



An online and nonuniform timeslicing method for network visualisation

Jean R. Ponciano^{a,b,*}, Claudio D. G. Linhares^{a,c}, Elaine R. Faria^a, Bruno A. N. Travençolo^a

^aFaculty of Computing, Federal University of Uberlândia, Brazil

^bSchool of Applied Mathematics, Getulio Vargas Foundation (FGV), Brazil

^cInstituto de Ciências Matemáticas e de Computação (ICMC), University of São Paulo (USP), São Carlos, Brazil

ARTICLE INFO

Article history:

Keywords: Temporal network visualisation, Nonuniform timeslicing, Streaming network, Network sampling

ABSTRACT

Visual analysis of temporal networks comprises an effective way to understand the network dynamics. It facilitates the identification of patterns, anomalies, and other network properties, thus resulting in fast decision making. The amount of data in real-world networks, however, may result in a layout with high visual clutter due to edge overlapping. This is particularly relevant in the so-called *streaming networks*, in which edges are continuously arriving (online) and in non-stationary distribution. All three network dimensions, namely *node*, *edge*, and *time*, can be manipulated to reduce such clutter and improve readability. This paper presents an online and nonuniform timeslicing method that enhances temporal and streaming network analyses. We conducted experiments using two real-world networks to compare our method against uniform and nonuniform timeslicing strategies. The results show that our method automatically selects timeslices that effectively reduce visual clutter in periods with bursts of events. As a consequence, decision making based on the identification of global temporal patterns becomes faster and more reliable.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Networks represent a useful and widely adopted structure to model systems from distinct areas, such as computer science, biology, sociology, and others [1]. A network is defined in terms of nodes (instances) and edges (the relationship involving them) [2]. In this way, a network may be used to represent the World Wide Web (Web pages connected by hyperlinks), an organism cell (chemicals linked by chemical reactions), social interactions, and many others [3]. In several situations, using only information about nodes and edges may not be enough to represent and comprehend the relations in the network. In social network analysis, for example, the information of *when*

the connections occur may be crucial to describe such relations with less (or even without) loss of context.

The so-called temporal (or dynamic) network considers time information in addition to nodes and edges [4]. It can be represented by $G = (V, E)$, where $V = \{n_1, n_2, \dots, n_N\}$ is the set of nodes in the network and $E = \{e_1, e_2, \dots, e_M\}$ is the set of edges. In our context, each edge $e_i = (n_x, n_y, t_k)$ connects two nodes $n_x, n_y \in V$ at a particular and discrete timestamp t_k [5, 4]. Considering t_{end} as the end of the observation period, $0 \leq t_k \leq t_{end}$. In fact, an edge that occurs at t_k actually occurs in the interval $[t_k, t_k + \tau)$, where τ is the temporal resolution [5]. To simplify, in our context, self-edges (i.e., edges connecting a node to itself) are removed [5]. In temporal networks, the presence of an edge represents the occurrence of an *event* at the respective timestamp. Moreover, all events are previously known and available to be used in the analysis (offline scenario) [4]. A temporal network has a delimited observation period, its data usually fit in primary memory and unrestricted random access is allowed.

*Corresponding author

e-mail: jeanrobertop@gmail.com (Jean R. Ponciano), claudiodgl@gmail.com (Claudio D. G. Linhares), elaine@ufu.br (Elaine R. Faria), travencolo@gmail.com (Bruno A. N. Travençolo)

In several real-world applications, however, data are produced in a massive and continuous way (online scenario). Streaming data that represent interactions among elements may be naturally represented as a *streaming network* [6, 7, 8]. For example, in a telecommunication context, phone calls form a network between the involved phone numbers [9]. In the same way, web pages and the links between them form a web network. Methods for processing streaming networks require efficient and real-time processing. This means that a streaming (or online) algorithm, besides the restricted access to the stream data (nodes/edges), must process the stream in a single scan, or in a small number of scans [9]. We define a streaming network $S = \{e_1, e_2, \dots, e_m, \dots\}$ as a temporal network $G = (V, E)$ as follows: $e_i = (n_x, n_y, t_k)$, $e_i \in E$, represents an edge that occurs at a discrete timestamp t_k , $0 \leq t_k \leq \infty$, between $n_x, n_y \in V$, $|V| \rightarrow \infty$. Note that it is possible to have more than one edge (event) per timestamp. This definition is different from the ones that consider each event arriving in a different timestamp [10, 11, 12].

Both temporal and streaming networks can be analysed through different strategies. Statistical analysis represents a common approach and is useful to identify specific trends and patterns in the data, being used, for example, in connection prediction [13] and burst analysis [14]. When there is only a numeric output, however, it may represent a “black-box” to the user, thus impairing pattern comprehension. Another approach involves Information Visualisation [15, 16], whose strategies assist data analysis by providing interactive and graphical computational tools, thus including the user in the entire process of exploration and validation. An adequate Information Visualisation strategy allows a visual analysis that is as much intuitive as possible and also helps the user in finding unexpected patterns, anomalies, and other behaviours in the data, thus resulting in a fast and reliable decision making. Examples of recent visualisation methods applied to temporal and streaming networks can be found in [5, 17, 18, 19, 20, 10, 21, 22, 23].

Network visualisation techniques, such as node-link diagrams [24, 25] and the *Massive Sequence View (MSV)* layout [25, 26], suffer from visual clutter caused by the amount of information, greatly impairing the analysis. The network temporal resolution plays an important role in the layout construction and, consequently, in the visual analysis. In several scenarios, as, for example, when the network is temporally sparse, changing the temporal resolution through the grouping of events from subsequent timestamps – a process called *timeslicing* – may facilitate the analysis and highlight patterns that would be difficult to see using the original resolution [17, 27]. Choosing the length of time that each resulting timestamp (also known as timeslice) must have, however, is not a trivial task. A naive and widely adopted timeslicing approach is to consider timeslices of equal length to represent the network (*uniform timeslicing*), each comprising, e.g., all events from 1 minute or 1 day interval [28, 17, 29, 25]. Despite the simplicity, the adopted length is a global, static, and pre-defined value that does not faithfully represent the number of events and their distribution. In both temporal and streaming networks, such distribution is non-stationary and changes over time, so the times-

licing method should consider this nonuniform behaviour. Although nonuniform timeslicing is often used in other contexts (e.g., multithreaded communication [30], data transfer [31], and systems performance degeneration analysis [32]), it has only been considered recently in temporal network visualisation [33]. Despite recent advances, we are not aware of online and nonuniform timeslicing methods, characteristics necessary for streaming network visualisation.

In this paper, we propose an online and nonuniform timeslicing method that automatically adapts the network temporal resolution scale according to the non-stationary distribution of events over time. Our method allows the identification of visual patterns, mostly global ones, that would be lost or difficult to find with a uniform timeslicing. Our focus is on online scenarios (streaming networks), which brings new challenges, as, for example, the need for fast (often real-time) methods and the immediate disposal of edges after processing [10]. Since any method developed for online scenarios can be applied in offline ones [6], our method can also be used to enhance temporal network analysis. Even knowing all events and having the possibility of unrestricted random access, improving the overall layout readability is already a challenging problem in temporal network visualisation [29, 25]. Our proposal thus benefits both scenarios.

The paper is organised as follows. Section 2 presents related concepts and Section 3 discusses related work. Our timeslicing method is presented in Section 4. Section 5 presents the method evaluation through two case studies using real-world networks. Finally, Section 6 describes the method’s limitations and Section 7 discusses conclusion and future work.

2. Background

This section presents fundamental concepts and discusses strategies focused on network visualisation and timeslicing. The terms “timestamp” and “timeslice” will be used interchangeably.

2.1. Network Visualisation

The employment of an effective temporal network visualisation strategy helps the user in the network evolution comprehension and facilitates the identification of patterns, anomalies, and other network properties. In this context, several visual strategies may be adopted, such as matrix-based [34, 35] and circular approaches [18], node-link diagrams [36, 17], and *Massive Sequence View (MSV)* layouts [37, 26]. Among these, node-link diagrams and MSV represent the best strategies when the task is to analyse the edge (event) distribution over time [18, 5].

Timeline-based visualisations. The *Massive Sequence View* [37, 26] is a timeline-based layout [38] similar to Bio-Fabric [39]. Its x -axis represents the timestamps and the y -axis represents the nodes of the network. In this layout, nodes cannot change their positions over time. Every time there is an edge between a pair of nodes, a vertical line is drawn linking them in the respective timestamp. The construction of MSV using the tabular (raw) data is illustrated in Fig. 1(a,b).

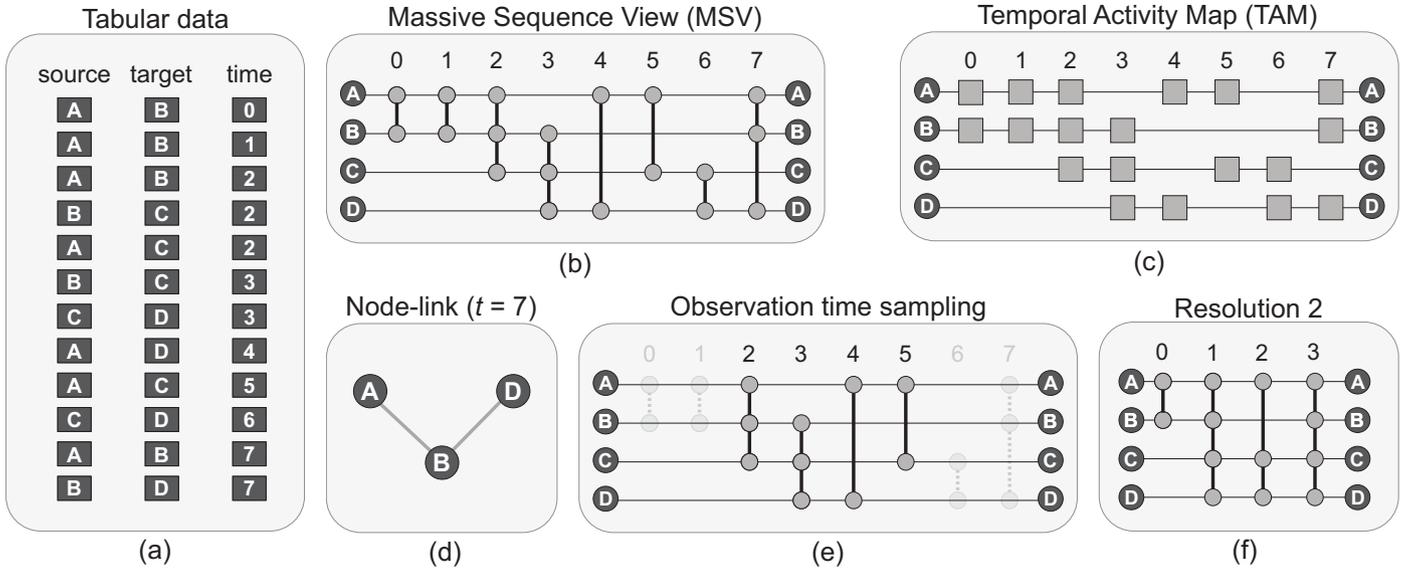


Fig. 1. Network visualisation and possibilities of temporal dimension manipulation. (a) Tabular (raw) data; (b) Massive Sequence View (MSV). (c) Temporal Activity Map (TAM). (d) Node-link diagram showing the network state at time $t = 7$. (e) Temporal dimension manipulation – observation time sampling. (f) Temporal dimension manipulation – timeslicing.

