Unfolding temporal networks through statistically significant graph evolution rules

Alessia Galdeman
Department of Computer Science
University of Milan
Milan, IT
alessia.galdeman@unimi.it

Matteo Zignani
Department of Computer Science
University of Milan
Milan, IT
matteo.zignani@unimi.it

Sabrina Gaito
Department of Computer Science
University of Milan
Milan, IT
sabrina.gaito@unimi.it

***Abstract*—Understanding and extracting knowledge from temporal networks is crucial to understand their dynamic nature and gain insights into their evolutionary characteristics. Existing approaches to network growth often rely on single-parameterized mechanisms, neglecting the diverse and heterogeneous behaviors observed in contemporary techno-social networks. To overcome this limitation, methods based on graph evolution rules (GER) mining have proven promising. GERs capture interpretable patterns describing the transformation of a small subgraph into a new subgraph, providing valuable insights into evolutionary behaviors. However, current approaches primarily focus on estimating subgraph frequency, neglecting the evaluation of rule significance. To address this gap, we propose a tailored null model integrated into the GERM algorithm, the first and most stable graph evolution rule mining method. Our null model preserves the graph's static structure while shuffling timestamps, maintaining temporal distribution, and introducing randomness to event sequences. By employing a z-score test, we identify statistically significant rules deviating from the null model. We evaluate our methodology on three temporal networks representing co-authorship and mutual online message exchanges. Our results demonstrate that the introduction of the null model affects the evaluation and interpretation of identified rules, revealing the prevalence of under-represented rules and suggesting that temporal factors and other mechanisms may impede or facilitate evolutionary paths. These findings provide deeper insights into the dynamics and mechanisms driving temporal networks, highlighting the importance of assessing the significance of the evolution patterns in understanding network evolution.**

***Index Terms*—graph evolution mining, subgraph mining, graph evolution rule, null models**

I. Introduction

Understanding, modeling and extracting knowledge from temporal networks representing complex and interconnected systems is becoming essential for uncovering their dynamic nature and gaining valuable insights into their evolutionary and temporal characteristics. Moreover, deep comprehension and mining of the patterns ruling their evolution and growth are fundamental in many applications from link prediction and change point detection to graph generative models for statistical inference [1] and resource management and forecasting. In recent years, various models, mechanisms, and measures have been proposed to describe the growth of networks. Most of these approaches, such as triadic closure [2], homophily [3], or node latent features [4], assume that growth is guided by a single parameterized mechanism. However, contemporary techno-social networks are characterized by diverse and heterogeneous behaviors, involving multiple choices and mechanisms. To better understand the underlying mechanisms driving network evolution without relying on predefined assumptions, it is more effective to employ methods that focus on identifying small frequent subgraphs that evolve over time.

In this context, the mining of graph evolution rules (GERs) has emerged as a promising approach. GERs offer interpretable patterns that describe the transformation of a small subgraph into a new subgraph, where the former serves as a prefix to the latter. Indeed, in the existing literature, there are numerous works that leverage the graph evolution rules approach to gain valuable insights into the evolutionary behaviors of networks. For instance, in [5] authors employed a GER mining algorithm to analyze different social and transaction networks and identified common and platform-specific evolution rules, shedding light on the evolutionary behavior of these networks; while Coscia and Szell [6] extended the mining of graph evolution rules to multiplex networks enhancing the task of multiplex link prediction by predicting new node arrivals and leveraging higher order structures with four or more nodes. Despite the successful application of GERs in analyzing network evolution, to draw meaningful conclusions from these rules it is essential to have a robust framework that allows for the evaluation of their significance; an aspect that is neglected by most of the current approaches, more focused on efficient solutions for estimating the frequency of subgraphs. In fact, previous studies have made attempts to evaluate the reliability of the rules discovered [7], [8] but not their statistical significance. Only Leung *et. al.* [9] have proposed a null model but restricted to a specific type of graph evolution rules, limiting their applicability. To the best of our knowledge, what is missing is the development of a tailored null model, that would enable the identification of statistically significant patterns through a broader graph evolution rule mining algorithm, thereby enhancing our comprehension of the dynamics and mechanisms driving network evolution.

In this paper, we move toward the above direction by developing a null model for the GERM algorithm [7], the first and most stable graph evolution rule mining method. The null model preserves the static structure of the graph while

shuffling the timestamps, thereby maintaining the temporal distribution and offering randomness on the temporal sequence of events. By employing a z-score test, we can identify statistically significant rules that deviate significantly from the random model. We evaluated our methodology on three different temporal networks based on two types of relationships: co-authorships and mutual online message exchanges. Alongside the quantitative findings on the number of rules found on the real and random graphs, we evaluated the effectiveness of being able to identify statistically significant and over/under-represented evolution patterns. Our findings show that the introduction of a null model impacts the evaluation and interpretation of the identified rules: some highly frequent rules are not significant at all, only a few are over-represented, while the majority of the GERs are under-represented. The high prevalence of under-represented rules supports the reasonable hypothesis that the shuffling of timestamps diminishes or eliminates specific temporal factors or mechanisms that may impede or facilitate evolutionary paths. Furthermore, the side information provided by GERM through the timespan $tspan$ of a rule, i.e. the time necessary to form the final subgraph, allowed us to extend the above hypothesis to the speed of the formation process of subgraphs: there may exist temporal factors and mechanisms favoring the fast formation of subgraphs since rules expressing this phenomenon are over-represented.

II. Background and Related Works

A. Graph Evolution Rules - GERs

Similar to association rules in data mining [10], a *graph evolution rule* - GER - consists of a precondition (referred to as the *body*) and a postcondition (referred to as the *head*). These rules can be interpreted as indicating that a subgraph matching the body is likely to evolve into the head, providing human-readable and explainable outcomes. For instance, Fig. 1 depicts a graph evolution rule that indicates the presence of triadic closure in a directed graph. Graph evolution rules offer a powerful approach to uncovering complex mechanisms within temporal networks. In fact, they not only provide insights into the underlying dynamics but also facilitate the development of more precise models for predicting how the network will evolve. Furthermore, the collection of graph evolution rules extracted from a specific network can serve as a differentiating factor from other graphs [5], which may differently evolve driven by distinct mechanisms.

