)}. \quad (\text{S9})
$$

Summing over all the possible labels we have:

$$
P(\Theta|A) = \sum_Z P(Z,\Theta|A), \quad (\text{S10})
$$

which is the quantity that we need to maximize to extract the optimal Θ. It is more convenient to maximize its

logarithm, as the two maxima coincide. We use the Jensen's inequality:

$$\log P(\Theta|A) = \log \sum_Z P(Z, \Theta|A) \geq \sum_Z q(Z) \log \frac{P(Z, \Theta|A)}{q(Z)} \quad, \tag{S11}$$

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

$$q(Z) = \frac{P(Z, \Theta|A)}{\sum_Z P(Z, \Theta|A)} \quad, \tag{S12}$$

this definition is also maximizing the right-hand-side of Eq. (S11) w.r.t. q. We then need to maximize it with respect to Θ to get the latent variables. This can be done in an iterative way using Expectation-Maximization, alternating between maximizing w.r.t. q using Eq. (S12) and then maximzing Eq. (S11) w.r.t. Θ.

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

$$\mathcal{L}(T, \Theta) = \log[P(\{A(t)\}|\{A(t-1)\}, \Theta)]$$

$$= - \sum_Z q(Z) \log q(Z) + \sum_{i,j} [Q_{ij} \log \mu + (1 - Q_{ij}) \log(1 - \mu) + Q_{ij} [-\pi + A_{ij}(0) \log \pi]]$$

$$+ \sum_{i,j} \left\{ (1 - Q_{ij}) [-\lambda_{ij} + A_{ij}(0) \log \lambda_{ij}] + \sum_{t=1}^T \left\{ (1 - Q_{ij}) [-\beta \lambda_{ij}(t) + [(1 - A_{ij}(t-1))A_{ij}(t) + A_{ij}(t-1)(1 - A_{ij}(t))] \log \beta \right. \right.$$

$$+ (1 - A_{ij}(t-1))A_{ij}(t) \log \lambda_{ij}(t) + A_{ij}(t-1)A_{ij}(t) \log(1 - \beta)] + Q_{ij} [-\phi \ell + (1 - A_{ij}(t-1))A_{ij}(t) \log \ell]$$

$$\left. \left. + Q_{ij} [[(1 - A_{ij}(t-1))A_{ij}(t) + A_{ij}(t-1)(1 - A_{ij}(t))] \log \phi + A_{ij}(t-1)A_{ij}(t) \log(1 - \phi)] \right\} \right\} \quad. \tag{S13}$$

To deal with the summation in the log, we again apply Jensen's Inequality,

$$\log \beta \sum_{k,q} u_{ik} v_{jq} w_{kq} \geq \sum_{k,q} \rho_{ijkq} \log \beta \, u_{ik} v_{jq} w_{kq} - \sum_{k,q} \rho_{ijkq} \log \rho_{ijkq} \quad. \tag{S14}$$

The equality will be established when,

$$\rho_{ijkq} = \frac{u_{ik} v_{jq} w_{kq}}{\sum_{k,q} u_{ik} v_{jq} w_{kq}} \quad. \tag{S15}$$

It leads to the log-posterior as a function of the probability distributions,

$$\mathcal{L}(T,\Theta) = \log[P(\{A(t)\}|\{A(t-1)\},\Theta)] =$$

$$-\sum_{Z} q(Z)\,\log q(Z) + \sum_{i,j}\left[Q_{ij}\log\mu + (1-Q_{ij})\log(1-\mu)\right]$$

$$+\sum_{i,j}\left\{(1-Q_{ij})\left[-\sum_{k,q}u_{ik}^{(0)}v_{jq}^{(0)}w_{kq}(0) + A_{ij}(0)\left(\sum_{k,q}\rho_{ijkq}\log u_{ik}^{(0)}v_{jq}^{(0)}w_{kq}(0) - \sum_{k,q}\rho_{ijkq}^{(0)}\log\rho_{ijkq}^{(0)}\right)\right]\right.$$

$$+Q_{ij}\left[-\pi + A_{ij}(0)\log\pi\right]$$

$$+\sum_{t=1}^{T}\left\{(1-Q_{ij})\left[-\beta\sum_{k,q}u_{ik}v_{jq}w_{kq}(t) + (1-A_{ij}(t-1))A_{ij}(t)(\log\beta + \sum_{k,q}\rho_{ijkq}\log u_{ik}v_{jq}w_{kq}(t) - \sum_{k,q}\rho_{ijkq}\log\rho_{ijkq})\right.\right.$$

$$+A_{ij}(t-1)(1-A_{ij}(t))\log\beta + A_{ij}(t-1)A_{ij}(t)\log(1-\beta)\right] + Q_{ij}\left[-\phi\ell + (1-A_{ij}(t-1))A_{ij}(t)\log\ell\right]$$

$$\left.\left.+Q_{ij}\left[[(1-A_{ij}(t-1))A_{ij}(t) + A_{ij}(t-1)(1-A_{ij}(t))]\log\phi + A_{ij}(t-1)A_{ij}(t)\log(1-\phi)\right]\right\}\right\} \quad . \tag{S16}$$

We initiate the process by taking the derivative of Eq. (S16), concerning the individual parameters, such as u_{ik}. We make the assumption of a uniform prior with respect to Θ, but we can readily integrate more intricate selections if required. Furthermore, we examine static values for u and v, while considering dynamic variations for w. We define $\hat{A}_{ij}(t) = A_{ij}(t)(1-A_{ij}(t-1))$ when $t>0$, $\hat{A}_{ij}(0) = A_{ij}(0)$, and $\hat{\beta}(t) = 1$ for $t=0$, and $\hat{\beta}(t) = \beta$ for $t>0$.