When applied to real-world networks with a large amount of data, MSV suffer from visual clutter caused by overlapping edges, and thus important patterns may not be perceived. The *Temporal Activity Map (TAM)* [17], which omits all edges of the MSV layout and changes the shape of nodes from circles to squares for better sense of continuity, represents an alternative layout useful to identify patterns based in the node activity (Fig. 1(c)). However, when the edge visualisation is still needed or the amount of visual information in the layout is still large, these issues must be solved.

Animated visualisations. Contrary to timeline-based layouts, which show all timestamps of the network in a single image (time-to-space mapping [38]), animated layouts rely on a series of images that are used as animation frames (time-to-time mapping), each of them representing a different timestamp and containing the nodes and edges active at the moment. The matrix layout [40, 41], for example, visually represents the network adjacency matrix and can be used to analyse the network evolution if associated with animation. Another popular representation is the *node-link diagram* (Fig. 1(d)). In this layout, nodes are spatially placed on the screen with edges (lines) connecting them [5]. These layouts may also suffer from visual clutter due to the amount of information in real-world networks.

Improving layout readability. One strategy to reduce clutter and improve readability in layouts that explicitly show the edges is by changing the node positioning, which affects the length of the edges and, consequently, the number of overlaps and visible patterns. For this purpose, several node ordering algorithms have been proposed in the literature. For the MSV layout, examples include naive approaches, such as those based on the node appearance order, degree (in/out) and lexicographic, as well as more complex approaches, such as *Optimized MSV* [26], *Recurrent Neighbors* [17], and *Community-based Node Ordering (CNO)* [25]. These strategies, however, are not suitable

for streaming networks since they require all edges in primary memory.

Besides node positioning, sampling strategies can also improve layout readability by reducing the number of edges under analysis [25, 29, 42, 43]. In this case, not only edges can be sampled, but also nodes – only those edges connecting the sampled nodes are maintained. In the case of node-link diagrams, because of the difficulty to maintain the mental map, strategies for reducing visual clutter in this representation (through node positioning [44], edge-bundling [45], and others) are generally ineffective even if combined with animation [5].

Finally, a strategy suitable for reducing clutter in various layouts, regardless of the exhibition of edges or timeline/animation aspects, is to change the network temporal resolution scale through timeslicing strategies, an approach that is described in details in the following.

2.2. Network Timeslicing

Along with node positioning and sampling strategies, the temporal dimension can also be manipulated to improve layout readability. In this case, one possibility is to choose an observation time of interest (e.g., only the first or the second day of the network), as adopted by Zhao et al. [29] (see Fig. 1(e)). Another possibility is to change the network temporal resolution scale through timeslicing strategies. Timeslicing in our context means that events from subsequent timestamps will be grouped in a single timestamp [28]. The higher the temporal resolution scale, the more subsequent timestamps will be grouped into one and, as a consequence, the longer will be the time interval represented by the resulting timestamp.

Uniform timeslicing. In the simplest timeslicing approach (known as *uniform timeslicing*), the temporal resolution scale is a global and static value, so all timestamps of the network represent the same length of time without considering cases where

the network has non-stationary event distribution. As an example, if each timestamp in resolution 1 represents a 20-second interval, then each timestamp in resolution 2 will represent a 40-second interval. Linhares et al. [17] change the timestamp in which each event occurs by following a uniform timeslicing as follows (Eq. 1).

$$t_{\text{new}} = \left\lfloor \frac{t_{\text{ori}} - t_s}{\tau} \right\rfloor \tau + t_s \quad (1)$$

where t_{new} is the new timestamp of the event, t_{ori} is the timestamp of the event in the original temporal resolution, t_s is the first timestamp of the network and τ is the desired resolution scale. Repeated events (edges) are considered as a single one if their timestamps are merged. As a result, one may identify temporal patterns that would be difficult to see in the original resolution, especially in temporally sparse networks [17]. This timeslicing process is exemplified in Fig. 1(f), where in resolution 2 (new resolution defined by $\tau = 2$) each pair of adjacent timestamps from the original network (Fig. 1(b)) are merged into one, thus the events represented in (A,B,0) and (A,B,1) become a single event in resolution 2 (A,B,0) and so on.

In uniform timeslicing, such global and static temporal resolution scale is empirically chosen through initial exploratory analyses or by a domain specialist that knows *a priori* which one is adequate for the analysis given the event distribution. This scale does not consider the underlying network structure and thus may not faithfully represent the non-stationary distribution of events over time. Notwithstanding, in temporal networks all events are available in the visual analysis, so different scales may be tested until an adequate one is found.

In streaming scenarios, although one can adopt uniform timeslices as well, the employment of uniform approaches is even more difficult due to the non-stationary distribution of future events. Exploratory analysis may not be possible because usually there are no *a priori* data to support the decision. Since the event distribution can change, considering only an initial set of events in the stream to support the choice may be inefficient as well. This is often ignored when it is assumed a uniform event distribution (also referred as uniform event density), as if the events came in consecutive timestamps [10, 11, 12].

Nonuniform timeslicing. Given the many drawbacks of uniform timeslicing, it is necessary mechanisms to create timeslices that represent different lengths of time, which must be defined according to the non-stationary distribution of events over time. For a given temporal network, we call *nonuniform timeslicing* the presence of timestamps with different lengths.

3. Related Work

Timeslicing in network mining. An effective timeslicing is useful for areas beyond visualisation. Studies from the network (graph) mining community have shown that the timeslicing impacts the network structure and mining efficacy [46, 33]. To define suitable timeslices in this context, there are basically three strategies [33]. The first one comprehends the analysis of the accuracy obtained by different timeslicing choices in the task of interest (e.g., event prediction) [47]. Another, more popular,

involves the minimisation of the variance of a given network statistic (e.g., exponent of degree distribution) [48, 46, 49]. Not least, one can also define timeslices based on the similarity between the network structure from consecutive timestamps. This last approach has been used not only in mining tasks (e.g., [50]), but also in the visualisation of (spatial and non-spatial) temporal networks (e.g., [51, 52]).

Timeslicing in network visualisation. In (non-spatial) temporal network visualisation, several studies employ uniform timeslicing in the networks under analysis [17, 29, 5, 25, 28]. In [5], for instance, the authors analyse a high-school network considering each timestamp of the MSV layout as a three-minute interval whilst the original network temporal resolution is a 20-second interval per timestamp. In the same way, the Enron network [53] was analysed using a particular temporal resolution scale in [42] and a different one in [17].

Small MultiPiles [52] is a matrix-based visualisation that creates nonuniform timeslices in the form of piles containing timestamps with similar network structure (similar adjacency matrices). As the interval comprehended by each pile depends on changes in the network structure, the identification of topology-based patterns is facilitated. More recently, Wang et al. [33] proposed a nonuniform timeslicing method for temporal network visualisation that creates timeslices with a balanced number of events (equal visual complexity) by using an approach similar to the histogram equalisation, well-established in the discipline of digital image processing. Their strategy (hereafter named *Balanced Visual Complexity – BVC*) uses more timestamps to represent high-activity periods (with bursts of events) and less timestamps otherwise.

The method we propose in this paper goes in the opposite direction of BVC, i.e., we also consider that high-activity periods contain too much visual information, but we propose to represent them with higher resolution scales (consequently reducing the number of timestamps) instead of redistributing them in more timestamps. In the produced layout, the identification of global temporal patterns (e.g., birth and death of highly-active groups of nodes, bursts of events) is facilitated. Contrary to Small MultiPiles, our method is layout-agnostic, i.e., it can be used to enhance a variety of visualisations, from animated (e.g., node-link diagrams) to timeline-based ones (e.g., TAM, MSV). Finally, contrary to both BVC and Small MultiPiles, our method runs online and thus is suitable for streaming network analysis. To the best of our knowledge, no other study has proposed online and nonuniform timeslicing methods for network visualisation.

4. Online and nonuniform timeslicing method

The idea behind our method is that intervals with the same length but that have different event densities must be represented by different resolution scales. Having more events lead to higher resolutions and consequently in fewer timestamps, so the amount of visual information is reduced to an appropriate level in an attempt to optimise the identification of patterns.

Our method considers the number of events and their distribution on a fixed-size window w . The window is divided into

w_{size} slots. Each slot represents a timestamp in the current resolution and contains all events that exist in such timestamp. In the sequence, we use this window to decide the temporal resolution scale that will be applied to the next window (i.e., next timeslice). After computing such a new resolution scale, it is necessary to change the timestamp of incoming events accordingly. We summarise our approach in Algorithm 1 and provide details in the sequence.

Algorithm 1: Nonuniform timeslicing method

```

1 foreach timestamp  $t_i$  in the stream do
2   if end of a window then
3     compute new resolution  $\sigma_n$ ; // Eq. 2
4   end
5   foreach event  $e$  arriving in  $t_i$  do
6     if  $t_i$  does not belong to the first window then
7       change timestamp of  $e$  using  $\sigma_n$ ; // Eq. 4
8     end
9      $e.status \leftarrow$  processed;
10  end
11 end

```

Initially, we adopt the original resolution scale in the first window (cold start). From there, the resolution value σ_n of the next non-overlapping window considers two aspects of the current window: its resolution value and its event density (Eq. 2).

$$\sigma_n = \lfloor \delta \sigma_c + (1 - \delta) f_s(w_{size}) \rfloor \quad (2)$$

where δ ($0 \leq \delta \leq 1$) is a constant that determines the importance of the current resolution value (σ_c) in the computation of the new resolution. The component $f_s(w_{size})$ is presented in the following.

Recent elements are more relevant to the current state of the stream than old elements, so an approach commonly used to discount older information is the forgetting mechanism *fading sum* [54]. According to [55], the fading sum $S_{x,\alpha}(i)$ over the elements of a stream x is computed at time i as $S_\alpha(i) = x_i + \alpha S_\alpha(i-1)$, where $S_\alpha(1) = x_1$ and α ($0 \ll \alpha \leq 1$) is the fading factor, i.e., a constant such that the higher its value, the more importance is given to old elements. In our context, the component $f_s(w_{size})$ from Eq. 2 is a fading sum that quantifies the event density within the current window, while reducing the importance of old events inside the window. It is computed according to Eq. 3:

$$f_s(i) = \frac{x_i}{t_{wc}} + \alpha f_s(i-1) \quad (3)$$

where x_i is the number of events in the slot i of the window, t_{wc} is the number of slots in the window that presents at least one event (constant for a given window – if $t_{wc} = 0$ then the window has no events, so $\sigma_n = 0$), $f_s(1) = \frac{x_1}{t_{wc}}$ is the initial term. The higher the fading factor α , the more importance is given to old events and, consequently, the higher the resulting resolution scale. The rationale behind computing $f_s(w_{size})$ according to Eq. 3 is to obtain a weighted event density $d = f_s(w_{size})$, $d \in \mathbb{R}$,

that gives greater importance to recent events inside the window under analysis (through the fading factor component); the denser the most recent timestamps, the higher d .