The identification of graph evolution rules has been the subject of different research works. In general, current state-of-the-art methods for detecting the topological evolutionary processes in a network follow a two-step methodology. Initially, rules are extracted through frequent subgraph mining, followed by the application of filtering techniques utilizing measures such as support and confidence, i.e. frequency-based properties. These steps collectively contribute to the identification and characterization of the evolutionary patterns of the network. One of the earliest methods introduced in the literature for extracting graph evolution rules is GERM,

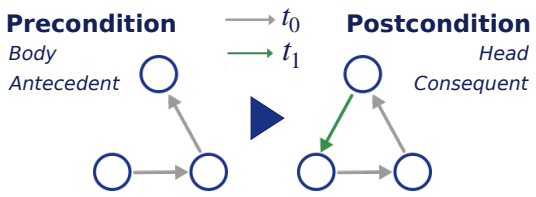


Fig. 1: Example of graph evolution rule - GER. On the left, the body of the rule, i.e. a graph with two links - grey arrows - created at time t_0, and on the right, the head of the rule represents the state of the body on the left side after evolving, indicated by the addition of a new link (green arrow) at the successive timestamp t_1.

developed by Berlingerio *et al.* [7] in 2009. Its rules can detect undirected edge insertion events, with relative time, but the processes of removing edges and relabeling nodes and edges are not captured. Leung *et al.* [9], and later Ozaki *et al.* [11], introduced the LFR (Link Formation Rule) algorithm, which aims to capture the creation of directed links between source and destination nodes. Both GERM and LFR algorithms utilized the minimum image-based support [12] and gSpan-based techniques for frequent subgraph mining [13]. Ozaki *et al.* [11] extended LFR to an undirected version and proposed a method for identifying relationships between rules. LFR rules from both [9] and [11], similarly to GERM rules, do not capture edge or node deletion and relabeling. However, LFR rules represent a subset of the ones obtained with GERM, because it adds constraints focusing on the insertion of a new link between a given source and destination node. The targeted approach of LFR rules in reducing the search space for rule extraction accelerates the process, but it results in a reduction of the obtained results. Additionally, Vakulík [8] developed the DGR miner, which incorporates edge deletion and relabeling in the evolution rules. The most recent work on GER identification is the EvoMine approach, introduced by Scharwächter [14]. Evomine extracts frequent rules of events (including relabeling and deletion) occurring in subsequent time windows. Other works in the literature focus more on the evolution of attributes and give less importance to the structural evolution of networks and the rules governing their growth [15], [16].

B. Microcanonical Randomized Reference Models - MRRMs

The aforementioned methods for GER extraction primarily emphasize algorithmic aspects, often disregarding the evaluation of rule significance. In this study, we address these limitations by introducing a method to assess the significance of the rules, incorporating specific null models. Null models are crucial for understanding both theoretical and practical aspects of networked systems, providing a baseline for comparison against which the observed patterns, features, and dynamics can be evaluated. By generating randomized or synthetic versions of the original temporal network, null models allow us to assess the significance and uniqueness of observed patterns, identify deviations from randomness, and uncover meaningful

structures and processes in the data. According to [17], given a space $\mathcal{G}$ of all possible temporal networks (states) and the single observation $G^* \in \mathcal{G}$, all models that sample a random graph G from a conditional probability $P(G|G^*)$ are defined *randomized reference models* (RRMs). In this context, the most popular models include configuration models, Erdös-Rènyi (ER) models, and exponential random graph models [1]. Depending on the application scenario, it could become necessary to preserve specific features or properties while generating random temporal graphs. This is where *micro-canonical randomized reference models* (MRRMs) come into play, providing a framework that allows us to retain specific characteristics of the graph while randomizing the rest. The concept behind this approach is preserving certain features while maximizing randomness. Relying on the principle of maximum entropy, the use of such models is theoretically justified since they offer the least biased approach among all possible degrees of freedom [17].

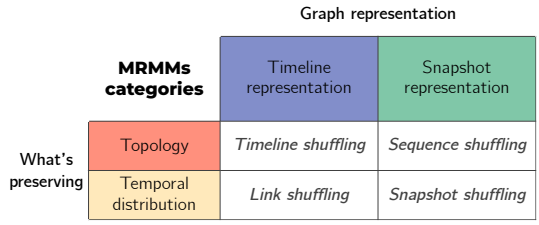


Fig. 2: Four primary categories of microcanonical randomized reference models. Each row represents the characteristic of the temporal network which is preserved, and each column refers to the representation of the temporal network, i.e. stream-based or snapshot-based.

MRRMs can be classified according to two factors: *i)* preservation of temporal distribution or topology, and *ii)* the representation of temporal networks as either timelines or snapshots. When models preserve the temporal distribution they favor capturing the temporal dependencies and dynamics of the original network. Conversely, other models might prioritize preserving network topology to gain insights into structural properties. The second factor revolves around the representation of temporal networks. In the timeline representation, the temporal ordering is crucial, with the graph capturing the static topology and additional time-series information describing the timing and duration of edges/events for each node. On the other hand, the snapshot model treats each time window as a separate graph (potentially with the same set of nodes), enabling a focus on individual time points. The combination of these two factors leads to four primary categories of MRRMs depicted in Fig. 2. Models in the first category, namely *timeline shuffling*, focus on preserving topology while utilizing a timeline representation. It involves maintaining the static network structure while shuffling timestamps within or between timelines and allows highlighting significative dynamics. In contrast, the category of *sequence shuffling* aims to preserve topology but uses a snapshot representation. It shuffles the order of snapshots while keeping the same network topology in each snapshot. When a model preserves the temporal distribution on a network modeled through a timeline representation, it falls under the *link shuffling* category. Here, the static links are shuffled, while their associated timelines remain unmodified. Lastly, *snapshot shuffling*, the fourth category, preserves the temporal distribution on graphs represented as temporal snapshots. It involves shuffling the edges within each snapshot, enabling the examination of individual time points in isolation.

In general, MRRMs represent a valid tool for assessing the significance of measurements on real-world temporal networks and for extracting meaningful insights about their dynamics and structure. Indeed, the literature offers various applications and fields where MRRMs have been employed, including but not restricted to contagion processes [18], temporal motifs [19], and random walks [20]. However, to the best of our knowledge, the application of a null model on a graph evolution rule algorithm has not been explored, specifically for rules tracking the relative timestamp of edges without temporal window constraints. Although the LFR algorithm [9] introduces a null model, the rules in that context are highly constrained.