$$\frac{\partial\mathcal{L}(T,\Theta)}{\partial u_{ik}} = 0 \quad , \tag{S17}$$

$$\sum_{j}\left\{[1-Q_{ij}]\left[-\sum_{q}v_{jq}w_{kq} + \sum_{q}\rho_{ijkq}\hat{A}_{ij}(0)\frac{1}{u_{ij}}\right] + \sum_{t=0}^{T}\left([1-Q_{ij}]\left[-\beta\sum_{q}v_{jq}w_{kq} + \hat{A}_{ij}(t)\times\sum_{q}\rho_{ijkq}\frac{1}{u_{ij}}\right]\right)\right\} = 0 \quad . \tag{S18}$$

This yields:

$$u_{ik} = \frac{\sum_{j,q,t}[1-Q_{ij}]\,\rho_{ijkq}(t)\,\hat{A}_{ij}(t)}{\sum_{j,q}v_{jq}[1-Q_{ij}]\sum_{t=0}^{T}\hat{\beta}(t)\,w_{kq}(t)} \quad . \tag{S19}$$

We find similar expression for v_{ik} and w_{kq}.

$$v_{ik} = \frac{\sum_{j,q,t}[1-Q_{ij}]\,\rho_{ijkq}(t)\,\hat{A}_{ji}(t)}{\sum_{j,q}u_{jq}[1-Q_{ij}]\sum_{t=0}^{T}\hat{\beta}(t)\,w_{kq}(t)} \quad , \tag{S20}$$

$$w_{kq}(t) = \frac{\sum_{i,j}[1-Q_{ij}]\,\rho_{ijkq}\,\hat{A}_{ij}(t)}{\sum_{i,j}[1-Q_{ij}]\hat{\beta}(t)\,u_{ik}\,v_{jq}} \quad . \tag{S21}$$

Following the same procedure, we find the expressions for μ and ℓ:

$$\mu = \frac{1}{N(N-1)/2}\sum_{i<j}Q_{ij} \quad , \tag{S22}$$

$$\pi = \frac{\sum_{i,j}Q_{ij}\,\hat{A}_{ij}(0)}{\sum_{i,j}Q_{ij}} \quad , \tag{S23}$$

$$\ell = \frac{\sum_{i,j} \sum_{t=1}^{T} Q_{ij}\, \hat{A}_{ij}(t)}{T \sum_{i,j} \phi\, Q_{ij}} \quad . \tag{S24}$$

The static parameters β and ϕ lack closed-form updates. Hence, to determine their values, we need to employ numerical root-finding methods on their respective equations:

$$\sum_{i,j} \sum_{t=1}^{T} [1 - Q_{ij}] \left\{ -\beta \left[\lambda_{ij}(t) + \frac{1}{1-\beta} A_{ij}(t-1) A_{ij}(t) \right] + \left[\hat{A}_{ij}(t) + A_{ij}(t-1)(1 - A_{ij}(t)) \right] \right\} = 0 \quad , \tag{S25}$$

and

$$\sum_{i,j} \sum_{t=1}^{T} Q_{ij} \left\{ -\phi \left[\ell + \frac{1}{1-\phi} A_{ij}(t-1) A_{ij}(t) \right] + \left[\hat{A}(t) + A_{ij}(t-1)(1 - A_{ij}(t)) \right] \right\} = 0 \quad . \tag{S26}$$

To estimate $q(Z)$, we insert the estimated parameters into Eq. (S9):

$$q(Z) = \frac{\prod_{i,j} \left[\mathrm{Pois}(A_{ij}(0);\pi) F_a(\phi,\ell,A_{ij},T) \right]^{Z_{ij}} \left[\mathrm{Pois}(A_{ij}(0);\lambda_{ij}(0)) F_r(\beta,\lambda_{ij}(t),A_{ij},T) \right]^{1-Z_{ij}} \prod_{i<j} \mu^{Z_{ij}} (1-\mu)^{(1-Z_{ij})}}{\sum_Z \prod_{i,j} \left[\mathrm{Pois}(A_{ij}(0);\pi) F_a(\phi,\ell,A_{ij},T) \right]^{Z_{ij}} \left[\mathrm{Pois}(A_{ij}(0);\lambda_{ij}(0)) F_r(\beta,\lambda_{ij}(t),A_{ij},T) \right]^{1-Z_{ij}} \prod_{i<j} \mu^{Z_{ij}} (1-\mu)^{(1-Z_{ij})}}$$

$$= \prod_{i<j} \frac{\left[\mathrm{Pois}(A_{ij}(0);\pi) \mathrm{Pois}(A_{ji}(0);\pi)\, F_a(\phi,\ell,A_{ij},T) F_a(\phi,\ell,A_{ji},T)\, \mu \right]^{Z_{ij}}}{\sum_{Z_{ij}=0,1} \left[\mathrm{Pois}(A_{ij}(0);\pi) \mathrm{Pois}(A_{ji}(0);\pi)\, F_a(\phi,\ell,A_{ij},T) F_a(\phi,\ell,A_{ji},T)\, \mu \right]^{Z_{ij}}} \tag{S27}$$

$$\times \prod_{i<j} \frac{\left[\mathrm{Pois}(A_{ij}(0);\lambda_{ij}(0)) \mathrm{Pois}(A_{ji}(0);\lambda_{ji}(0))\, F_r(\beta,\lambda_{ij}(t),A_{ij},T) F_r(\beta,\lambda_{ji}(t),A_{ji},T)\, (1-\mu) \right]^{(1-Z_{ij})}}{\sum_{Z_{ij}=0,1} \left[\mathrm{Pois}(A_{ij}(0);\lambda_{ij}(0)) \mathrm{Pois}(A_{ji}(0);\lambda_{ji}(0))\, F_r(\beta,\lambda_{ij}(t),A_{ij},T) F_r(\beta,\lambda_{ji}(t),A_{ji},T)\, (1-\mu) \right]^{(1-Z_{ij})}} \tag{S28}$$