Back to Eq. 2, if $\sigma_n = 0$, then σ_n is set as the average value of all past resolutions, so large inactivity periods (i.e. without events) may be represented by a resolution scale different from the original. With this decision, our method reduces not only the number of timestamps devoted to intervals with high density, but also the idle ones.

Within a new window, whose resolution scale is the computed σ_n , it is now necessary to change the timestamp attribute of each incoming event. To meet cases where exists an inactivity period between the last event of the previous window and the first event of the new window, we decompose the computation of the new timestamp t_{new} of an event e into two parts, one considering all past timestamps within the new window (which are, therefore, under resolution σ_n – see the first component of Eq. 4), and another considering all timestamps between the last event of the previous window and the beginning of the new window (which are, therefore, under a resolution that is (potentially) different from σ_n – see component t_{ref} in Eqs. 4 and 5).

$$t_{new}(e) = \left\lfloor \frac{t_{orig}(e) - t_{ini}}{\sigma_n} \right\rfloor + t_{ref} \quad (4)$$

where $t_{orig}(e)$ is the timestamp of e in the original resolution, t_{ini} is the first timestamp considered by the current window, and t_{ref} is the timestamp that acts as a reference in order to apply the resolution scale in inactive timestamps. The value of t_{ref} is computed only when dealing with the first event of a new resolution (new timeslice) and is defined according to Eq. 5:

$$t_{ref} = \left\lfloor \frac{t_{ini} - t_{orig}(e')}{\sigma_c} \right\rfloor + t_{new}(e') \quad (5)$$

where $t_{orig}(e')$ is the original timestamp of the last event from the previous window (event e') and $t_{new}(e')$ is the timestamp of e' in σ_c .

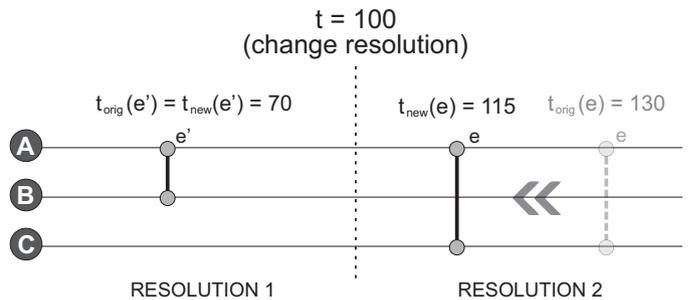


Fig. 2. Example of timeslicing using our proposal. In resolution 2, the timestamp of e is changed from 130 to 115.

Fig. 2 shows an example of how the timestamp of an event is changed according to Eq. 4. In this figure, $t_{orig}(e') = t_{new}(e') = 70$ because, up to this point, the original resolution scale (1) was maintained in the network. Note that a new window began at $t = t_{ini} = 100$ with resolution $\sigma_n = 2$. Any incoming event that belongs to this new window must therefore follow this resolution. In this sense, the timestamp attribute of the event e is changed from



Fig. 3. TAM layout showing four classes and all teachers of the Primary School network using resolution 1 (original). The interval between both days (from 5.21pm to 8.29am) does not present any edge and was omitted due to its size in the layout. Nodes are grouped according to the classes and grades. The “Tch” profile refers to the teachers of the school. The layout is horizontally large, dense and has few visible patterns, as, for example, the absence of Classes 4A and 4B near the end of the second day. Supplementary Fig. (a) shows the same image in a better quality and with the omitted interval.

$t_{orig}(e) = 130$ to a different value ($t_{new}(e)$). In this example, $\sigma_c = 1$ and $t_{ref} = \lfloor (100 - 70)/1 \rfloor + 70 = 100$ (Eq. 5). According to Eq. 4, $t_{new}(e) = \lfloor (130 - 100)/2 \rfloor + 100 = 115$. As expected, each timestamp in resolution 2 is twice the time interval represented by a timestamp from resolution 1, so $t_{orig}(e) = 130$ and $t_{new}(e) = 115$. Assuming an event e'' with $t_{orig}(e'') = 131$, then $t_{new}(e'')$ would be equal to 115 as well, and so on. As stated, Eq. 4 takes into account inactivity periods, respecting their occurrence in the converted timestamps.

As we consider only the most recent window of events in the computation, our method is capable of handling from temporal networks with few timestamps to potentially unbounded streaming networks. If the method presents high performance in the worst-case scenario of a highly-cluttered window, then every window with less or equal visual clutter will be also improved, regardless of the number of timestamps in the network.

5. Case Studies

In this section, we present visual analyses of two real-world temporal networks manipulated timestamp-by-timestamp to simulate streaming scenarios. In other words, the elements (edges/nodes) of each timestamp flow in and out in a way that the method (i) does not have access to all network data (therefore it is not possible to perform random accesses), (ii) does not know future elements or the number of timestamps, (iii) does not store past elements (except those from the current window, that are also discarded once the window is processed). Our goal is to compare our nonuniform timeslicing method against the original network resolution, uniform timeslicing approaches, and BVC [33].

In all analyses, we consider resolution 1 (Res. 1) as the original resolution of the network. After empirical tests, we defined $\delta = 0.2$ as the importance of the current resolution in the computation of the new one (see Eq. 2). To validate our method and illustrate its application, we rely on MSV, TAM and node-link diagrams. All visual analyses were performed using the software DyNetVis [56], which implements our method and all layouts presented in this section. DyNetVis is freely available at www.dynetvis.com.

5.1. Primary School

The first network, *Primary School* [57, 58], represents face-to-face interactions (contacts) involving teachers and students of a primary school between October 1st - 2nd of 2009. A face-to-face interaction is recognised by *Radio-Frequency Identification (RFID)* sensors from badges. Whenever two individuals are in front to each other (from 1 up to 1.5 meters), a contact

is registered in a time interval of 20 seconds. Such interval corresponds to the original temporal resolution of the network (Res. 1), i.e., each timestamp in Res. 1 comprises a 20-second interval. This network contains 242 nodes and 125,773 edges distributed in 5,846 timestamps in Res. 1. The data represent contacts from the first to fifth grade, with each grade having Classes A and B (for convenience, we will adopt terms such as “4A” and “4B” when referring to a certain class of a given grade). The majority of contacts occurs between students of the same class and each class has an assigned teacher [58].

Fig. 3 presents the TAM layout for four classes and all teachers of the Primary School network in resolution 1 (original). The nodes are grouped according to the classes and grades. The layout is horizontally large (due to the number of timestamps), which impairs the identification of global patterns and requires more screen space and scrolling, which impairs the user’s perception of temporal changes during the network evolution (mental map preservation [59]). Moreover, the layout is dense (a lot of events over time) and only a few patterns are easily identified, as, for example, the absence of Classes 4A and 4B students near the end of the second day. The network does not register contacts during sports activities [58], so it is possible to assume that these classes were involved in such activities or dismissed. Another possibility is that the students were taking exams or other activities without interacting with each other.

5.1.1. Parameter Analysis

To improve pattern identification, we applied our nonuniform timeslicing method in the Primary School network. Different values for α (fading factor) and w_{size} (window size) were evaluated and their impact in the adopted resolution scales is shown in Fig. 4. In the figure, the whole network is considered (activity in day 1, interval, activity in day 2). By comparing the plots in which the α value is the same ($\alpha = 0.9$ in (a,c) and $\alpha = 0.99$ in (b,d)), it is possible to see that a large window makes that the perception of changes in the number of events be late, delaying the resolution change. As a consequence, patterns related to these changes may be lost or identified only many timestamps later. This is especially relevant in streaming network analysis, in which the past data may have already been discarded. By comparing the plots in which the w_{size} is the same ($w_{size} = 50$ in (a,b) and $w_{size} = 200$ in (c,d)), one can notice higher resolution values when adopting higher α . This is expected since high α values increase the importance of old events. Finally, the plots show the resolution adopted in the interval between both days of the network, in which there is no event. This value is computed based on the average value of the past resolutions. This decision is related to the space of the layout required to represent such interval, which would be many times greater in the

original resolution.

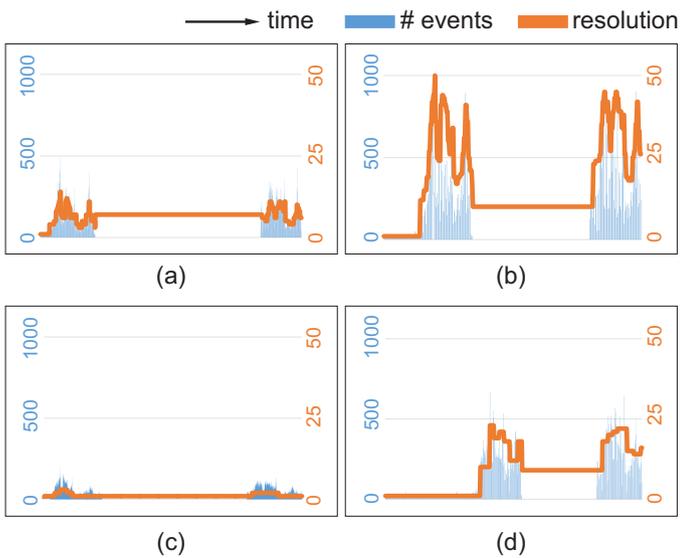


Fig. 4. Our nonuniform timeslicing and the relation between the adopted resolution scales and the event distribution for the Primary School network. (a) $w_{size} = 50$ and $\alpha = 0.9$ (1,443 timestamps). (b) $w_{size} = 50$ and $\alpha = 0.99$ (353 timestamps). (c) $w_{size} = 200$ and $\alpha = 0.9$ (4,880 timestamps). (d) $w_{size} = 200$ and $\alpha = 0.99$ (541 timestamps). The choice of the Fading Factor (α) and the window size (w_{size}) affects the resolution scale and, consequently, the layout and visible patterns.

5.1.2. Visual Analysis

Temporal Activity Map (TAM). Fig. 5 shows TAM layouts for different timeslicing scales considering the same four classes and teachers from Fig. 3. Fig. 5(a-e) shows TAM layouts for uniform timeslices adopting different resolutions (Res. 10, 25, 39, 100, and 200, respectively). Fig. 5(f) shows the TAM layout generated by our nonuniform timeslicing method ($w_{size} = 100$ and $\alpha = 0.99$, chosen empirically). Resolutions 10, 25, and 39 (Fig. 5(a-c)) were chosen because they represent the lower, the average and the higher resolution values adopted by our method for this network. Resolutions 100 and 200 (Fig. 5(d-e)) are arbitrary values. As expected, higher resolution values generate denser and (horizontally) smaller layouts, which impairs the visual analysis and the identification of patterns. Our method (Fig. 5(f)), however, automatically defines resolution scales that represent appropriate levels of visual density.