III. Methodology

In this section, we present the methodology employed in our study to assess the significance of the evolution of patterns in temporal networks and analyze their main properties and roles. First, we describe the original graph evolution rule mining algorithm (GERM), which serves as the foundation for our analysis. Subsequently, we outline the process of constructing the null model, a critical component in assessing the significance of observed patterns. Then, we explain the process of extracting over and under-represented rules, enabling us to identify statistically significant patterns. Finally, we compute the general mapping of patterns, which aids in understanding the overall trends within the graph. Through this methodology, we provide a robust framework for analyzing evolution patterns and uncovering their underlying significance.

A. GERM (Graph Evolution Rules Miner)

Among the different alternatives for GER extraction presented in Section II, we opt for the GERM algorithm since it provides a more robust implementation and is able to identify GERs spanning many consecutive snapshots. Here we provide a brief overview of the methodological and implementation aspects of GERM, as introduced by Berlingerio *et al.* [7]. The idea is to extract frequent patterns, filter them based on a support measure, and subsequently derive graph evolution rules from the remaining patterns. We discuss the input graph format, support and confidence measures, and the method for extracting rules from temporal patterns in the following.

Graph representation. The GERM algorithm applies a modified version of the gSpan method [13] to a single graph temporal representation, called *union graph*. Formally, the union graph is described as $\mathcal{G} = (V, E, t)$, where each edge $(u, v, t(u, v)) \in E$ indicates that the first link between nodes

u and v occurred in timestamp $t(u, v)$, with $t : E \to [0, \dots T]$ being a function that maps edges into timestamps in the interval $[0, \dots T]$. The author modified the original gSpan algorithm to handle *relative-time patterns*, i.e. time-shifted isomorphic graphs, such as the one reported in Fig. 3. In the example, the two temporal graphs are equal except for a time constant $\Delta = 7$, so they can be aggregated into the same equivalence class. Specifically, the leftmost graph (Fig. 3a) is representative of the pattern since it contains timestamps starting from zero. The identification of relative-time patterns impacts the time complexity of the algorithm and reduces the number of redundant patterns.

Support measure. In order to identify frequent patterns in GERM, a crucial aspect is defining the notion of pattern frequency, i.e. *support*. The authors employ the minimum image-based (MIB) support [12], as it returns a reliable approximation of the actual occurrence count. Briefly, the MIB support of a subgraph p in a graph G refers to the minimum number of distinct nodes in the input graph that are assigned to each node in the subgraph p. The MIB support not only preserves the anti-monotonicity property, but it also offers a computational advantage compared to counting exactly the occurrences. Moreover, the authors also proposed a confidence measure, that can help to find frequent evolution patterns, but not related to the concept of confidence interval. Indeed, confidence is computed by dividing the support of the pattern's head by the support of its body. This ratio provides an estimate of the probability the head evolves into a subgraph that matches the body, and, thanks to the anti-monotonicity property its value falls in the range $[0, 1]$.

Rules extraction. The most important feature of rules derived through the GERM algorithm is that for each head exists only one body. This property plays a crucial role in simplifying the support computation, as the support of a rule is defined to be the same as the support of its head (the final stage). The transformation from a temporal pattern to a graph evolution rule involves excluding the last evolutionary step from the head, which then becomes the body of the rule. Formally, given a pattern head $H = (V_h, E_h)$, the body is the subgraph $B = (V_b, E_b)$, where $E_b = \{e \in E_h | t(e) < max(\{t(e) | e \in E_h\})\}$. Finally, if the body is a connected subgraph, then the rule r is defined as the couple (B, H).

B. Timeline shuffled null model

Inspired by the taxonomy of MRRMs for temporal networks proposed by Gauvin *et al.* [17], we implemented a microcanonical randomized reference model that falls in the timeline shuffling category. We based our selection of the proper null model on two guiding principles. Firstly, our focus is on analyzing the characteristics of the dynamics of the network, rather than its specific topology. Therefore, we specifically consider models that preserve the original topology. Secondly, we aim to maximize entropy, striving for highly randomized models that are as less biased and conservative as possible. Moreover, the graph representation adopted by GERM corresponds to a simplified version of the timeline representation described in [17]. In fact, GERM requires a unified topology merging all the temporal windows and then it adds a single timestamp to each edge - not a time series - corresponding to the first appearance of the interaction. For these reasons, the null model choice falls into the timeline shuffling category.

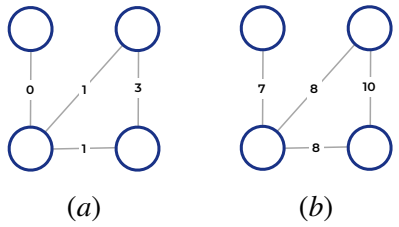


Fig. 3: Example of relative-time patterns. (a) and (b) are equal except for a time constant $Delta = 7$ for the link timestamps. The rightmost graph is representative of the class equivalence the graphs belong to.

Formally, given an input temporal network $G^* = (V^*, E^*, f^*) \in \mathcal{G}$, our null model SM returns a $G = (V^*, E^*, f) \in \mathcal{G}$ with probability $P_x(G|G^*)$ where:

- V^*, E^* are respectively the node and edge set of the static input network;
- $f^* : E^* \to T^*$ is a function that maps the edges of G^* into a timestamp $t \in T^*$;
- $f : E^* \to T^*$ is a function that maps the edges of G into a timestamp $t \in T^*$;
- $\mathcal{G}$ is the state space, i.e. a predefined finite set of temporal networks among which the MRRM selects G;
- $P_x(G|G^*) = \frac{\delta_{(x(G),x(G^*))}}{\Omega_x(G*)}$;
- $\Omega_x(G*) = \sum_{G' \in \mathcal{G}} \delta_{(x(G'),x(G^*)}$;

and δ being the Kronecker delta function, with the feature x being the intersection of two features, namely the edge feature ($\mathcal{E}$) and the timestamp feature ($\mathcal{T}$). Specifically, δ is defined as follows:

$$\delta_{(x(G),x(G^*))} = \begin{cases} 1 & \text{if } \mathcal{E}(G) = E^* \text{ and } \mathcal{T}(G) = T^* \\ 0 & \text{otherwise} \end{cases}$$

According to the notation proposed in [17], we will refer to our model as $P[\mathcal{E}, \mathcal{T}]$, addressing the features that it has to maintain in sampling the randomized graphs. In other words, among all networks in the state space $\mathcal{G}$ having the same nodes set V^* and the same timespan (links can be assigned to a timestamp from 0 to $max(T^*)$), the null model SM samples a temporal network G having the same edge set as the input network G^*, i.e. $\mathcal{E}(G) = \mathcal{E}(G^*)$, and having the same set of timestamps - $\mathcal{T}(G) = \mathcal{T}(G^*)$) - but with a different mapping function f. The pseudo-code of the shuffle model is depicted in Algorithm 1.