$$= \prod_{i<j} Q_{ij}^{Z_{ij}} (1 - Q_{ij})^{(1-Z_{ij})} \quad , \tag{S29}$$

where

$$Q_{ij} =$$

$$\frac{\mu\, \mathrm{Pois}(A_{ij}(0);\pi) \mathrm{Pois}(A_{ji}(0);\pi) \times F_a(\phi,\ell,A_{ij},T)\, F_a(\phi,\ell,A_{ji},T)}{\mu\, \mathrm{Pois}(A_{ij}(0);\pi) \mathrm{Pois}(A_{ji}(0);\pi) F_a(\phi,\ell,A_{ij},T) F_a(\phi,\ell,A_{ji},T) + (1-\mu) \mathrm{Pois}(A_{ij}(0);\lambda_{ij}(0)) \mathrm{Pois}(A_{ji}(0);\lambda_{ji}(0)) F_r(\beta,\lambda_{ij}(t),A_{ij},T) F_r(\beta,\lambda_{ji}(t),A_{ji},T)}, \tag{S30}$$

and

$$F_r(\beta,\lambda_{ij},A_{ij},T) = \prod_{t=1}^{T} \mathrm{e}^{-\beta \times \lambda_{ij}(t)} \left[[\beta\, \lambda_{ij}(t)]^{(1-A_{ij}(t-1))A_{ij}(t)} [\beta]^{A_{ij}(t-1)(1-A_{ij}(t))} [1-\beta]^{A_{ij}(t-1)A_{ij}(t)} \right] \tag{S31}$$

$$F_a(\phi,\ell,A_{ij},T) = \prod_{t=1}^{T} \mathrm{e}^{-\phi \times \ell} \left[[\phi\, \ell]^{(1-A_{ij}(t-1))A_{ij}(t)} \phi^{A_{ij}(t-1)(1-A_{ij}(t))} [1-\phi]^{A_{ij}(t-1)A_{ij}(t)} \right]. \tag{S32}$$

S4. Covergence criteria

The EM algorithm involves the random initialization of $\Theta \equiv \{u, v, w(t), \beta, \mu, \pi, \phi, \ell\}$, followed by iterations over Eqs. S19-S24, and S30, until achieving convergence of the log-posterior in Eq. (S13). For the computation of $\log q(Z)$, we employ Eq. (S29), representing a Bernoulli distribution.

S5. Generative model

Being generative, our model can be used to generate synthetic networks that include both anomalous edges and community structure over time. To this end, we sample the parameters $\Theta \equiv \{u, v, w(t), \beta, \mu, \pi, \phi, \ell\}$ and then, given these latent variables, we sample Z. Finally, given the Z and the latent variables, we can sample the adjacency matrix A.

To build the adjacency matrix at the first time step, i.e., $T = 0$, and for a given set of community parameters as the input [25, 26], we sample anomalous edges from a Poisson distribution as in Eq. (1), with a Bernoulli prior as in Eq. (2). 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 $N\mu(1 - e^{-\pi})$, and $(1 - \mu)\sum_{i,j}(1 - e^{-\lambda_{ij}(0)})$, respectively. Assuming a desired total number of edges E, we can multiply π, μ and $\lambda_{ij}(0)$ by suitable sparsity constants that tune: i) the ratio

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

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

$$E(1 - \rho_a) = (1 - \mu)\sum_{i,j}(1 - e^{-c\lambda_{ij}(0)}) \quad , \tag{S33}$$

which can be solved numerically with root-finding methods.

To construct adjacency matrix at the next time steps, we use the transition rates from Sec. S1.

S6. Results on synthetic datasets

a. Experimental settings for synthetic networks We consider networks with hard communities of equal size and assortative structure. We vary the anomaly density within the range $\rho_a \in [0, 1]$ and set the disappearance rate of anomalous and regular edges to $\phi = 0.2$ and $\beta = 0.2$, respectively. For comparison, we establish a baseline model, termed DynCD, which simplifies our model to standard community detection for dynamic networks without any anomaly detection mechanism. This is obtained by setting $\mu = 0$ and $\lim \ell \to 0$, which are kept fixed as hyper-parameters in the inference task. Having fixed the parameters, we generate 10 samples of networks for each value of ρ_a. For each network, we generate an initial state followed by up to T further snapshots. The initial state is formed based on the the community structure and anomaly density outlined in Eq. (1). The successive snapshots are generated according to transition rates in Eq. (S3), as explained in Sec. S5.

b. Community detection As the presented algorithm hinges on incorporating the community structure as the regular model for tie formation between nodes, an important aspect is to demonstrate the competence of the model in accurately retrieving the community structure. The results presented in Fig. S1(a) demonstrate the effectiveness of our proposed anomaly-community detection algorithm, DynACD, compared to the model without an anomaly detection mechanism, DynCD. The plot shows the cosine similarity (CS) between the ground truth and inferred communities for synthetic networks with $N = 300$ nodes, average degree $\langle k \rangle = 8$, and $K = 8$ communities, and total number of time stamps $T = 20$.

The results indicate that DynCD performs similar to DynACD for small values of anomaly density (ρ_a). Nevertheless, as the number of edges without community structure increases, the algorithm's ability in community detection decreases significantly, aligning with expectations. On the other hand, DynACD, which includes an anomaly detection mechanism, is greatly impacted by the number of time steps. Despite its weaker performance in community detection during the initial time steps, it exhibits continuous improvement over time. This demonstrates the effectiveness of the proposed anomaly detection mechanism in improving the overall community detection accuracy.