The adopted timeslicing strategy highly affects pattern identification. Fig. 6 presents visual analyses over the TAM layouts generated by our method (adopting $w_{size} = 100$ and $\alpha = 0.99$, Fig. 6(a)) and by uniform timeslices using resolutions 25 and 200 (Fig. 6(b,d), respectively). These are the same layouts from Fig. 5(f,b,e). The layout generated by BVC is also considered (Fig. 6(c) – for a high quality image of BVC’s layout, please see Supplementary Fig. (b)). In the best-case scenario, at least seven patterns can be identified: (1) all students from class 2B joined the network after the other classes and the group of teachers; (2) lunch break – several students go home for lunch, which reduces the number of nodes in such interval [58]; (3) there is no interaction involving class 2B students in a time interval near the end of the first day – probably due

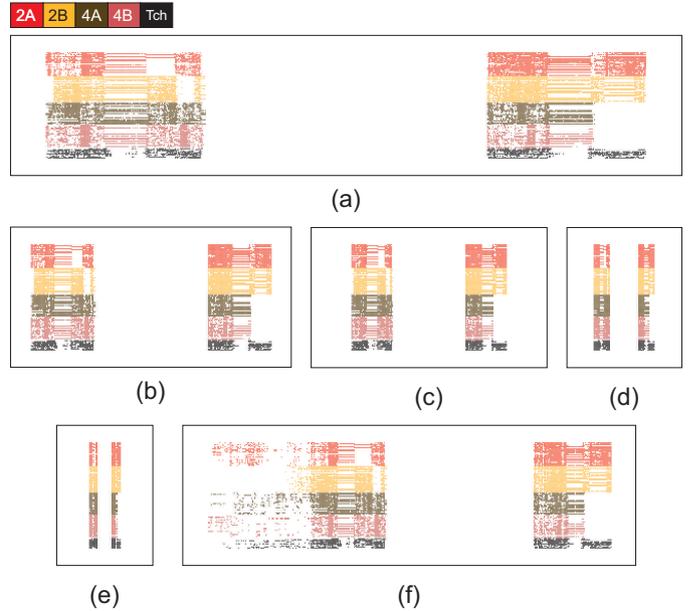


Fig. 5. TAM layouts showing four classes and all teachers of the Primary School network according to our nonuniform timeslicing method and five uniform resolution scales. (a) Res. 10. (b) Res. 25. (c) Res. 39. (d) Res. 100. (e) Res. 200. (f) Our method ($w_{size} = 100$ and $\alpha = 0.99$).

to sports activities [58]; (4) absence of Classes 4A and 4B students near the end of the second day; (5) two teachers left the network after lunch in the second day – probably the teachers from Classes 4A and 4B; (6) there are students that did not join the network in the first day; and (7) inactivity period due to the absence of classes (from 5.21pm to 8.29am).

Our method allows the identification of all seven patterns (Fig. 6(a)), being six of them considered as easy to found (1-5, 7). Pattern 6 is harder to identify because of the method’s cold start (adoption of resolution 1 at the beginning of the layout), which pollutes the layout and impairs the perception of this pattern. Although this original resolution serves only as a start point, considering it inside the layout facilitated the perception of pattern 1, that can also be noticed when adopting only resolution 1 in the analysis (see Fig. 3), but not as fast as with our proposal. Patterns 2, 5, and 6, on the other hand, cannot be identified with resolution 1 (Fig. 3). By adopting a uniform timeslicing using resolution 25 (Fig. 6(b)), all seven patterns can be identified as well, five of them being considered as easy to found (2-5, 7) and two of them being a little harder (1,6). Although this layout allows the identification of all patterns, recall that this resolution is the average value considered by our nonuniform method, which supports our method’s quality. By considering BVC in the network analysis, one may see the event redistribution caused by BVC’s histogram equalisation. As a consequence, pattern 7 is lost. Due to the number of timestamps, patterns 5-6 are also lost and patterns 1-3 are difficult to perceive. Only pattern 4 is considered easy to found. Such pattern, however, is also easy to identify with our method and with uniform resolutions 1 and 25. By using the uniform resolution 200 (Fig. 6(d)), patterns 1 and 3 are lost and only patterns 5-7 are considered as easy to identify. Note that pattern 6 is more easily perceived in this layout, so a uniform timeslic-

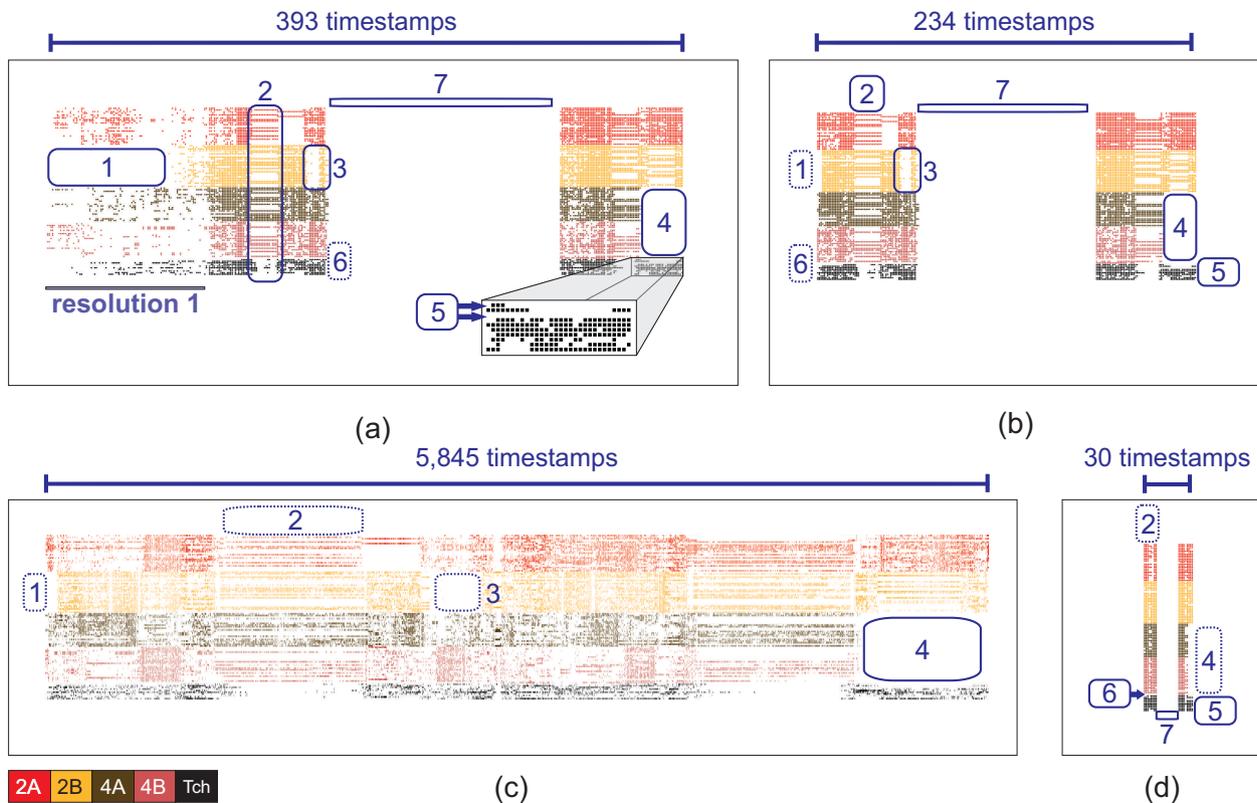


Fig. 6. Visible patterns in TAM layouts generated by different timeslicing approaches for the Primary School network. (a) Our method ($w_{size} = 100$ and $\alpha = 0.99$). (b) Res. 25. (c) BVC. (d) Res. 200. A maximum of seven patterns can be identified: (1) class 2B students joined the network after the others; (2) lunch break; (3) no interaction involving class 2B students near the end of the 1st day; (4) absence of Classes 4A and 4B near the end of the 2nd day; (5) two teachers left the network after lunch in the 2nd day; (6) some students did not join the network in the 1st day; and (7) inactivity period. Continuous rectangles represent patterns considered as easy to identify. Dotted rectangles represent patterns with difficult perception. Supplementary Fig. (b) shows BVC’s layout in a better quality.

ing that considers a higher resolution value may be useful in specific scenarios as well.

In summary, our method reduces the amount of visual information to an appropriate level that optimises the identification of global patterns that are lost or difficult to perceive with other strategies, including BVC. Our method for this network automatically divided the layout in 393 timestamps (against 5,845 from BVC) while preserving all seven analysed patterns. Less timestamps leads to less screen space and decreases the need of (horizontal) scrolling, which tends to facilitate the perception of temporal changes in the network (better mental map preservation). Considering uniform timeslicing, one should test different resolution scales until the better one is found. This approach, however, is only possible when dealing with (non-streaming) temporal networks (see Section 2.2). Our method not only provides adequate timeslices, but is suitable for streaming scenarios in which events are continuously arriving in non-stationary distribution.

Node-link diagram. Since streaming (online) analysis usually relies on animated visualisations, and given that BVC was originally designed for node-link diagrams [33], we also performed a comparison between our method and BVC considering animated node-link diagrams. Our goal is to analyse questions related to the current/recent development of the network. Recall that our method automatically defined a number of timestamps

that is many times fewer than with BVC (393 vs 5,845). We therefore applied a uniform timeslicing over the BVC’s result in order to have the same number of timestamps on both methods (393). As this would not be possible in real-world streaming scenarios, our analysis using TAM did not consider such a manual setting, i.e., the comparison took into account the different and automatically defined numbers of timestamps.

Fig. 7 shows a comparison between BVC (top) and our method (bottom) with respect to the end of the first day at primary school. With our method, one can easily see when the last students and teacher leave the school and the inactivity period starts ($t_k, t_k + 1, t_k + 2$). Only several timestamps later ($t_k + n$), students and teacher come back to school (second day in the network). With BVC, one cannot perceive the gradual decreasing of students at the end of the first day. Moreover, as expected with BVC, the inactivity period is totally lost: there are students/teachers in all snapshots from t_k to $t_k + n$.

In another comparison, Fig. 8 shows the students from 4A and 4B leaving the school on the second day. With both BVC (top) and our method (bottom), it is possible to see they leaving. Our method, however, allows one also to see a lot of interactions a few moments before (snapshot at t_j). Such burst of interactions (that involves all school classes and not only 2A, 2B, 4A, and 4B) refers to the lunch break. As can be seen with our method, students from 4A and 4B left the school just af-

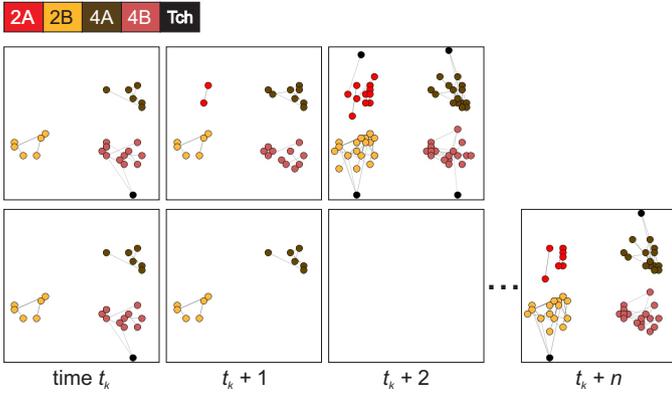


Fig. 7. Comparison between BVC (top) and our method (bottom) with respect to the end of the first day and the inactivity period at primary school. With our method it is possible to perceive the gradual decreasing of students at the end of the first day (time t_k and $t_k + 1$) and also when the inactivity period starts ($t_k + 2$).

ter lunch. This perception, that is not possible with the node-link diagram from BVC, is also supported by the analysis using TAM (Fig. 6, pattern 4).