C. Significative GERs

According to the *Bonferroni's Principle* [10], frequent patterns can be discovered even in random data. As the dataset size increases, the occurrences of these patterns also tend to increase. Such frequent patterns or events are considered false positives in the search for patterns that characterize the data

Algorithm 1 Timeline shuffling model $P[\mathcal{E}, \mathcal{T}]$

Input: $G^* = (V^*, E^*, f^*)$, T^* **Output:** G

$G = (V, E)$
$V = V^*, E = E^*$
$T = shuffle(T^*)$
$n = 0$
for $(i, j) \in E$ **do**
    $f(i, j) \leftarrow T_n$
    $n \leftarrow n + 1$
end for

we are analyzing. By applying the Bonferroni correction, the significance level of each test is adjusted to ensure that the probability of false positives is appropriately controlled. This correction helps avoid spurious findings and strengthens the reliability of the results. The *zeta score test*, also known as the standard score or z-score test, is commonly used to assess the significance of an observation compared to a reference distribution. In the case of identifying significant patterns using random models, the zeta score test allows us to quantify how deviant or exceptional a pattern is compared to what would be expected by chance alone.

In this work, we apply the GERM algorithm to an input graph G^* to get the real support s_p of each frequent pattern p. Then, we run the GERM algorithm on fifty different realizations through the MRRM described in Section IIIB to extract the expected support μ_p and its standard deviation σ_p for each pattern $p \in \bigcap_{i=1}^{50} SM_i$, where $\bigcap_{i=1}^{50} SM_i$ is the set of patterns that are frequent in all the realizations of the null model. Note that the z-scores are computed over the mentioned set of rules ($\bigcap_{i=1}^{50} SM_i$) since the support measure (μ_p) for each run is required; this aspect will be further discussed in Section VC. The expected support corresponds to the average support of the pattern over all the 50 realizations of the model SM. Formally, it is defined as follows:

$$\mu_p = (\sum_{i=1}^{50} s_p^i)/50$$

where s_p^i is the support of p in the $i-th$ realization of the null model. Similarly, the standard deviation σ_p of each pattern $p \in \bigcap_{i=1}^{50} SM_i$ is computed on the supports of p over the realizations of SM. Finally, the z-score of the pattern p is computed as follows:

$$z_p = \frac{s_p - \mu_p}{\sigma_p}$$

After computing the z-score for each pattern $p \in \bigcap_{i=1}^{50} SM_i$, a pattern is deemed significant or overrepresented when its z-score exceeds the critical value of 1.96. This critical value corresponds to a significance level of 0.05, assuming a normal distribution. On the contrary, patterns with $z < -1.96$ are significantly more frequent in the randomized model's realizations, thus being under-represented or uncommon in the observed data. Analyzing such patterns can provide insights into the absence or suppressed occurrence of certain dynamics or relationships within the dataset. It is important to investigate both positive and negative z-scores to gain a comprehensive understanding of the patterns' significance and potential implications in the context of the analysis.

D. Mapping of temporal patterns across null model realizations

One challenge encountered during the computation of the z-score of patterns p was the lack of a canonical form for identifying the same pattern across the different outcomes of the GERM algorithm on the realizations of the null model. Indeed, the application of GERM on each realization generates patterns with their respective edge lists and supports, identified by independent incremental IDs. To overcome this issue and facilitate the comparison of results across multiple models, it is necessary to extract equivalence classes for each pattern, incorporate timestamp information, and assign a global ID to each temporal pattern. This process ensures consistency and allows for meaningful comparisons of patterns and their properties, even across different datasets. A preliminary observation that significantly contributes to reducing computational time is the frequent occurrence of patterns composed of the same set of edges within the realizations of the null model. Leveraging this observation, an initial step in the mapping procedure involves extracting the unique set of edge lists while preserving their respective occurrences. The pseudo-code of the procedure is reported in Algorithm 2. It requires as input the variable $germ$, which refers to the results of GERM on each realization of the MRRM (frequent patterns and their support). The algorithm returns a dictionary $edge\_set$ whose keys are the set of edges found in all the realizations, and the values are the list of occurrences of the key edge list in the form (p, m) with p being the original ID of the pattern and $m \in [1, 50]$ the ID of the realization it appears in.

Algorithm 2 General mapping: set of edges

Input: results of GERM on each realization of the null model $germ$

Output: $edges\_set$

$edges\_set = dict()$
$m = 0$
for $model \in germ$ **do**
    for $p, pattern \in model$ **do**
        push($edges\_set[pattern_{edges}]$, (p, m))
    end for
    $m \leftarrow m + 1$
end for

However, there is still a chance that the edge lists in the dictionary keys include isomorphic subgraphs. To avoid redundancy we check the absence of isomorphic patterns, before assigning a global identifier to each pattern/edge list. Algorithm 3 describes the procedure: it initializes the i variable to the maximum existing key in the $mapping$ dictionary. The reason for that is to create a global mapping for each dataset,

so when processing the results we may have some patterns that have been already enumerated. We iterate on the edge lists and their occurrences collected by Algorithm 2. After initializing the temporary id to zero, we search in mapping if there is already a pattern isomorphic to the one associated with the edge list we are considering. If so, we assign p to id. Afterward, if after the search id is still zero (there is no isomorphic pattern in $mapping$), then the enumeration variable i gets increased, and assigned to id. At this point, $mapping$ is updated with the new temporal isomorphism class i, with the subgraph $G(edges)$ obtained by the so far unseen edge list. At the end of each iteration of the outer loop on $edge\_set$, the new data dictionary $new\_shuffle\_germ$ is updated with the new global id and the same information as the original one.

