

Research Article

Detecting Dynamic States of Temporal Networks Using Connection Series Tensors

Shun Cao ^{1,2} and Hiroki Sayama ^{1,2,3}

¹Center for Collective Dynamics of Complex Systems, Binghamton University, State University of New York, Binghamton, NY 13902-6000, USA

²Department of Systems Science and Industrial Engineering, Binghamton University, State University of New York, Binghamton, NY 13902-6000, USA

³Waseda Innovation Lab, Waseda University, Shinjuku, Tokyo 169-8050, Japan

Correspondence should be addressed to Shun Cao; scao12@binghamton.edu

Received 19 July 2020; Revised 17 November 2020; Accepted 30 November 2020; Published 23 December 2020

Academic Editor: Pietro De Lellis

Copyright © 2020 Shun Cao and Hiroki Sayama. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many temporal networks exhibit multiple system states, such as weekday and weekend patterns in social contact networks. The detection of such distinct states in temporal network data has recently been studied as it helps reveal underlying dynamical processes. A commonly used method is network aggregation over a time window, which aggregates a subsequence of multiple network snapshots into one static network. This method, however, necessarily discards temporal dynamics within the time window. Here we propose a new method for detecting dynamic states in temporal networks using connection series (i.e., time series of connection status) between nodes. Our method consists of the construction of connection series tensors over non-overlapping time windows, similarity measurement between these tensors, and community detection in the similarity network of those time windows. Experiments with empirical temporal network data demonstrated that our method outperformed the conventional approach using simple network aggregation in revealing interpretable system states. In addition, our method allows users to analyze hierarchical temporal structures and to uncover dynamic states at different spatial/temporal resolutions.

1. Introduction

Temporal networks are a useful framework to represent and analyze time-dependent changes and underlying dynamics of complex systems [1–3]. Many phenomena, ranging from disease spread [4–6] and human communication [7–9] to financial transactions [10, 11] and human brains [12, 13], can generate large-scale temporal network data. In many cases, the temporal network data can often be broken down into a sequence of discrete system states, some of which may reoccur many times. For example, air traffic networks can show seasonal variations [14, 15] and peak/off-peak weekly patterns [15], which can be modeled and studied as a temporal sequence of distinct system states. System state detection captures the temporal state change of the whole system at a collective level, in contrast to more commonly

studied node-level clustering on dynamic networks [16–18]. System state detection is useful for investigating the dynamics of time-varying complex systems and making better interpretation of large-scale temporal network data sets.

To detect the system states in temporal networks, Masuda and Holme recently proposed an approach that used network aggregation and graph similarity [19]. Their method first partitioned a given temporal network into subsequences and aggregated each subsequence into a static network. Then a graph similarity was measured among the aggregated static networks to generate a distance matrix, to which hierarchical clustering was applied and the number of system states was determined using Dunn's index [20]. In their method, the timelines of interactions between nodes within a time window were aggregated as static edge weights. Yet, these timelines of interactions may incorporate critical

information in exploring the system dynamics of temporal networks. For example, the patterns regarding fluctuation of connections in brain networks abstracting interactions between distinct brain regions can indicate various brain activities or states [21–23]. Additionally, the method in [19] focused on the optimal division of system states based on mathematical optimization, which may also hinder the discovery of suboptimal yet informative system states in temporal networks.

In this study, we propose a method to detect dynamic states of temporal networks using connection series between nodes, i.e., the sequence of connection status between two nodes represented as a binary-valued vector (0: disconnected, 1: connected). Figure 1(a) gives an example of the connection series between two nodes. Figure 1(b) provides a comparison of connection series and network aggregation between two illustrative cases. In Figure 1(b), though the timeline of interactions is different between windows 1 and 2, the aggregated networks (i.e., aggregation 1 and aggregation 2) are the same since the number of interactions between nodes is identical in each time window. Meanwhile, the connection series incorporates information regarding both amounts and temporal fluctuation of interactions between a pair of nodes, which may be more useful when detecting dynamic states of temporal networks.

Figure 2 shows two real-world examples of connection series of face-to-face contacts between two students in a primary school [24] and contacts between two attendees at an academic conference [25]. We can observe distinct fluctuation patterns between each pair of individuals over time. Both of the data sets were downloaded from SocioPattern.org.