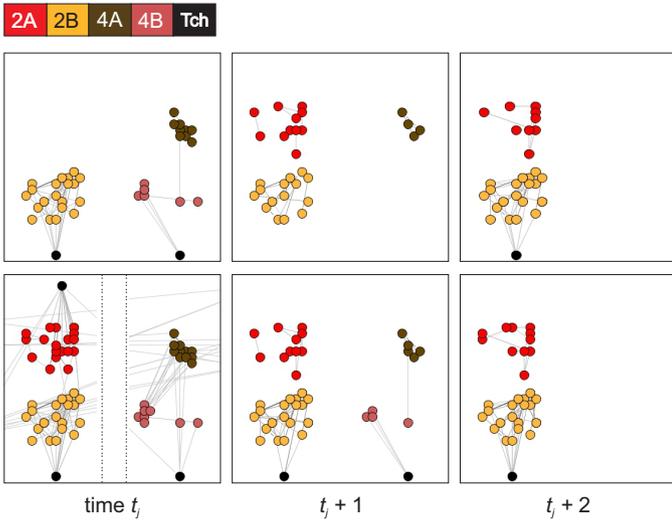


Fig. 8. Comparison between BVC (top) and our method (bottom) with respect to students from 4A and 4B leaving after lunch in the second day of school. With our method it is possible to identify a lot of interactions a few moments before the students leave school (snapshot at t_j).

Massive Sequence View (MSV). The visual analysis can be performed from a different perspective by showing only edges, as illustrated in Fig. 9, that shows the interactions involving Classes 2A, 2B, and 4A over a MSV layout generated by our nonuniform timeslicing ($w_{size} = 100$ and $\alpha = 0.99$). This layout reaffirms: (i) students from class 2B joined the network after the others; (ii) students from class 4A left the network earlier than the others in the second day; (iii) the absence of the majority of 2A students, as well as 2B students, during a period after the lunch break in the first day. Besides, this layout reveals new patterns, such as the perception that the only two students from class 2A that stayed in the network during the time interval after lunch in the first day connected to one another. Moreover,

the layout shows that students from one class have few interactions with students from other classes, with the majority of these interactions occurring during lunch. Not least, students from the 2nd grade interact more between themselves than with class 4A. This behaviour is also observed in the rest of the network (a lot of interactions among students of the same grade and few interactions involving different grades). These situations are expected in the network [58] and easily perceived in this layout.

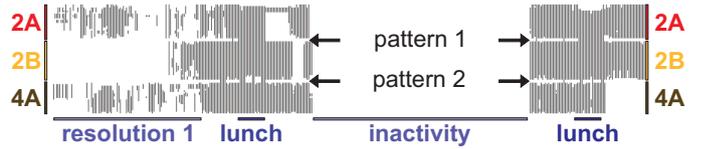


Fig. 9. MSV layout with our nonuniform timeslicing ($w_{size} = 100$ and $\alpha = 0.99$) showing the interactions among Classes 2A, 2B and 4A. Pattern 1: many interactions between 2A and 2B during lunch. Pattern 2: few interactions between 2B and 4A in the network.

5.1.3. Event Distribution

Fig. 10 shows the spread of events over time according to different timeslicing approaches: the original resolution, BVC, our method ($w_{size} = 100$ and $\alpha = 0.99$), and uniform Res. 25. The absence of events in the middle of plots (a,c,d) corresponds to the inactivity period between both days of the network. While BVC (Fig. 10(b)) changes the event distribution because of its histogram equalisation, our method (Fig. 10(c)) provides a distribution similar to those from uniform approaches (Fig. 10(a,d)). Since our timeslicing adopts the original resolution in the first window (cold start), one may see a “shift” in the time dimension at the plot (Fig. 10(c)).

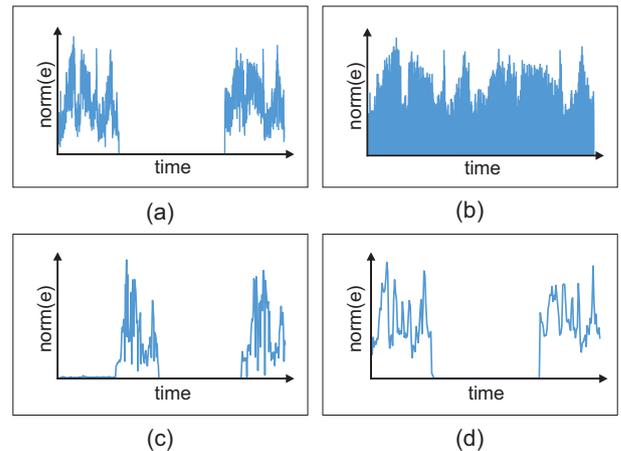


Fig. 10. Spread of events according to different timeslicing approaches for the Primary School network. (a) Res. 1. (b) BVC. (c) Our method ($w_{size} = 100$ and $\alpha = 0.99$). (d) Res. 25. “norm(e)” refers to the normalisation of the number of events to values between 0 and 1.

Fig. 11 shows the empirical cumulative distribution function (ECDF) considering the events from our method’s layout ($w_{size} = 100$ and $\alpha = 0.99$, Fig. 11(a)) and from resolution 1’s layout (Fig. 11(b)). Our method produces less timestamps without events when compared with Res. 1 (36.6% vs 47% – blue

dotted lines), which is justified by the resolution scale used in inactivity periods, that is different from the original. Furthermore, 25.4% of the time contains very few events in our layout (cold start window). By observing the third quartile (red dotted lines), after 75% of the time our layout contains a maximum of 213 events per timestamp (26% of the maximum number of events per timestamp), while in Res. 1 the number of events per timestamp is almost 43% of the maximum number of events per timestamp (40 out of 94 events).

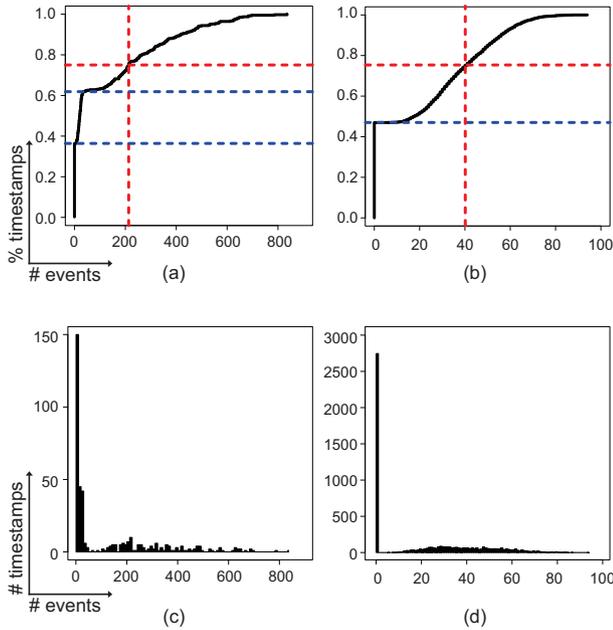


Fig. 11. Empirical cumulative distribution function (ECDF) and event distribution (ED) considering the events from the Primary School network. (a) ECDF our method ($w_{size} = 100$ and $\alpha = 0.99$). (b) ECDF Res. 1. (c) ED our method (393 timestamps, $w_{size} = 100$ and $\alpha = 0.99$). (d) ED Res. 1 (5,846 timestamps).

5.2. Enron

The second network, *Enron* [60, 53], contains email communications from Enron Inc., a former energy company involved in the biggest American accounting fraud [50]. The network is composed of 148 nodes and 24,667 edges distributed in 1,346 timestamps, each representing one day in Res. 1 (from Nov. 4th 1998 to July 11th 2002) [17]. Enron was studied by several works in literature, from offline visual analysis using MSV (e.g., [17, 26, 29]) to streaming-fashion mining tasks (e.g., [50]). Unlike the Primary School network, whose number of events varies a lot in each day and which contains a large time interval without any event (the period between the two days), the Enron network presents a growing number of events over time. We applied our method in the network to analyse the evolution of the resolution under this circumstance.

5.2.1. Parameter Analysis

Fig. 12 presents our method's behaviour under different values of w_{size} and α for the Enron network. Comparing the plots (a,c), it is possible to see the impact of the fading factor in the resolution computation. As can be seen, the high number of

events near the end of the network is reflected in the timeslicing for the two α values tested. Comparing the plots (b,c,d), one can see how frequent the timeslicing occurs according to the window size. As discussed, large windows make the change in the resolution scale less frequent and, as a consequence, each resolution may not faithfully represent the different number of events and their distribution. One can see such situation occurring in the Enron network by analysing the resolution evolution under $w_{size} = 200$ and $\alpha = 0.99$ (Fig. 12(d)): at the end of the network, the number of events decreases abruptly, but the resolution scale remains high.

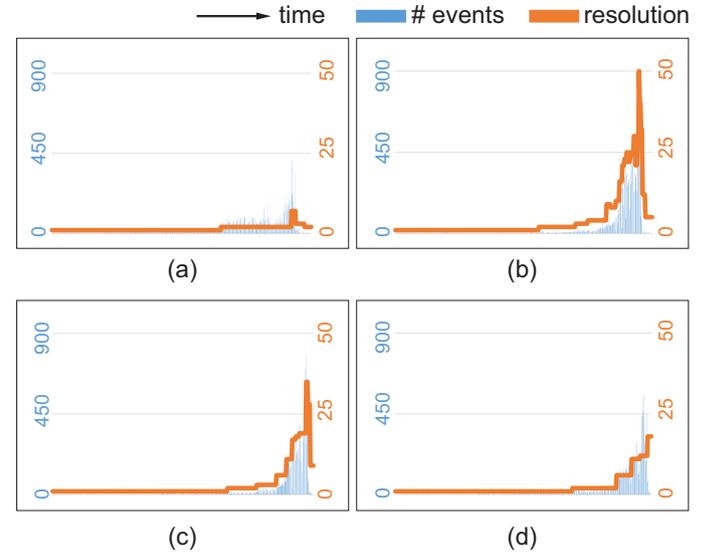


Fig. 12. Our nonuniform timeslicing and the relation between the adopted resolution scales and the event distribution for the Enron network. (a) $w_{size} = 100$ and $\alpha = 0.9$ (921 timestamps). (b) $w_{size} = 50$ and $\alpha = 0.99$ (357 timestamps). (c) $w_{size} = 100$ and $\alpha = 0.99$ (448 timestamps). (d) $w_{size} = 200$ and $\alpha = 0.99$ (579 timestamps).