Algorithm 3 General mapping: isomorphism check

Input: $edges\_set$
Output: $new\_shuffle\_germ$

$mapping = dict()$
$new\_shuffle\_germ = dict()$ ▷ Inspired by Python, dict() creates an associative map
$i = max(mapping.keys)$
for $edges, occurrences \in edges\_set$ **do**
  $id \leftarrow 0$
  for $p, pattern \in mapping$ **do**
    if $G(edges)$ *is_isomorphic* $G(pattern)$ **then**
      $id \leftarrow p$
      $continue$
    end if
  end for
  if $id = 0$ **then**
    $i \leftarrow i + 1$
    $id \leftarrow i$
    $mapping[i] = G(edges)$
  end if
  for $p, m \in occurrences$ **do**
    $new\_shuffle\_germ[m][id] = germ[m][p]$
  end for
end for

IV. Datasets

We applied our methodology (see Section III) to two different human-centered temporal datasets. The first one is a bibliographic network representing co-authorship relationships, and the second one encodes information about students of an American university that exchange messages on an internal platform.

A. DBLP

DBLP (Digital Bibliography & Library project) is a computer science bibliography website and the dataset used in this work is available at https://www-kdd.isti.cnr.it/GERM/. The dataset gathers from DBLP all the co-authorship interactions from 1992 to 2002 with a yearly granularity. In this case, an undirected temporal edge (a, b, t) indicates that there was at least one publication that both authors a and b co-authored in the year t. As the graph evolution rules algorithm - GERM - cannot handle multiple edges, we only consider the first edge that appears between each pair of nodes. The obtained network has 129073 nodes and 277081 edges, with 11 possible timestamps. Two additional datasets are accessible, encompassing the time intervals of 2003 to 2005 and 2005 to 2007, respectively. Nevertheless, the time span of these two datasets is comparatively brief, comprising solely two years of data. Consequently, we abstain from presenting any findings derived from them.

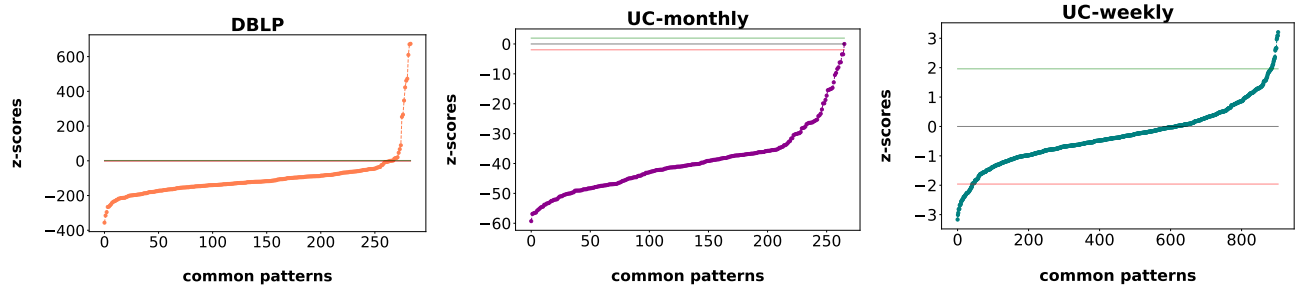
B. UC Social

The second dataset is obtained from a directed temporal network available at http://konect.cc/networks/opsahl-ucsocial/, which collects the message interactions among students of the University of California (Irvine) in an online community. The graph represents users as nodes, and each directed edge (u, v, t) denotes a message sent from u to v at time t. Similar to the previous dataset, we only consider edges with their earliest timestamp due to the inability of the algorithm to handle repeated interactions. Moreover, the dataset offers a directed type of interaction, however, the GERM algorithm only handles undirected graphs. So, we processed the directed graph G_d to create a mutual undirected graph G_m: each edge $(u, v, t_1) \in G_d$ is inserted into G_m if and only if $(v, u, t_2) \in G_d$. In this case, the timestamp of $(u, v) \in G_m$ is $min(t_1, t_2)$. This way, we obtain an undirected graph with 1280 nodes and 12916 edges, that reflects reciprocal relationships. To make the analysis more tractable, we aggregated the original timestamps of edges by weekly and monthly granularities. This allows us to track the mesoscopic evolution of the network over time without being overwhelmed by a large number of possible timestamps. With the aggregation, we obtain two different graphs UC-monthly and UC-weekly, having the same size and order, but with the former having a coarser time granularity. Precisely, the UC-monthly graph has only 7 possible timestamps, while UC-weekly has 28 possible timestamps.

It is worth noting that the two datasets offer different advantages: the first one provides a large temporal network, while the second one allows for arbitrarily tuning the time granularity.

V. Findings

We apply the methodology described in Section III to the GERM algorithm, for each of the graphs described in Section IV. In this section we analyze the main results, starting from a quantitative description of the rules found in both the real graphs and their corresponding null models. Then, we discuss the distribution of z-scores, and evaluate the impact of the patterns that were not included in the z-score computation. Finally, we also group the patterns by the time they take to form to gain a deeper understanding of the graph evolution.



(a) (b) (c)

Fig. 4: Distribution of the z-scores of GERs extracted from (a) DBLP, (b) UC-monthly, and (c) UC-weekly. The thresholds above and below which patterns attain statistical significance are depicted by green and red horizontal lines, respectively.

A. *GERM outcomes on real and randomized networks*

GERM algorithm was applied to the three networks as well as to their timeline shuffled counterparts, setting a maximum of 4 edges (patterns involve at most four links) and generating 50 realizations of the null models. The minimum support values were selected starting from a value of 5000, which is the original choice of GERM's authors for the DBLP dataset. As concerns the UC networks, the support values were selected based on the minimum value needed to obtain non-empty outputs, starting from 5000. Table I reports the selected support thresholds and the corresponding number of rules identified. Specifically, the *GERM* column reports the number of rules found running the GERM algorithm with the aforementioned parameters on the original/real graphs, while the "*mean shuffle*" and "*union shuffle*" columns refer to the application of the algorithm on the timeline shuffled graphs (null model). The former indicates the average number of patterns found by executing the GERM algorithm on 50 randomized graphs. The latter (*union shuffle*) indicates the different rules/patterns found in the union of the 50 runs. The union is computed utilizing the general mapping algorithm described in Section IIID.

dataset	support	GERM	mean shuffle	union shuffle
DBLP	5000	296	3795	3871
UC-monthly	150	266	1269	1378
UC-weekly	600	999	1039	1235

TABLE I: Overview of GERM outcomes on the three temporal networks. "GERM" column displays the number of rules identified in the real graphs. The "mean shuffle" column reports the average number of patterns over 50 timeline shuffled graphs. The "union shuffle" column presents the number of distinct rules discovered through the union of 50 realizations.