c. Link prediction performance In addition, we assess the performance of DynACD in the link prediction task, comparing it with DynCD. For each time step $t \in [1, T]$, we hide an individual snapshot $A(t)$ (test set), and fit the data using the previous snapshots $A(0), \ldots, A(t - 1)$ (training set). We test whether the model is able to predict the network's evolution using the area under the curve (AUC) calculated on the test set, which gauges the likelihood that a randomly chosen edge possesses a higher expected value than a randomly selected non-existent edge. A perfect reconstruction is indicated by a value of 1, while 0.5 denotes random chance. To compute the expected value of a
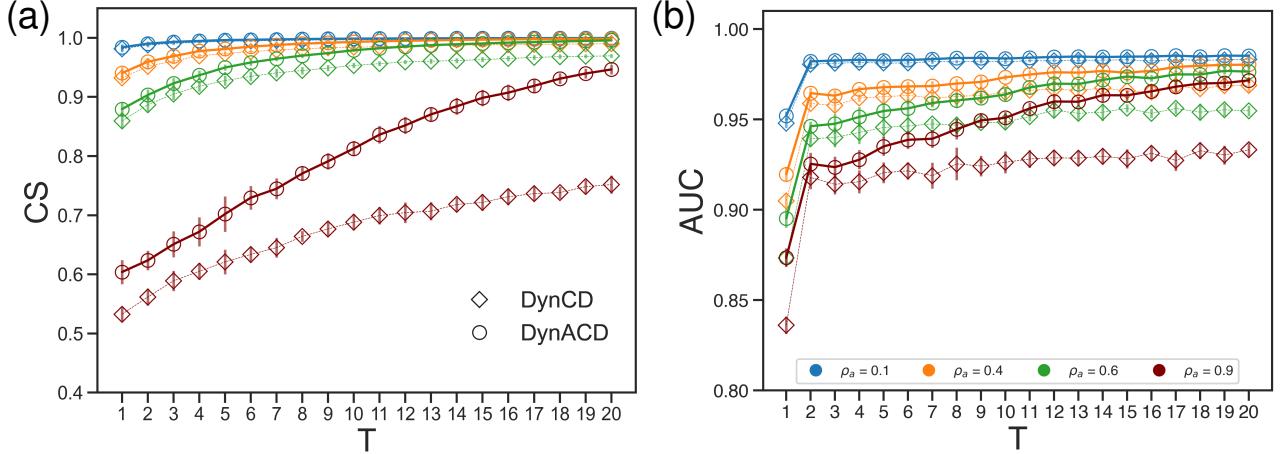
FIG. S1: **Community detection and Link Prediction in synthetic networks.** a) Cosine similarity (CS) between ground truth and inferred communities. The model without anomaly detection mechanism, DynCD, performs significantly for small values of anomaly density, ρ_a. However, by increasing ρ_a, i.e., more edges without community structure, the algorithm's ability in community detection decreases, as it is expected. b) DynACD exhibits a clear and consistent increase in the estimated (AUC) over time. Synthetic networks were generated with $N = 300$ nodes, average degree $\langle k \rangle = 8$, and $K = 8$. In addition the model parameters were fixed to $\beta = 0.2$, $\ell = 0.2$, $\phi = 0.2$. Lines are averages and standard deviations over 10 sampled networks. Anomaly: True ($\circ$), and False ($\diamond$).

regular edge, Eq. (S34) is employed:

$$\mathbb{E}\left[A_{ij}(t)\right] \;=\; \begin{cases} \frac{\beta \,\lambda_{ij}(t)}{1+\beta \,\lambda_{ij}(t)} & \text{if } A_{ij}(t-1) = 0 \\ 1 - \beta & \text{if } A_{ij}(t-1) = 1 \end{cases}. \tag{S34}$$

Similarly, the expected value of an anomalous edge is computed using Eq. (S35):

$$\mathbb{E}\left[A_{ij}(t)\right] \;=\; \begin{cases} \frac{\phi \,\ell_{ij}}{1+\phi \,\ell_{ij}} & \text{if } A_{ij}(t-1) = 0 \\ 1 - \phi & \text{if } A_{ij}(t-1) = 1 \end{cases}. \tag{S35}$$

It is important to note that while the expected value at time t relies only on the network at the preceding time step, all parameters are inferred using the entire network history (excluding the final time step, i.e. test set). Therefore, our model is trained with the complete set of snapshots $A(0), \ldots, A(t-1)$. In general, both DynACD and DynCD demonstrate reasonable performance in link prediction. However, DynCD reaches a plateau in its performance over time. In contrast, DynACD exhibits a clear and consistent increase in the estimated AUC over time, as illustrated in Fig. S1(b). Notably, while the estimated AUC values by DynACD converge to a maximum over time across all ranges of anomaly density, ρ_a, DynCD exhibits lower AUC values, particularly for higher values of ρ_a. In other words, as the number of anomalous edges increases, DynCD loses its effectiveness in link prediction.

d. Estimating parameters Considering that DynACD is a generative model, it has the capability to learn the model parameters, $\Theta \equiv \{u, v, w(t), \beta, \mu, \pi, \phi, \ell\}$, from the provided dataset. These parameters enable the algorithm to generate synthetic datasets with statistical features mirroring those of the original dataset. Therefore, accurate inference of the model parameters is crucial for the model's proficiency. Table S1 presents the estimated values of the latent variables in synthetic data as we vary the density of anomalous edges ρ_a, illustrating the ability of the algorithm in retrieving values similar to the ground truth value.

S7. REAL-WORLD DATA SETS: DATASET DESCRIPTION

Table S2 provides a summary of the key characteristics of the studied datasets. The datasets of US-Air and Transfermarkt have undergone a preprocessing step that involved the splitting the edges in time snapshots, the removal of self-loops, retaining only nodes with both non-zero incoming and outgoing edges, and keeping only the

TABLE S1: **Estimated latent variables** Synthetic networks with $N = 300$ nodes, average degree $\langle k \rangle = 8$, and $K = 8$ communities of equal-size unmixed group membership generated with our generative model. Ground truth values are set to: $\beta = 0.2$, $\ell = 0.2$, $\phi = 0.2$. We report estimated averages and standard deviations over 10 samples for each of the main model's parameters.