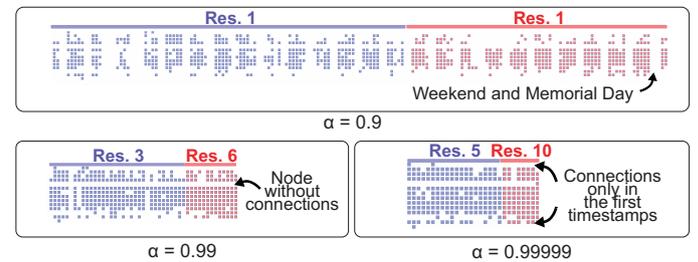


Fig. 13. Impact of different fading factor (α) values on the layout ($w_{size} = 100$). Different α lead to different visual patterns. The change in the node colour represents a change in the resolution scale (new timeslice). Node ordering defined by Recurrent Neighbors [17] using resolution 1.

Fig. 13 shows an approximation of the same time interval (near Dec. 12th, 1999 to near May 31th, 2000)¹ and the same group of nodes in three distinct layouts obtained by adopting

¹Since the resolution scale may aggregate different days in a single timestamp, the first timestamp may also represent few days before the first day of the interval depending on the adopted resolution. In the same way, the last timestamp may also represent few days after the last day of the interval.

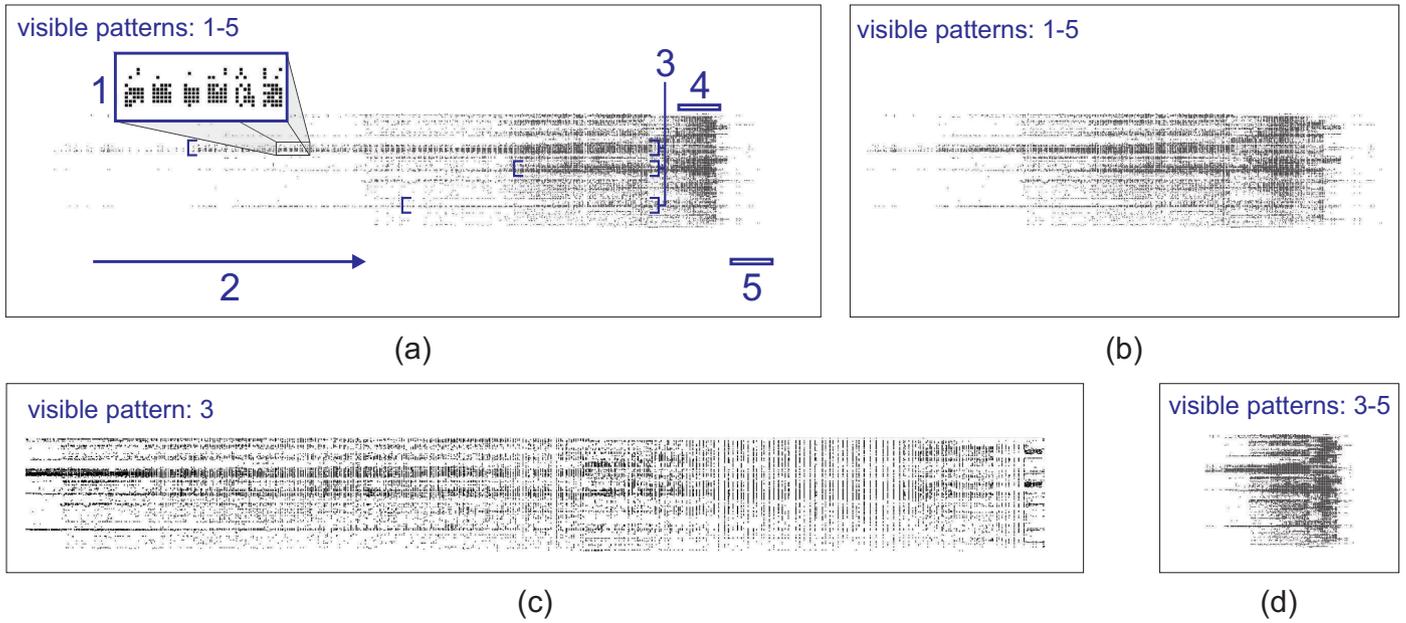


Fig. 14. TAM layouts generated by different timeslicing approaches and their visible patterns in the Enron network. (a) Our method (921 timestamps, $w_{size} = 100$ and $\alpha = 0.9$). (b) Resolution 2 (673 timestamps). (c) BVC (1,345 timestamps). (d) Resolution 7 (193 timestamps). Depending on the layout, a maximum of five visual patterns can be identified: (1) weekdays, in which there are interactions among nodes, and weekends, that are days without interactions; (2) perception of the growing number of events over time; (3) identification of highly active groups of nodes; (4) a time interval with a burst of events near the end of the network; and (5) abrupt decrease in the number of events followed by the end of the network. Node ordering defined by Recurrent Neighbors [17] using resolution 1.

$w_{size} = 100$ and three different fading factor values ($\alpha = 0.9$, $\alpha = 0.99$ and $\alpha = 0.99999$). The first layout, with $\alpha = 0.9$, maintained the original resolution scale during the whole interval. By doing so, each timestamp refers to a 1-day interval and so it was possible to identify days without events. Such days are usually weekends and holidays, such as the highlighted weekend May 28th – 29th, 2000 and holiday May 30th, 2000 (Memorial Day). By adopting $\alpha = 0.99$, one can see that the weekend/holiday pattern is lost due to the aggregation of days in a single timestamp. Another pattern, however, is revealed: it is easier to identify a node without interactions, i.e., a person in the company that did not receive or send any emails in this period. By analysing the layout with $\alpha = 0.99999$, it is possible to see that the node without interactions from the previous layout appears in the network in the last timestamp. Moreover, one can notice that the first and the last nodes of the layout had interactions only in the first timestamps. These last two patterns are visible in all three layouts, but they are more easily perceived in the layouts with higher α values.

5.2.2. Visual Analysis

Fig. 15 shows TAM layouts for the Enron network considering different timeslicing approaches. Fig. 15(a-h) shows the TAM layouts for uniform timeslicing using resolutions 1, 2, 7, 15, 25, 50, 75, and 100, respectively. Fig. 15(i) shows the TAM layout generated by our method ($w_{size} = 100$ and $\alpha = 0.9$). Resolutions 1, 2, and 7 (Fig. 15(a-c)) are shown because they represent the lower (and original), the average and the higher resolution scales adopted by our method for this network. The other resolutions (Fig. 15(c-h)) are arbitrary values.

As illustrated in Fig. 14, depending on the timeslicing be-

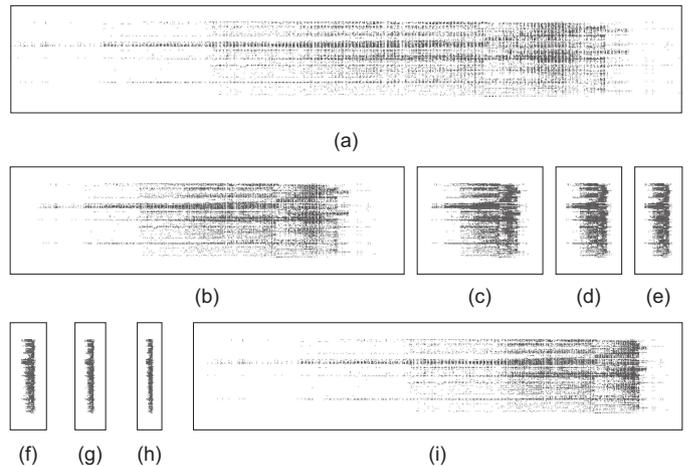


Fig. 15. TAM layouts for the Enron network considering different resolution scales. (a) Res. 1. (b) Res. 2. (c) Res. 7. (d) Res. 15. (e) Res. 25. (f) Res. 50. (g) Res. 75. (h) Res. 100. (i) Our method ($w_{size} = 100$ and $\alpha = 0.9$). Node ordering defined by Recurrent Neighbors [17] using resolution 1.

ing used, more or less patterns can be identified. The layout generated by our method allows the identification of at least 5 patterns (Fig. 14(a)): (1) weekdays, in which there are interactions among nodes, and weekends (without interactions); (2) perception of the growing number of events over time; (3) identification of highly active groups of nodes; (4) a time interval with a burst of events near the end of the network; and (5) abrupt decrease in the number of events followed by the end of the network. The uniform timeslicing using resolution 2 (Fig. 14(b)) also allows the perception of all five patterns. However, recall that this resolution represents the average value adopted

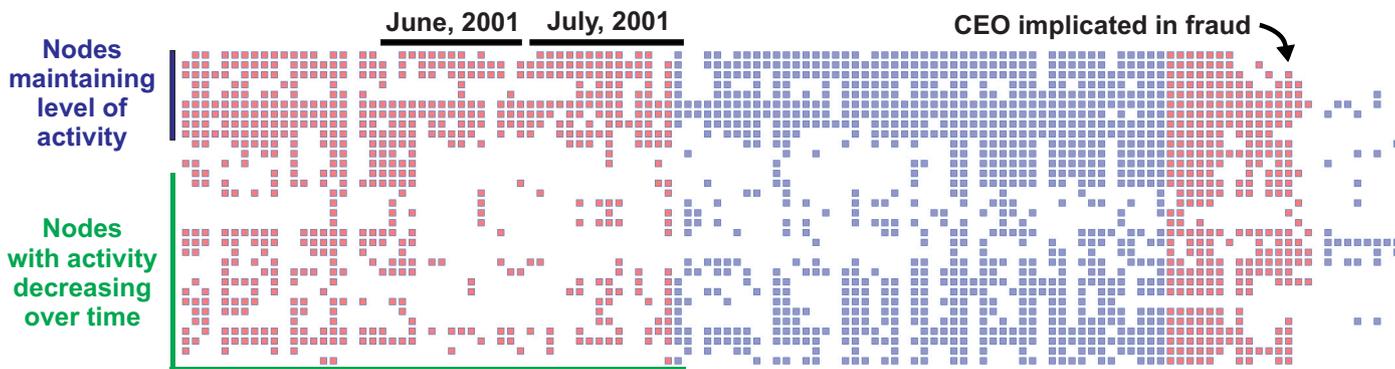


Fig. 16. TAM layout generated by our method ($w_{size} = 100$ and $\alpha = 0.9$) showing a portion of the network. Two patterns are visible: (i) a decrease in the number of events in June and July 2001; and (ii) an abrupt decrease in the number of events followed by the end of the network. The change in the node colour represents a change in the resolution scale (new timeslice). Node ordering defined by Recurrent Neighbors [17] using resolution 1.

by our method, which supports the claim that it chooses resolution scales that are indeed suitable for the network analysis. As expected, BVC redistributed the events along the timestamps, and so these temporal patterns (all but pattern 3) are lost (Fig. 14(c)). By using resolution 7 in a uniform timeslicing (Fig. 14(d)), patterns 1 and 2 are lost. Each timestamp in this resolution represents 7 days and so there is no separation of weekdays and weekends or the perception of growing node activity. One can note that layouts with temporal resolutions above 7 (see Fig. 15(d-h)) are even worse for the Enron network visual analysis.