Upon examining the results in Table I, we observe that on average in null models there are more patterns than in the original graphs. Further, this difference is consistent across all 50 realizations of the randomized graphs, as we can note from the small difference between the average number of patterns (fourth column) and the union of the rules (fifth column). This observation indicates that in timeline shuffled realizations there are a consistently higher number of patterns having negligible support in the real temporal networks, i.e. timestamps and the ordering they induce in the real graphs impact the frequency of the evolution rules.

B. *Analysis of z-scores*

The application of a null model on the graph evolution rules enables the identification of significant rules specific to the temporal graph under consideration, extending beyond frequency-based measures. As described in Section III we compute the z-score for the patterns existing in all the realizations of the null model and in the real graph. Fig. 4 shows the distribution of the z-score for the three temporal networks, where horizontal lines specify the noteworthy thresholds: the grey line stands for the zero level (patterns with identical support in both real and null graphs), while the green and red horizontal lines indicate the over and under-representation thresholds (± 1.96). While the under-representation of patterns is a consistent trend across all three datasets, there are notable differences in the distribution of z-scores across the three temporal networks. First, the UC-weekly case (see Fig. 4c) is a special case because the majority of patterns (93.1%) are concentrated within the ± 1.96 region, meaning that their supports in the null or real graphs do not differ so much, i.e. they are not significative. In the other networks, almost every pattern is under-represented; for instance, in the DBLP case (see Fig. 4a), the over-represented rules are only 16 against 264 under-represented. Second, in the DBLP case, z-scores present a very wide range of values, meaning that the supports of patterns in the real graph are extremely lower or higher than the ones in the null models. Third, in the UC-monthly network, all patterns are under-represented over the null model (see Fig. 4b). We will investigate the underlying reasons in subsequent sections.

C. *Frequency of GERs in real and randomized networks*

While the common set of patterns obtained from GERM on the original graph and its randomized versions provides valuable insights into over and under-represented graph evolution rules, the patterns that are not in the intersection of all runs of GERM over the realizations may be worth attention too. In fact, when running GERM algorithm on the real graph and on the 50 realizations of the null model, we obtain a set of frequent rules from each of the 51 runs, generating different

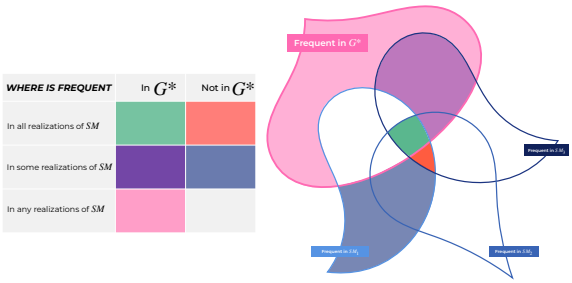


Fig. 5: Tabular and visual representation of possible cases a rule/pattern may be involved in. Here, we depict 3 realizations of the null model (SM). A pattern in the green set is a GER in the real network G^* and a GER in all the realizations, while if it is present in some realizations but not all, it is purple. If the pattern is not a GER in the real network - its support is below the threshold - it may belong to the red set, i.e. it is a GER in all the realizations, or it is in the blue set, i.e. it is a GER in some realizations but not all. Finally, patterns belonging to the pink case are GERs in the real network, but in all the realizations their support is below the threshold.

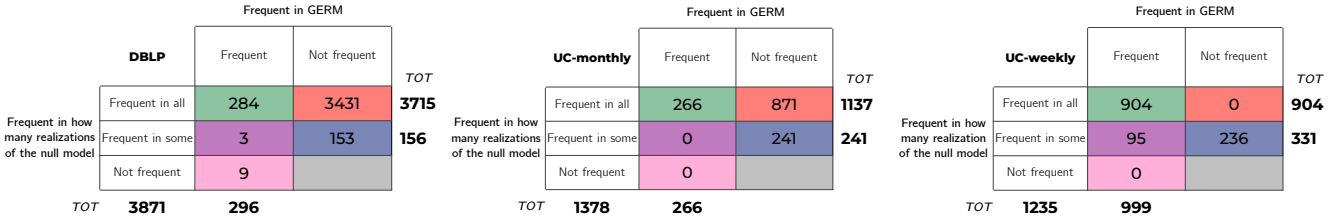
scenarios depending on the frequency of a pattern in the real and randomized realizations. In Fig. 5 we summarize all the possible scenarios combining a tabular representation and a set representation. According to the above representation, up to now, we have analyzed the patterns belonging to the green set, i.e. the set containing patterns appeared to be frequent in all realizations of the null models (SM_i) and even in the real graph (G^*). On the opposite side, we have the pink and red sets: the first one contains all patterns that were frequent only in the real graph (returned by GERM) but not in any randomized network: these are reasonably considered as over-represented patterns since the support in the null model is always lower than the real one. On the other hand, the patterns in the red set (frequent in all realizations of the null model but not on the real graph) are reasonably under-represented for the complementary reason, and so equally worthy of attention. Finally, since we generated many realizations of the null model, it is likely that some patterns are not always frequent (blue and purple sets). Still, if they are frequent in most of the 50 realizations, they may be worthy of analysis. In essence, by relaxing the initial conditional requirement, as required in the definition of the z-score, that restricts the evaluation of statistical significance solely to GERs extracted in all realizations of the null model ($p \in \bigcap_{i=1}^{50} SM_i$), we can broaden the analysis of significance to encompass a wider range of evolution rules.

We evaluate the extent to which patterns can be reintegrated into our analysis by examining the tables presented in Fig. 6, which provide a comprehensive overview of the pattern distribution across various scenarios for the three temporal networks. The analysis reveals that the red and green sets are generally larger than the blue and purple sets, indicating that we can consistently enlarge the set of under-represented GERs in DBLP and UC-monthly. On the other hand, the pink set was nearly non-existent in all datasets, so the extension of over-represented patterns is very marginal. Furthermore, the size of the purple set is found to be smaller than the blue set, suggesting that patterns that are not frequent in all the realization of the null model, are probably uncommon in the real graph as well. In general, the expansion of assessable GERs aligns with the trend observed in the analysis of the z-scores, wherein a majority of the evolutionary rules demonstrate under-representation, particularly within the DBLP and UC-monthly datasets. The notable prevalence of under-represented GERs underscores how temporal constraints and/or evolutionary mechanisms within the three temporal networks may hinder the manifestation of certain evolution rules. These factors are loosened in the realizations of the null model.