ρ_a	π	ϕ	ℓ	β
0.1	0.2434 ± 0.0220	0.2091 ± 0.0078	0.2151 ± 0.0094	0.1994 ± 0.0019
0.4	0.2360 ± 0.0117	0.2085 ± 0.0043	0.2072 ± 0.0057	0.1981 ± 0.0033
0.6	0.2285 ± 0.0070	0.2096 ± 0.0054	0.2044 ± 0.0079	0.1963 ± 0.0035
0.9	0.2249 ± 0.0088	0.2112 ± 0.0025	0.1966 ± 0.0046	0.1904 ± 0.0043

giant connected components.

TABLE S2: **Real-world datasets description.**

Network	Abbreviation	N	E	T	Ref.
Data on European football clubs' player transfers	Transfermarkt	261	7613	5	[34]
Network of United States Air Transportation	US-Air	867	8039	6	[35]
The UC Irvine messages network	UCI Messages	1810	18693	6	[36]

1. US Air Transport Networks
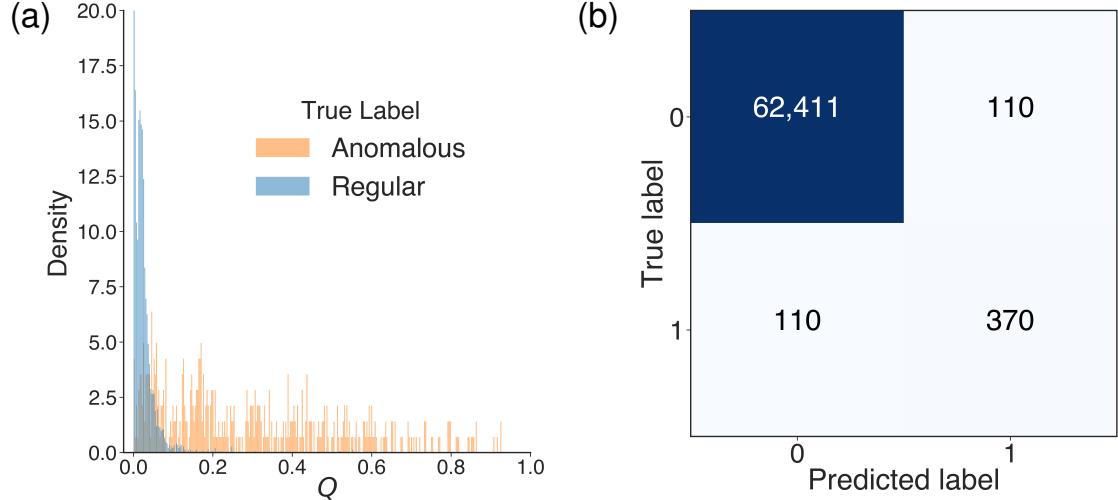
The dataset used in this study provides the dynamics of the United States air transportation network from 2007 to 2016. The original format of the dataset contains three different representations of nodes (airport, city, state) and edges (number of passengers, number of flights, amount of freight), resulting in a total of nine different networks. The dataset allows for a comprehensive analysis of the air transportation system, considering various perspectives and levels of granularity [35]. For this study, the we focused on the period from 2011 to 2016 and used airports as nodes. The edges represent the flights between airports, with the weight of the edges indicating the number of flights. This US Air Transport Networks contribute to a detailed understanding of the connectivity and activity within the air transportation network during the specified time frame.

2. Transfermarkt

The Transfermarkt dataset contains information on player transfers among eight European men football clubs starting from the 1992/93 season [34] . Transfermarkt is a dedicated website that offers a wide range of data on football transfers, market values, rumors, and statistics. The dataset is structured into separate files for different leagues, including the English Premier League, French Ligue 1, German 1.Bundesliga, Italian Serie A, Spanish La Liga, Portuguese Liga NOS, Dutch Eredivisie, and Russian Premier Liga. Key variables within the dataset include club name, player name, position, transfer fee, transfer movement, transfer period, league name, year, season, and country. Our analysis focuses on the dataset spanning the 2008/2009 season to the 2022/2023 season, and for algorithmic application, we segmented the dataset into five time snapshots, each spanning a four-year duration. Here, edges indicate transfers of at least one player between two clubs in a given time window.

3. UCI Messages

The UC Irvine messages network dataset is a directed network that contains sent messages between users in an online community of students from the University of California, Irvine [36]. The dataset consists of nodes that represent the users and directed edges that represent the sent messages.

FIG. S2: **Anomaly Detection in the Transfermarkt dataset with injected anomalous edges.** a) Anomaly Score Distribution (Q) Separated by True Labels. The histogram distinguishes between true labels, highlighting the density of scores for both "Anomalous" and "Regular" instances. b) The confusion matrix illustrates the classification performance of the model, breaking down the predicted and true labels for each class. The diagonal elements represent correctly classified instances, while off-diagonal elements indicate misclassifications. We achieve a recall rate of 0.77 and an AUC(Z) of 0.96 on the Transfermarkt dataset. Here, $\rho_a = 0.1$.

S8.   Results on REAL-WORLD DATA SETS

In this section, we present a detailed examination of the results obtained from anomaly detection experiments conducted on real-world datasets. Specifically, we analyze the modified Transfermarkt dataset with injected anomalous edges at a density of $\rho_a = 0.1$. The histogram in Sec. S8(a) provides a visual representation of the distribution of inferred anomaly scores (Q) assigned to edges by DynACD. It effectively discerns between true labels, emphasizing the density patterns of scores for both "Anomalous" and "Regular" instances. It is evident that edges labeled as "Regular" exhibit an inferred anomaly score (Q) with a peak around zero. In contrast, the inferred values of Q for injected edges—those labeled as "Anomalous"—extend across higher values, trending towards one. The confusion matrix in Sec. S8(b) also visualizes the classification performance of the model, particularly in distinguishing between anomalous and regular edges.