The ideal timeslicing depends on the network being analysed. For the Primary School network, the uniform timeslicing using resolution 25 allowed the identification of several patterns (see Fig. 6(b)). On the other hand, resolution 25 is not a good choice for the Enron network (Fig. 15(d)). In the same way, a uniform timeslicing using resolution 2 would not improve Primary School analysis. Our method is capable of adapting the resolution scale according to the number and distribution of events, thus enhancing the network visual analysis.

Fig. 16 presents two other patterns observed in the layout generated by our method when *zooming in* the time interval with a burst of events showed in Fig. 14(a). According to Sun et al. [50], in June 2001 occurred an important company episode related to the Enron accounting fraud: “*Rove divests his stocks in energy*”. In the layout, it is possible to see a decrease in the number of events (emails) in the majority of the days in June and July involving the majority of the nodes. Such pattern may be related to this important episode. The layout also shows the moment in which there is an abrupt decrease in the number of events followed by the end of the network. Such decrease is related to another company episode, “*Lay [Enron CEO] implicated in plot to inflate profits and hide losses*” [50], which happened in Feb 4th, 2002. After the decrease of events, our method changed the resolution scale from 7 to 3, reflecting the new number of events. Temporal patterns such as these are probably lost when using BVC because of its event redistribution (Fig. 17(b)). Our method (Fig. 17(c)), on the other hand, provides a distribution similar to those from uniform approaches (Fig. 17(a,d)), and thus is capable of highlighting such temporal patterns.

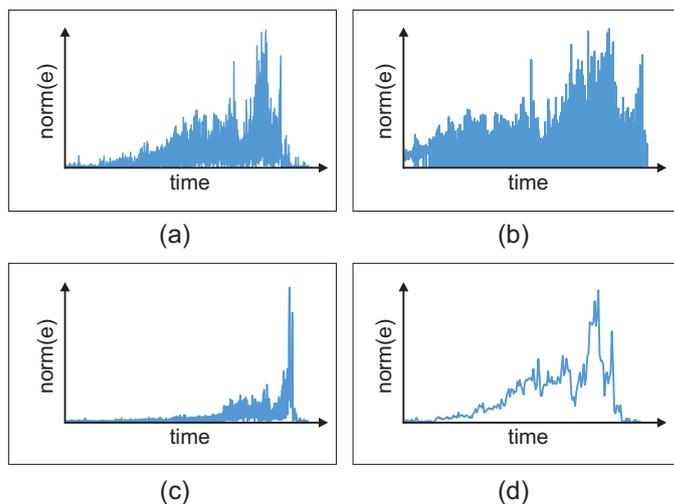


Fig. 17. Spread of events over time according to different timeslicing approaches for the Enron network. (a) Original resolution (1,346 timestamps). (b) BVC (1,345 timestamps). (c) Our method (921 timestamps, $w_{size} = 100$ and $\alpha = 0.9$). (d) Res. 25 (193 timestamps). While BVC changes the event distribution because of its histogram equalisation procedure, our method provides a distribution similar to those from uniform approaches. “norm(e)” refers to the normalisation of the number of events to values between 0 and 1.

6. Limitations

Method. When timeslicing, one should be aware that events may be lost to improve network comprehension (due to the consecutive timestamp grouping), and so relevant information may be lost in the process. Such characteristic exists in any other sampling strategy. Our proposal, however, considers the number of events and maintains their non-stationary distribution in an attempt to reduce such impairment.

Our method takes into account only event density to define the temporal resolution scales. As the network topology is not considered in the process, the density would be the same independently of the nodes involved in the connections. Although the network structure is not considerably affected by small changes in the temporal resolution [28, 5], we did not measure the intensity of changes in the structure caused by the resolution scales our method can choose. This study would al-

low us to establish an upper limit for the resolution scales or lead to a new method that would consider the network topology.

Misleading conclusions. Since two timestamps in the layout may represent completely different time intervals, one should pay attention in the resolution scale adopted in each of them when the task depends on this information (e.g., to decide which node has been active for the longest time in the network). Changes in the node colour, as used in the experiments, and an analysis using a line graph that depicts the resolution scale evolution (recall Figs. 4 and 12) attenuate this limitation, but other visual encodings can be used. In such cases, where the nonuniform timeslicing impairs the analysis, our method remains useful as the average resolution scale computed by it represents a good choice for a uniform timeslicing (as occurred with resolution 25 in our Primary School analysis).

Layout readability. Although our method improves the layout by manipulating the network temporal dimension, the node positioning represents another important aspect that has to be considered, since the ordering quality may impact layout readability. We thus recommend the adoption of a high-quality node positioning method. Eventually, the joint employment of sampling strategies may be required.

Visualisation techniques. We have demonstrated our method's quality using node-link diagram, TAM, and MSV. Although our method runs online, these last two visual representations draw the network elements (nodes and/or edges) in an offline manner. This is a characteristic of these layouts and not a limitation of our method. Although they could be adapted to handle online scenarios by plotting consecutive windows over time, this adaptation is out of the scope of this paper. As demonstrated, our method does not rely on particular layouts' characteristics (e.g., length/positioning of edges or animated vs timeline layouts) and thus could be applied in different layouts as well. In animated layouts, however, the visual analysis would probably be impaired in some cases, since the number of frames devoted to high-activity periods would reduce, potentially breaking the user's mental map.

7. Conclusion

We proposed in this paper an online and nonuniform timeslicing method for network visualisation that highlights temporal patterns such as bursts of events, highly-active groups of nodes, and others. It enhances visual analysis of both temporal and streaming networks and can be used with a variety of visualisation techniques, for example, TAM, node-link diagrams, and MSV.

Without our method, when handling temporal networks one should test different uniform timeslices until the "less worst" is found. Besides the effort of such preliminary tests, analyses of different networks require different temporal resolutions. For streaming scenarios, exploratory analysis to support the timeslicing may not be possible because events are arriving online

and in non-stationary distribution. For the same reason, considering an initial set of events to support the choice of the resolution scale may be inefficient as well.

Our method considers the number of events and their distribution to adapt the layout. This is possible because the choice of each new resolution scale uses only events from a sliding window, with old information being discounted by a forgetting mechanism. The method has low time and spatial computational complexity, since there is no need for various scans in the data and edges can be discarded once they are processed. In our experiments, we have analysed two real-world networks with different characteristics and the results show that the resolution scales automatically adopted are indeed suitable for each network analysis.

As future work, we intend to perform user experiments to validate our method considering the quality of the produced layout and the decisions regarding how easy the users perceive changes in the resolution scale. Besides, the choice of both window size and fading factor value directly affects the layout. These are currently user-dependent parameters and we will try to automate them. We also intend to measure how much information is lost when the network temporal resolution is changed and how this affects the network topology.

Acknowledgments

This research was supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq [grant number 456855/2014-9], Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES PrInt - Grant number 88881.311513/ 2018-01), and São Paulo Research Foundation (FAPESP - Grant number 2020/10049-0).

References

- [1] Boccaletti, S, Latora, V, Moreno, Y, Chavez, M, Hwang, DU. Complex networks: Structure and dynamics. *Physics Reports* 2006;424(4):175–308. URL: <https://www.sciencedirect.com/science/article/pii/S037015730500462X>. doi:<https://doi.org/10.1016/j.physrep.2005.10.009>.
- [2] Albert, R, Barabási, AL. Statistical mechanics of complex networks. *Reviews of Modern Physics* 2002;74(1):47–97. doi:10.1103/revmodphys.74.47.
- [3] Estrada, E. *Introduction to Complex Networks: Structure and Dynamics*. Cham: Springer International Publishing. ISBN 978-3-319-11322-7; 2015, p. 93–131. URL: https://doi.org/10.1007/978-3-319-11322-7_3. doi:10.1007/978-3-319-11322-7_3.
- [4] Holme, P, Saramäki, J. Temporal networks. *Physics Reports* 2012;519(3):97–125.
- [5] Linhares, CDG, Ponciano, JR, Paiva, JGS, Travençolo, BAN, Rocha, LEC. *Visualisation of Structure and Processes on Temporal Networks*. Cham: Springer International Publishing. ISBN 978-3-030-23495-9; 2019, p. 83–105. URL: https://doi.org/10.1007/978-3-030-23495-9_5. doi:10.1007/978-3-030-23495-9_5.
- [6] Ahmed, NK, Neville, J, Kompella, R. Network sampling: From static to streaming graphs. *ACM Trans Knowl Discov Data* 2013;8(2):7:1–7:56. doi:10.1145/2601438.
- [7] McGregor, A. Graph mining on streams. *Encyclopedia of Database Systems* 2009;;1271–1275.
- [8] Estrada, E. Communicability in temporal networks. *Physical Review E* 2013;88(4):042811.