Finally, we look at the support of the patterns in each set and investigate any similarities or differences that may emerge. Fig. 7 depicts the support distributions of each set for all three networks. While not all datasets contain all five sets, it is evident that the supports for the purple and blue sets are rather low, particularly when compared to the green and/or red sets. This last finding suggests that patterns that are not frequent in all the realizations of the null model may not be worthy of attention, given that they only marginally exceed the minimum support threshold. Therefore, we do not consider the purple and blue sets in further analysis. On the other hand, the pink set, which is exclusive to real networks, represents a small but significant set of patterns that cannot be detected in the realizations of the null models.

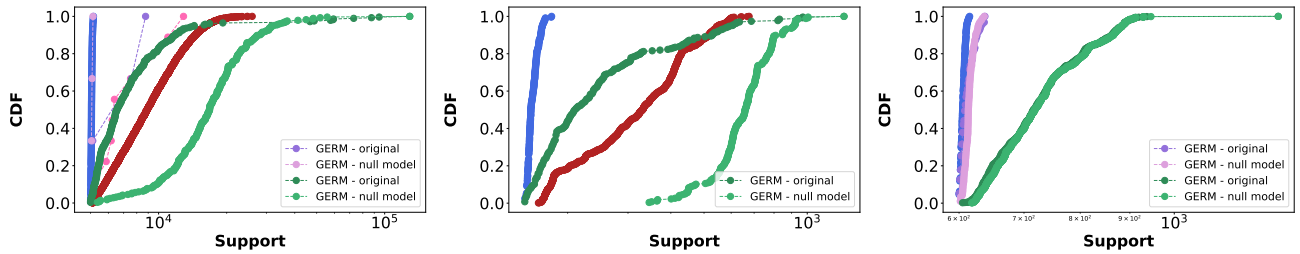
D. Analysis of timespans

A further advantage of GERM is that the returned GERs are provided with the time the body takes to form, i.e. its timespan $tspan$. Indeed, the timespan of a GER corresponds to the maximum timestamp in the body of the rule. We focus on the maximum timestamp because we consider relative-temporal patterns, with the minimum timestamp always set to zero. In the context of statistically significant GERs this information is important since we can discover signals that certain constraints or evolution mechanisms may favor or contrast slow or fast formation of specific subgraphs. Fig. 8 shows the frequency distribution of the timespans of the rules in the three datasets, grouped according to the set a pattern belongs to, namely the red and green sets. We observe that GERs within the green sets exhibit relatively lower timespans, typically around 3, indicating their prevalence within shorter temporal intervals and a relatively fast formation process. On the other hand, the red sets include patterns characterized by higher timespans, suggesting their persistence and prevalence over longer durations, and a slower formation process. This disparity in timespan distribution between the green and red sets provides valuable insights into the temporal dynamics within the analyzed datasets, suggesting that in the real graphs the frequent patterns are the ones that happen in a shorter time interval. In fact, the higher timespan patterns are more common in the null model but not in the real graph. The distinction in terms of timespan between the green and red sets is particularly evident when considering the support of these



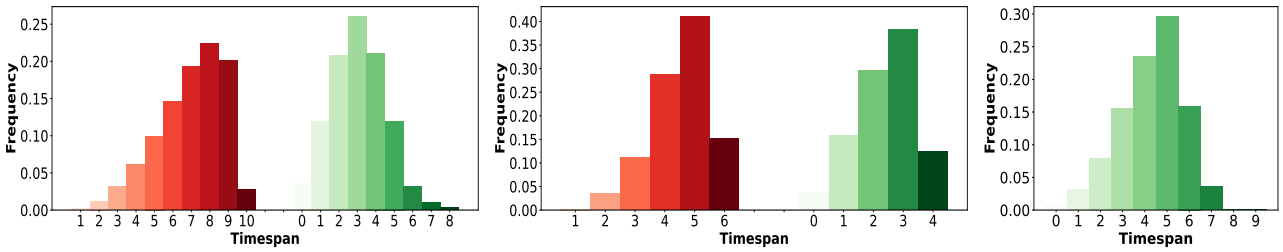
(a) (b) (c)

Fig. 6: Tabular representation of the different cases depicted in Fig. 5, for the different temporal networks (a) DBLP, (b) UC-monthly, and (C) UC-weekly. Columns in each table indicate whether a GER extracted by GERM is frequent or not in the real graph. Rows indicate how frequently a GER has been extracted by GERM over the 50 realizations of the null model. Each record reports how many GERs belong to the specific case.



(a) (b) (c)

Fig. 7: Distribution of the support of rules/patterns grouped by different cases described in the Fig. 5. As for the support of the GERs in the realizations of the null model, we report the distribution of the mean support. The x-axis has a logarithmic scale.



(a) (b) (c)

Fig. 8: Distribution (frequency) of the timespans of the evolutions rules in the three temporal networks (a) DBLP, (b) UC-monthly, and (c)UC-weekly. We grouped timestamps according to the class their pattern belongs to. The palettes of histograms recall the colors assigned to each class in Fig. 5.

patterns. For instance, a significant majority (around 89%) of the over-represented patterns in the DBLP graph exhibit a $tspan < 3$, while the 65% of most frequent patterns in the red sets present a $tspan \geq 4$. This observation provides additional support for the hypothesis that various factors and mechanisms influence the rapid formation of specific subgraphs, while other factors act against the formation of certain subgraphs, thereby slowing down the overall formation process.

E. Discussion

Finally, we illustrate a few examples showing the enhanced value provided by the null model extension on the graph evolution rule algorithm. By analyzing these patterns, we aim to show how the assessment of the significance can offer valuable insights beyond the outcomes traditional frequency-based mining algorithms provide.

In the case of the DBLP temporal network, the introduction of the null model reveals patterns with more intricate structures than usual chains or triangles. For instance, the GER in Fig. 9a is resulted to be significative, meaning that its support in the real graph is significantly more frequent than in the null model. However, in terms of support rank, it falls beyond the 100^{th} position. Without the null model, such a pattern might have been overlooked in a semantic analysis, yet it holds important information about the graph's evolution. Indeed, this GER represents two authors who start their collaboration at a specific time (t_0) and subsequently collaborate with a common co-author the following year, the target of a further collaboration with another author external to the initial pair.