The results inferred by DynACD and presented in Table S3 align well with our expectations from the algorithm. Notably, we observe detailed specificity in the transfer information between Udinese Calcio and Watford FC, along with insights into the transfers involving NEC Nijmegen and CA Osasuna. In both cases, the expected value of observing a regular transfer between the two involved clubs is negligible, i.e., $\lambda_{ij} \to 0$. Therefore, it is reasonable to infer the existence of an irregular interaction between such clubs, i.e., $Q_{ij} = 1$. This observation highlights that not only can the consistency in the appearance and disappearance of transfers over time contribute to a higher value for Q_{ij}, as seen in the case of Udinese Calcio and Watford FC, but also a single-time transfer out of community membership can yield the same result, as the transfer between NEC Nijmegen and CA Osasuna.

During our experiments, in the data preprocessing, we aggregated the transfer data of two Italian clubs, Inter Milan and AC Milan. Surprisingly, the algorithm inferred a very high value for Q_{ij} for many of the interactions of the aggregated club, referred to as Milan. This observation stemmed from a higher rate of connections between the aggregated club and numerous others, showcasing DynACD's ability to detect irregularities. Examples of such inferred Q_{ij}s are provided in Table S4.

Fig. S3 highlights the same points as mentioned in Sec. III C. However, this plot illustrates the out-going (soft) community membership of clubs.

S9. Embedding methods from deep learning

In machine learning several approaches have been developed for anomaly detection in networks [3, 8]. Many of them rely on extra information on nodes, e.g., node attributes or features, which is outside the scope of the problem presented in our work. In scenarios where only the network structure is given as input, as is the case here, a common approach is to utilize embeddings. Similar to our memberships u and v, embeddings are vectors assigned to nodes. Like our model, these embeddings are then utilized to detect anomalous edges. The various approaches differ in the way these embeddings are learned. For instance, the Netwalk algorithm [8] employs techniques from deep learning.

The main difference between these approaches and the one considered in this paper is that our model postulates explicitly the presence of anomalous edges. This is because in our case, the probability of an edge is explicitly defined in terms of u and v (and the other parameters). As a consequence, we can obtain automatically (without further post-processing) and unambiguously (with the precise definition of Z_{ij} in terms of the latent variables) the estimate of this probability. In contrast, embedding methods do not formulate explicitly this probability. Instead, they make a broader assumption that embeddings explain the network structure, hence, one can use them a posteriori to determine what is an anomalous and what is a regular edge. A main consequence of this broader assumption is that one has to make arbitrary decisions in the post-processing to determine how the embeddings should be used to estimate the probability of an edge being anomalous.

One important arbitrary decision is how to transform a representation that is assigned to nodes (the embedding vectors u_i and v_j) into a representation on edges (i, j). For instance, Netwalk considers various functions to combine the node representations, with best results obtained with the Hadamard product $z_{ij} = u_i \odot v_j$ [3]. In our case instead, we have automatically the estimated Q_{ij} as a function of the inferred u and v, a quantity that has a clear interpretation as the expected value of the posterior of Z_{ij}.

In addition, once the edge representation z_{ij} is selected, one has to further decide how to use its inferred value to assign the probability for an edge to be anomalous. A standard approach is to run the k-means clustering algorithm on the edge representation vectors to cluster edges in a space (not to be confused with the node embedding space). Then, one computes the distance between each edge and its cluster centre and edges further away from the cluster are considered anomalous, with the distance used as a score to quantify the magnitude of being anomalous. All these steps though require further arbitrary decisions and parameters to be tuned, e.g. what clustering algorithm, the number of clusters in this new edge representation space, the exact mapping between the distance from their clusters or other clusters and the anomaly score. Further complexities are added in case of dynamic networks, where embeddings may change in time.
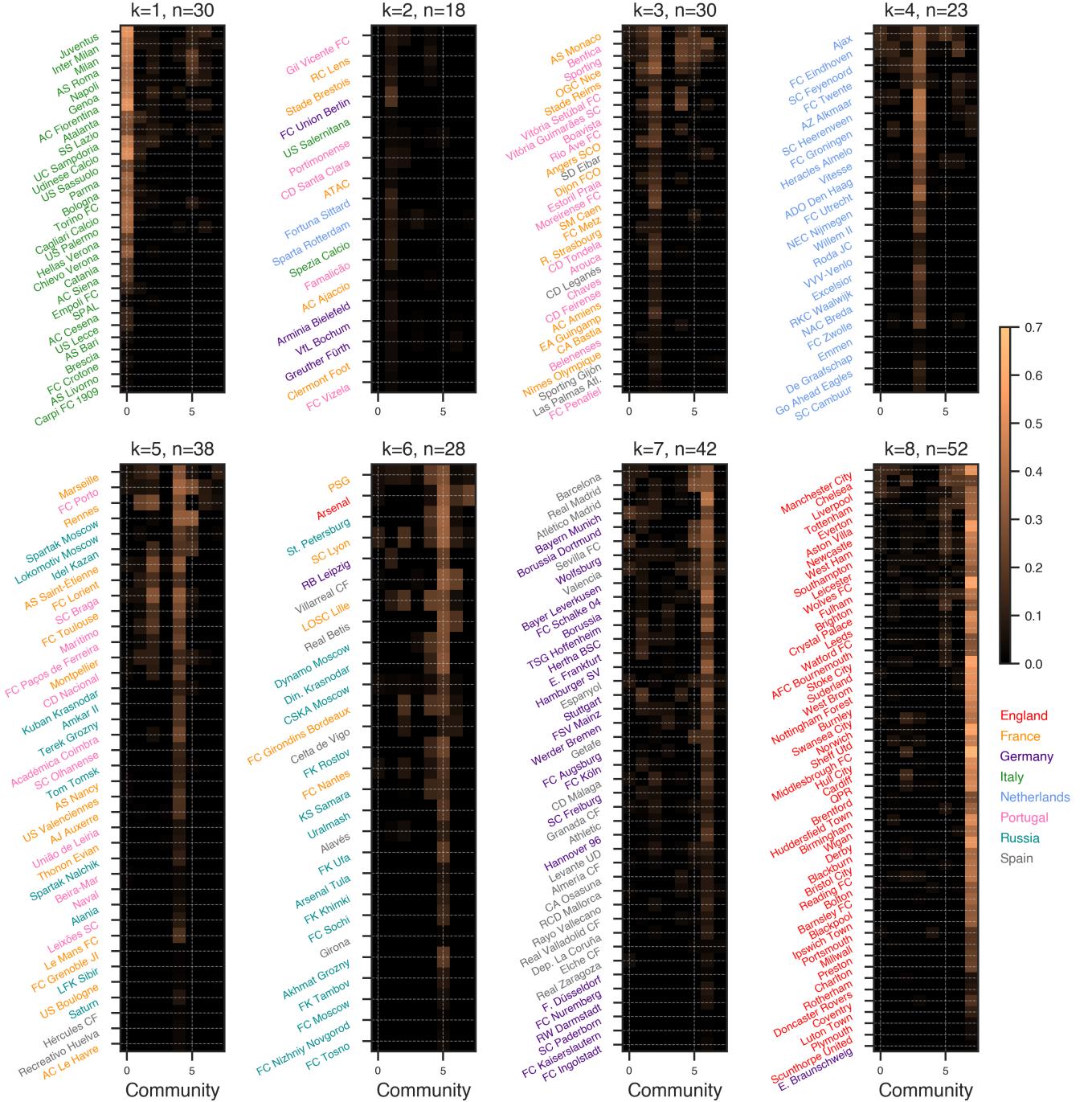
In the context of our work, we applied the Netwalk algorithm to real-world datasets studied. However, the performance of the Netwalk algorithm was found to be very poor when compared to DynACD. It is difficult to pinpoint exactly the cause of this, as any of the decisions above could impact performance. Consequently, while we provide a comprehensive discussion on the reasons behind these difference in performance, we refrain from detailing the specific results obtained from the application of the Netwalk algorithm to the datasets studied. More investigations would be needed to understand how to calibrate these models for a fair comparison, this is beyond the scope of our work.