- [9] Zhang, J. A Survey on Streaming Algorithms for Massive Graphs. Boston, MA: Springer US. ISBN 978-1-4419-6045-0; 2010, p. 393–420. URL: https://doi.org/10.1007/978-1-4419-6045-0_13. doi:10.1007/978-1-4419-6045-0_13.
- [10] Crnovrsanin, T, Chu, J, Ma, KL. An incremental layout method for visualizing online dynamic graphs. *J Graph Algorithms Appl* 2017;21(1):55–80.
- [11] Ahmed, NK, Duffield, N, Willke, TL, Rossi, RA. On sampling from massive graph streams. *Proc VLDB Endow* 2017;10(11):1430–1441. URL: <https://doi.org/10.14778/3137628.3137651>. doi:10.14778/3137628.3137651.
- [12] Etemadi, R, Lu, J. Pes: Priority edge sampling in streaming triangle estimation. *IEEE Transactions on Big Data* 2019;:1–1doi:10.1109/TBDATA.2019.2948613.
- [13] Bu, Z, Wang, Y, Li, HJ, Jiang, J, Wu, Z, Cao, J. Link prediction in temporal networks: Integrating survival analysis and game theory. *Information Sciences* 2019;498:41 – 61. URL: <http://www.sciencedirect.com/science/article/pii/S0020025519304529>. doi:<https://doi.org/10.1016/j.ins.2019.05.050>.
- [14] Jo, HH, Hiraoka, T. *Bursty Time Series Analysis for Temporal Networks*. Cham: Springer International Publishing. ISBN 978-3-030-23495-9; 2019, p. 161–179. URL: https://doi.org/10.1007/978-3-030-23495-9_9. doi:10.1007/978-3-030-23495-9_9.
- [15] Card, SK, Mackinlay, JD, Shneiderman, B. Readings in information visualization: using vision to think. Morgan Kaufmann; 1999.
- [16] Ware, C. *Information Visualization (Third Edition)*. Interactive Technologies; Boston: Morgan Kaufmann. ISBN 978-0-12-381464-7; 2013, p. 514. URL: <https://www.sciencedirect.com/science/article/pii/B9780123814647000181>. doi:<https://doi.org/10.1016/B978-0-12-381464-7.00018-1>.
- [17] Linhares, CDG, Travençolo, BAN, Paiva, JGS, Rocha, LEC. Dynetvis: A system for visualization of dynamic networks. In: *Proceedings of the Symposium on Applied Computing. SAC '17; Marrakech, Morocco: ACM*. ISBN 978-1-4503-4486-9; 2017, p. 187–194. doi:10.1145/3019612.3019686.
- [18] v. d. Elzen, S, Holten, D, Blaas, J, van Wijk, JJ. Dynamic network visualization with extended massive sequence views. *IEEE Transactions on Visualization and Computer Graphics* 2014;20(8):1087–1099. doi:10.1109/TVCG.2013.263.
- [19] Ponciano, JR, Linhares, CDG, Melo, SL, Lima, LV, Travençolo, BAN. Visual analysis of contact patterns in school environments. *Informatics in Education* 2020;19(3):455–472. doi:10.15388/infedu.2020.20.
- [20] Gorochowski, TE, di Bernardo, M, Grierson, CS. Using aging to visually uncover evolutionary processes on networks. *IEEE Transactions on Visualization and Computer Graphics* 2012;18(8):1343–1352. doi:10.1109/TVCG.2011.142.
- [21] Sarmento, R, Cordeiro, M, Gama, J. Visualization for streaming networks. In: *Proceedings of the 3rd Workshop on New Frontiers in Mining Complex Patterns (NFMCP 2014)*. 2014, p. 62–74.
- [22] Dang, TN, Pendar, N, Forbes, AG. Timearcs: Visualizing fluctuations in dynamic networks. In: *Computer Graphics Forum; vol. 35. Wiley Online Library*; 2016, p. 61–69.
- [23] Linhares, CD, Ponciano, JR, Pereira, FS, Rocha, LE, Paiva, JGS, Travençolo, BA. Visual analysis for evaluation of community detection algorithms. *Multimedia Tools and Applications* 2020;:1–23.
- [24] Battista, GD, Eades, P, Tamassia, R, Tollis, IG. Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry* 1994;4(5):235 – 282.
- [25] Linhares, CD, Ponciano, JR, Pereira, FS, Rocha, LE, Paiva, JGS, Travençolo, BA. A scalable node ordering strategy based on community structure for enhanced temporal network visualization. *Computers & Graphics* 2019;84:185 – 198. doi:<https://doi.org/10.1016/j.cag.2019.08.006>.
- [26] van den Elzen, S, Holten, D, Blaas, J, van Wijk, JJ. Reordering massive sequence views: Enabling temporal and structural analysis of dynamic networks. In: *2013 IEEE Pacific Visualization Symposium (PacificVis)*. 2013, p. 33–40.
- [27] Lee, E, Moody, J, Mucha, PJ. *Exploring Concurrency and Reachability in the Presence of High Temporal Resolution*. Cham: Springer International Publishing. ISBN 978-3-030-23495-9; 2019, p. 129–145. URL: https://doi.org/10.1007/978-3-030-23495-9_7. doi:10.1007/978-3-030-23495-9_7.
- [28] Rocha, LEC, Masuda, N, Holme, P. Sampling of temporal networks: Methods and biases. *Phys Rev E* 2017;96:052302. doi:10.1103/PhysRevE.96.052302.
- [29] Zhao, Y, She, Y, Chen, W, Lu, Y, Xia, J, Chen, W, et al. Eod edge sampling for visualizing dynamic network via massive sequence view. *IEEE Access* 2018;6:53006–53018. doi:10.1109/ACCESS.2018.2870684.
- [30] Gomez, JC, Mascarenhas, E, Rego, V. The clam approach to multithreaded communication on shared-memory multiprocessors: Design and experiments. *IEEE Transactions on Parallel and Distributed Systems* 1998;9(1):36–49.
- [31] Rajah, K, Ranka, S, Xia, Y. Advance reservations and scheduling for bulk transfers in research networks. *IEEE Transactions on Parallel and Distributed Systems* 2009;20(11):1682–1697.
- [32] Dietz, L, Dalton, J, Balog, K. Time-aware evaluation of cumulative citation recommendation systems. In: *SIGIR 2013 Workshop on Time-aware Information Access (TAIA2013); vol. 10*. 2013, p. 2484028–2491802.
- [33] Wang, Y, Archambault, D, Haleem, H, Moeller, T, Wu, Y, Qu, H. Nonuniform timeslicing of dynamic graphs based on visual complexity. In: *2019 IEEE Visualization Conference (VIS)*. IEEE; 2019, p. 1–5.
- [34] Bach, B. Unfolding dynamic networks for visual exploration. *IEEE Computer Graphics and Applications* 2016;36(2):74–82. doi:10.1109/MCG.2016.32.
- [35] Bach, B, Pietriga, E, Fekete, JD. Visualizing dynamic networks with matrix cubes. In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 2014, p. 877–886.
- [36] Eades, P, Tamassia, R. Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry* 1994;4(5):235 – 282. doi:[https://doi.org/10.1016/0925-7721\(94\)00014-X](https://doi.org/10.1016/0925-7721(94)00014-X).
- [37] Holten, D, Cornelissen, B, van Wijk, JJ. Trace visualization using hierarchical edge bundles and massive sequence views. In: *2007 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis*. 2007, p. 47–54. doi:10.1109/VISSOF.2007.4290699.
- [38] Beck, F, Burch, M, Diehl, S, Weiskopf, D. A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum* 2017;36(1):133–159. doi:10.1111/cgf.12791.
- [39] Longabaugh, WJ. Combing the hairball with biofabric: a new approach for visualization of large networks. *BMC bioinformatics* 2012;13(1):1–16.
- [40] Behrisch, M, Bach, B, Henry Riche, N, Schreck, T, Fekete, JD. Matrix reordering methods for table and network visualization. *Comput Graph Forum* 2016;35(3):693–716. doi:10.1111/cgf.12935.
- [41] Rufiange, S, Melançon, G. Animatrix: A matrix-based visualization of software evolution. In: *2014 Second IEEE Working Conference on Software Visualization*. 2014, p. 137–146.
- [42] Zhao, Y, Chen, W, She, Y, Wu, Q, Peng, Y, Fan, X. Visualizing dynamic network via sampled massive sequence view. In: *Proceedings of the 12th International Symposium on Visual Information Communication and Interaction. VINCI'2019; New York, NY, USA: ACM*. ISBN 978-1-4503-7626-6; 2019, p. 32:1–32:2. URL: <http://doi.acm.org/10.1145/3356422.3356454>. doi:10.1145/3356422.3356454.
- [43] Sarmento, R, Cordeiro, M, Gama, J. Streaming networks sampling using top-k networks. In: *Proceedings of the 17th International Conference on Enterprise Information Systems - Volume 1. ICEIS 2015; Setubal, PRT: SCITEPRESS - Science and Technology Publications, Lda*. ISBN 9789897580963; 2015, p. 228–234. URL: <https://doi.org/10.5220/0005341402280234>. doi:10.5220/0005341402280234.
- [44] Six, JM, Tollis, IG. A framework and algorithms for circular drawings of graphs. *Journal of Discrete Algorithms* 2006;4(1):25 – 50. URL: <http://www.sciencedirect.com/science/article/pii/S1570866705000031>. doi:<https://doi.org/10.1016/j.jda.2005.01.009>.
- [45] Lhuillier, A, Hurter, C, Telea, A. Ffteb: Edge bundling of huge graphs by the fast fourier transform. In: *2017 IEEE Pacific Visualization Symposium (PacificVis)*. 2017, p. 190–199. doi:10.1109/PACIFICVIS.2017.8031594.
- [46] Uddin, S, Choudhury, N, Farhad, SM, Rahman, MT. The optimal window size for analysing longitudinal networks. *Scientific reports* 2017;7(1):1–15.
- [47] Fish, B, Caceres, RS. A task-driven approach to time scale detection in dynamic networks. In: *Proceedings of the 13th International Workshop on Mining and Learning with Graphs (MLG)*. 2017,.

- [48] Sulo, R, Berger-Wolf, T, Grossman, R. Meaningful selection of temporal resolution for dynamic networks. In: Proceedings of the Eighth Workshop on Mining and Learning with Graphs. MLG '10; New York, NY, USA: Association for Computing Machinery. ISBN 9781450302142; 2010, p. 127–136. URL: <https://doi.org/10.1145/1830252.1830269>. doi:10.1145/1830252.1830269.
- [49] Soundarajan, S, Tamersoy, A, Khalil, EB, Eliassi-Rad, T, Chau, DH, Gallagher, B, et al. Generating graph snapshots from streaming edge data. In: Proceedings of the 25th International Conference Companion on World Wide Web. WWW '16 Companion; Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee. ISBN 9781450341448; 2016, p. 109–110. URL: <https://doi.org/10.1145/2872518.2889398>. doi:10.1145/2872518.2889398.
- [50] Sun, J, Faloutsos, C, Faloutsos, C, Papadimitriou, S, Yu, PS. Graphscope: Parameter-free mining of large time-evolving graphs. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '07; New York, NY, USA: ACM. ISBN 978-1-59593-609-7; 2007, p. 687–696. doi:10.1145/1281192.1281266.
- [51] Landesberger, T, Brodtkorb, F, Roskosch, P, Andrienko, N, Andrienko, G, Kerren, A. Mobilitygraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. IEEE Transactions on Visualization and Computer Graphics 2016;22:11–20. doi:10.1109/TVCG.2015.2468111.
- [52] Bach, B, Henry-Riche, N, Dwyer, T, Madhyastha, T, Fekete, JD, Grabowski, T. Small multipiles: Piling time to explore temporal patterns in dynamic networks. Computer Graphics Forum 2015;34(3):31–40. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgfm.12615>. doi:<https://doi.org/10.1111/cgfm.12615>.
- [53] Shetty, J, Adibi, J. The enron email dataset database schema and brief statistical report. Information sciences institute technical report, University of Southern California 2004;4(1):120–128.
- [54] Gama, J. Knowledge Discovery from Data Streams. 1st ed.; Chapman & Hall/CRC; 2010. ISBN 1439826110, 9781439826119.
- [55] Gama, J, Sebastião, R, Rodrigues, PP. On evaluating stream learning algorithms. Machine Learning 2013;90(3):317–346. URL: <https://doi.org/10.1007/s10994-012-5320-9>. doi:10.1007/s10994-012-5320-9.
- [56] Linhares, CDG, Ponciano, JR, Paiva, JGS, Rocha, LEC, Travençolo, BAN. Dynetvis - an interactive software to visualize structure and epidemics on temporal networks. In: 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). 2020, p. 933–936. doi:10.1109/ASONAM49781.2020.9381304.
- [57] Gemmetto, V, Barrat, A, Cattuto, C. Mitigation of infectious disease at school: targeted class closure vs school closure. BMC infectious diseases 2014;14(1):695. doi:10.1186/PREACCEPT-6851518521414365.
- [58] Stehlé, J, Voirin, N, Barrat, A, Cattuto, C, Isella, L, Pinton, J, et al. High-resolution measurements of face-to-face contact patterns in a primary school. PLOS ONE 2011;6(8):e23176. doi:10.1371/journal.pone.0023176.
- [59] Purchase, HC, Samra, A. Extremes are better: Investigating mental map preservation in dynamic graphs. In: International Conference on Theory and Application of Diagrams. Springer; 2008, p. 60–73.
- [60] Keila, PS, Skillicorn, DB. Structure in the enron email dataset. Comput Math Organ Theory 2005;11(3):183–199. doi:10.1007/s10588-005-5379-y.