On the other hand, the GER in Fig. 9b exemplifies a pattern with high support (14^{th} position) but a remarkably low z-score

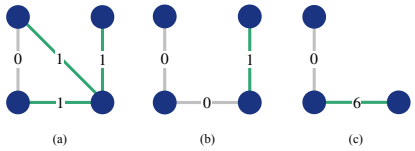


Fig. 9: Example of GERs extracted from the DBLP temporal network.

(-216). This means that its having a high support may not mean that is actually a pattern worth attention in explaining the dynamics of the graph, as its support becomes substantially higher when the network timestamps are shuffled. Without the null model extension, we might have considered the dynamic where a triangle between three authors does not close in the subsequent year as important, while in reality, it may not hold such significance.

Another observation could be drawn by looking at the lower tail of the z-score distribution: Fig. 9c depicts a rule that has one of the lowest z-scores (-257.7) that also has a large timespan (6 years). This is telling us that this kind of pattern presents a really higher support in the models, suggesting that an author that starts a collaboration at a certain t_0, may start another one in 6 years. However, this occurrence is not as common as in a random graph where temporal dependencies are loosened. Therefore, it represents a common behavior that could be even more frequent whereas temporal constraints or other mechanisms would not come into play.

By presenting these diverse examples, we aim to underscore the added value of the null model extension in capturing nuanced dynamics during the evolution of temporal networks. These examples showcase how the introduction of the null model framework enables us to uncover patterns that possess distinctive characteristics, highlighting their significance in understanding network evolution.

VI. Conclusions

In this study, we disentangle the evolution of different temporal networks by the identification of statistically significant graph evolution rules The assessment of the statistical significance results from the introduction of a proper null model applied to the GERM algorithm, the first and most stable method for mining graph evolution rules. The null model preserves the static structure of the graph while shuffling timestamps, ensuring the temporal distribution is maintained and introducing randomness to the sequence of events. By employing a z-score test, statistically significant rules that deviate significantly from the null model are identified. Although the significance of the identified GER has been almost neglected, our findings show that the introduction of a null model impacts the evaluation and interpretation of rules. First, a few highly frequent rules are not significant at all, only a few are over-represented, while the majority of the GERs are under-represented. So, by shuffling timestamps, we weaken or remove specific temporal factors or mechanisms that may inhibit or favor evolution paths. Furthermore, we also extended this observation to the speed of the formation process of subgraphs, where rules expressing fast formations are over-represented. As a future extension of this work, we plan to identify the mechanisms and the factors acting on the over and under-representation of the GERs and assess their role in the dynamics of subgraph formation.

References

[1] T. A. Snijders, "Statistical models for social networks," *Annual review of sociology*, vol. 37, pp. 131–153, 2011.

[2] G. Bianconi, R. K. Darst, J. Iacovacci, and S. Fortunato, "Triadic closure as a basic generating mechanism of communities in complex networks," *Physical Review E*, vol. 90, no. 4, p. 042806, 2014.

[3] F. Papadopoulos, M. Kitsak, M. Serrano, M. Boguná, and D. Krioukov, "Popularity versus similarity in growing networks," *Nature*, vol. 489, no. 7417, pp. 537–540, 2012.

[4] M. Kim and J. Leskovec, "Modeling social networks with node attributes using the multiplicative attribute graph model," in *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, 2011, pp. 400–409.

[5] A. Galdeman, M. Zignani, and S. Gaito, "Disentangling the growth of blockchain-based networks by graph evolution rule mining," in *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2022, pp. 1–10.

[6] M. Coscia and M. Szell, "Multiplex graph association rules for link prediction," in *Proceedings of the Fifteenth International AAAI Conference on Web and Social Media, ICWSM 2021*. United States: AAAI Press, 2021, pp. 129–139.

[7] M. Berlingerio, F. Bonchi, B. Bringmann, and A. Gionis, "Mining graph evolution rules," in *joint European conference on machine learning and knowledge discovery in databases*. Springer, 2009, pp. 115–130.

[8] K. Vaculík, "A versatile algorithm for predictive graph rule mining." in *ITAT*, 2015, pp. 51–58.

[9] C. W.-k. Leung, E.-P. Lim, D. Lo, and J. Weng, "Mining interesting link formation rules in social networks," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, ser. CIKM '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 209–218.

[10] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive data sets*. Cambridge university press, 2020.

[11] M. Yuuki, T. Ozaki, and O. Takenao, "Mining interesting patterns and rules in a time-evolving graph," *Lecture Notes in Engineering and Computer Science*, vol. 2188, 03 2011.

[12] B. Bringmann and S. Nijssen, "What is frequent in a single graph?" in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2008, pp. 858–863.

[13] X. Yan and J. Han, "gspan: Graph-based substructure pattern mining," in *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* IEEE, 2002, pp. 721–724.

[14] E. Scharwächter, E. Müller, J. Donges, M. Hassani, and T. Seidl, "Detecting change processes in dynamic networks by frequent graph evolution rule mining," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 1191–1196.

[15] P. Fournier-Viger, G. He, J. C.-W. Lin, and H. M. Gomes, "Mining attribute evolution rules in dynamic attributed graphs," in *International Conference on Big Data Analytics and Knowledge Discovery*. Springer, 2020, pp. 167–182.

[16] K.-N. T. Nguyen, L. Cerf, M. Plantevit, and J.-F. Boulicaut, "Discovering inter-dimensional rules in dynamic graphs," in *Proceedings of the 1st International Conference on Dynamic Networks and Knowledge Discovery-Volume 655*, 2010, pp. 5–16.

[17] L. Gauvin, M. Génois, M. Karsai, M. Kivelä, T. Takaguchi, E. Valdano, and C. L. Vestergaard, "Randomized reference models for temporal networks," *SIAM Review*, vol. 64, no. 4, pp. 763–830, 2022.

[18] P. Holme and F. Liljeros, "Birth and death of links control disease spreading in empirical contact networks," *Scientific reports*, vol. 4, no. 1, p. 4999, 2014.

[19] M. Karsai, K. Kaski, A.-L. Barabási, and J. Kertész, "Universal features of correlated bursty behaviour," *Scientific reports*, vol. 2, no. 1, pp. 1–7, 2012.

[20] J.-C. Delvenne, R. Lambiotte, and L. E. Rocha, "Diffusion on networked systems is a question of time or structure," *Nature communications*, vol. 6, no. 1, p. 7366, 2015.