TABLE S3: **Transfermarkt Dataset:** Instances of transfers between clubs exhibiting a low expected value for tie formation based on community membership. Consequently, these interactions are identified as irregular, regardless of whether they feature high or low appearance rates.

Source Club (i)	Target Club (j)	$A_{ij}(0)$	$A_{ij}(1)$	$A_{ij}(2)$	$A_{ij}(3)$	$A_{ij}(4)$	$\sum_t A_{ij}(t)$	Q_{ij}	$\langle \lambda_{ij}^t \rangle$
Udinese Calcio	Watford FC	0	1	1	1	1	4	1.0	0.049
Watford FC	Udinese Calcio	0	0	1	1	1	3	1.0	0.013
NEC Nijmegen	CA Osasuna	1	0	0	0	0	1	1.0	0.000
CA Osasuna	NEC Nijmegen	0	1	0	0	0	1	1.0	0.000

TABLE S4: **Transfermarkt Dataset with mislabeled nodes.** Instances of transfers flagged as anomalous in a manipulated dataset where Inter Milan and AC Milan are merged into a unique node "Milan" due to mislabeling in the data collection process. Several transfers involving "Milan" are flagged as anomalous, including transfers with other teams with compatible communities, as shown by a releatively high value of the average $\langle \lambda_{ij}^t \rangle = \sum_t \lambda_{ij}^t / T$.

| Source Club (i)	Target Club (j)	$A_{ij}(0)$	$A_{ij}(1)$	$A_{ij}(2)$	$A_{ij}(3)$	$A_{ij}(4)$	$\sum_t A_{ij}(t)$	Q_{ij}	$\langle \lambda_{ij}^t \rangle$
Milan	Genoa	1	1	1	1	1	5	0.996	0.300
Genoa	Milan	1	1	1	1	1	5	0.996	0.239
Milan	UC Sampdoria	1	1	1	1	1	5	0.966	0.197
UC Sampdoria	Milan	1	1	1	1	0	4	0.966	0.223
AS Roma	Milan	1	1	1	1	1	5	0.945	0.238
Milan	AS Roma	1	1	1	1	0	4	0.945	0.320
Parma	Milan	1	1	1	1	0	4	0.901	0.181
Milan	Parma	1	1	1	1	1	5	0.901	0.133
Manchester City	Milan	1	1	1	1	1	5	0.658	0.150
Milan	Manchester City	1	1	0	0	0	2	0.658	0.176

FIG. S3: **Transfermarkt Datasets:** Visualization of the out-going (soft) community membership u of clubs. The colors of the y-labels indicate the country to which the clubs belong. This plot reveals a clear alignment between the community membership of clubs and their respective nationalities. The corresponding country of each league is shown in the legend on the right with the color assigned to that league.

[1] Othon Michail and Paul G. Spirakis, "Elements of the theory of dynamic networks," Commun. ACM **61**, 72 (2018).

[2] Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial, "Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey," IEEE Access **9**, 79143–79168 (2021).

[3] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z. Sheng, Hui Xiong, and Leman Akoglu, "A comprehensive survey on graph anomaly detection with deep learning," IEEE Transactions on Knowledge and Data Engineering **35**, 12012–12038 (2023).

[4] Gongxun Miao, Guohua Wu, Zhen Zhang, Yongjie Tong, and Bing Lu, "Addag-ae: Anomaly detection in dynamic attributed graph based on graph attention network and lstm autoencoder," Electronics **12** (2023), 10.3390/electronics12132763.

[5] Stephen Ranshous, Shitian Shen, Danai Koutra, Steve Harenberg, Christos Faloutsos, and Nagiza F. Samatova, "Anomaly detection in dynamic networks: a survey," WIREs Computational Statistics **7**, 223–247 (2015), https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.1347.

[6] Sarah Oliveira Pinto and Vinicius Amorim Sobreiro, "Literature review: Anomaly detection approaches on digital business financial systems," Digital Business **2**, 100038 (2022).

[7] William Villegas-Ch, Jaime Govea, and Angel Jaramillo-Alcazar, "Iot anomaly detection to strengthen cybersecurity in the critical infrastructure of smart cities," Applied Sciences **13** (2023), 10.3390/app131910977.

[8] Wenchao Yu, Wei Cheng, Charu C. Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang, "Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18 (Association for Computing Machinery, New York, NY, USA, 2018) p. 2672–2681.

[9] Huan Wang, Jia Wu, Wenbin Hu, and Xindong Wu, "Detecting and assessing anomalous evolutionary behaviors of nodes in evolving social networks," ACM Trans. Knowl. Discov. Data **13** (2019), 10.1145/3299886.

[10] Panpan Zheng, Shuhan Yuan, Xintao Wu, Jun Li, and Aidong Lu, "One-class adversarial nets for fraud detection," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19 (AAAI Press, 2019).

[11] Li Zheng, Zhenpeng Li, Jian Li, Zhao Li, and Jun Gao, "Addgraph: Anomaly detection in dynamic graph using attention-based temporal gcn," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19* (International Joint Conferences on Artificial Intelligence Organization, 2019) pp. 4419–4425.

[12] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena, "Deepwalk: online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14 (Association for Computing Machinery, New York, NY, USA, 2014) p. 701–710.

[13] Thomas N. Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations* (2017).

[14] Dhivya Eswaran and Christos Faloutsos, "Sedanspot: Detecting anomalies in edge streams," in *2018 IEEE International Conference on Data Mining (ICDM)* (2018) pp. 953–958.

[15] Xiaolei Li, Zhenhui Li, Jiawei Han, and Jae Gil Lee, "Temporal outlier detection in vehicle traffic data," (2009), conference contribution.

[16] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt, "Stochastic blockmodels: First steps," Social Networks **5**, 109–137 (1983).

[17] Brian Ball, Brian Karrer, and M. E. J. Newman, "Efficient and principled method for detecting communities in networks," Phys. Rev. E **84**, 036103 (2011).

[18] Alec Kirkley and M. E. J. Newman, "Representative community divisions of networks," Communications Physics **5**, 40 (2022).

[19] Hadiseh Safdari and Caterina De Bacco, "Anomaly detection and community detection in networks," Journal of Big Data **9**, 122 (2022).

[20] Tiago P Peixoto and Martin Rosvall, "Modelling temporal networks with markov chains, community structures and change points," in *Temporal Network Theory* (Springer, 2019) pp. 65–81.

[21] Tiago P Peixoto and Martin Rosvall, "Modelling sequences and temporal networks with dynamic community structures," Nature communications **8**, 1–12 (2017).

[22] Catherine Matias and Vincent Miele, "Statistical clustering of temporal networks through a dynamic stochastic block model," Journal of the Royal Statistical Society: Series B (Statistical Methodology) **79**, 1119–1141 (2017).

[23] Hadiseh Safdari, Martina Contisciani, and Caterina De Bacco, "Reciprocity, community detection, and link prediction in dynamic networks," Journal of Physics: Complexity **3**, 015010 (2022).

[24] Xiao Zhang, Cristopher Moore, and Mark EJ Newman, "Random graph models for dynamic networks," The European Physical Journal B **90**, 1–14 (2017).

[25] Caterina De Bacco, Eleanor A Power, Daniel B Larremore, and Cristopher Moore, "Community detection, link prediction, and layer interdependence in multilayer networks," Physical Review E **95**, 042317 (2017).

[26] Hadiseh Safdari and Caterina De Bacco, "Anomaly detection and community detection in networks," Journal of Big Data **9**, 122 (2022).

[27] Hadiseh Safdari, Martina Contisciani, and Caterina De Bacco, "Generative model for reciprocity and community detection in networks," Phys. Rev. Research **3**, 023209 (2021).

[28] Martina Contisciani, Hadiseh Safdari, and Caterina De Bacco, "Community detection and reciprocity in networks by jointly modelling pairs of edges," Journal of Complex Networks **10**, cnac034 (2022).

[29] Tiago P Peixoto, "Disentangling homophily, community structure, and triadic closure in networks," Physical Review X **12**, 011004 (2022).

[30] Martina Contisciani, Eleanor A Power, and Caterina De Bacco, "Community detection with node attributes in multilayer networks," Scientific reports **10**, 15736 (2020).

[31] Mark EJ Newman and Aaron Clauset, "Structure and inference in annotated networks," Nature communications **7**, 11863 (2016).

[32] Oscar Fajardo-Fontiveros, Roger Guimerà, and Marta Sales-Pardo, "Node metadata can produce predictability crossovers in network inference problems," Physical Review X **12**, 011010 (2022).

[33] C. W. Gardiner, *Handbook of stochastic methods for physics, chemistry and the natural sciences*, 3rd ed., Springer Series in Synergetics, Vol. 13 (Springer-Verlag, Berlin, 2004) pp. xviii+415.

[34] `https://github.com/ewenme/transfers`.

[35] Batyr Charyyev, Mustafa Solmaz, and Mehmet Hadi Gunes, "Dynamic network of united states air transportation at multiple levels," in *Complex Networks XI*, edited by Hugo Barbosa, Jesus Gomez-Gardenes, Bruno Gonçalves, Giuseppe Mangioni, Ronaldo Menezes, and Marcos Oliveira (Springer International Publishing, Cham, 2020) pp. 282–293.

[36] "Uc irvine messages network dataset – KONECT," (2017).