Like in [19], our method divides a given temporal network data into subsequences using nonoverlapping time windows. Our method then transforms each subsequence into a connection series tensor. While tensors have been widely used in machine learning and pattern recognition research [26–28], we use tensors specifically as an extended representation of adjacency matrices that involve temporal connection patterns. Namely, every element in the adjacency matrix is replaced by a connection series between the corresponding pair of nodes. These connection series tensors generated from multiple subsequences are then connected to each other into a metalevel network whose edge weights are similarities between these tensors. Nonoverlapping communities are detected on this metalevel network to classify each connection series tensor into one of distinct dynamic states (represented as communities in the metalevel network). Experiments using two empirical temporal network data sets demonstrated that our method was capable of detecting interpretable and practical dynamic states in temporal networks. Additionally, by comparing the detected states with the already known sequence of events that took place in each data set, our method also outperformed the previous approach.

The rest of this paper is organized as follows. Section 2 describes our proposed method. Section 3 describes empirical temporal network data sets used in experiments. Section 4 presents the results. In section 5, we conclude and discuss limitations.

2. Method

A schematic overview of our method is presented in Figure 3. Given a temporal network with n network snapshots $S_O = [G_{t_1}(V_{t_1}, E_{t_1}), \dots, G_{t_n}(V_{t_n}, E_{t_n})]$, where G_{t_i} is the network snapshot at time point t_i , in which V_{t_i} and E_{t_i} denote the set of nodes and the set of edges, respectively. In this representation, $t_{i+1} = t_i + \Delta t$, where Δt is the sampling interval of the original temporal network data set. First, we split the whole temporal network data into T subsequences using nonoverlapping time windows of length $w \ll n$. The length of each subsequence is w , except for the last one that can be shorter than w if n is not divisible by w . We denote these subsequences as $S_S = [s^1, \dots, s^T]$. Second, we transform each subsequence into a connection series tensor by setting each element in an adjacency matrix to a connection series between the corresponding pair of nodes. We denote the obtained connection series tensors as $A = [A^{(1)}, \dots, A^{(T)}]$, where $A^{(i)}$ is the i th connection series tensor. Each connection series tensor $A^{(i)}$ is given by

$$A^{(i)} = \begin{bmatrix} c_{1,1}^{(i)} & c_{1,2}^{(i)} & \cdots & c_{1,V^{(i)}}^{(i)} \\ c_{2,1}^{(i)} & c_{2,2}^{(i)} & \cdots & c_{2,V^{(i)}}^{(i)} \\ \vdots & \vdots & \vdots & \vdots \\ c_{V^{(i)},1}^{(i)} & c_{V^{(i)},2}^{(i)} & \cdots & c_{V^{(i)},V^{(i)}}^{(i)} \end{bmatrix}, \quad (1)$$

where $c_{j,k}^{(i)}$ denotes the connection series between node j and node k during subsequence s^i and $V^{(i)}$ is the set of nodes that appears in subsequence s^i . Third, we quantify the similarity between every pair of connection series tensors by a measure of similarity that will be described later. Fourth, we construct a fully connected, weighted metalevel network whose nodes and edges represent these tensors and the similarities between them, respectively. Fifth, we run the Louvain method [29] on the metalevel network to classify these connection series tensors (nodes in this metalevel network) into multiple communities that are interpreted as dynamic states. We can also adjust the community resolution, a tunable parameter of the Louvain method, to study dynamic states at different spatial/temporal resolutions in a given temporal network.

2.1. Similarity between Connection Series Tensors. Let the two connection series tensors we compare be $A^{(1)}$ and $A^{(2)}$, whose node sets may be different: say $V^{(1)}$ and $V^{(2)}$. To make the format of $A^{(1)}$ and $A^{(2)}$ consistent, we transform them into $A'^{(1)}$ and $A'^{(2)}$, respectively, both of whose node sets are redefined as $V'^{(1)} = V'^{(2)} = V^{(1)} \cup V^{(2)}$. The steps of our proposed similarity measure are as follows.

Step 1. We compute the similarity between every pairwise connection series in $A'^{(1)}$ and $A'^{(2)}$, $\text{sim}(c_{j,k}^{(1)}, c_{j,k}^{(2)})$.

Step 2. We average all the similarities obtained in Step 1 as the similarity between $A^{(1)}$ and $A^{(2)}$,

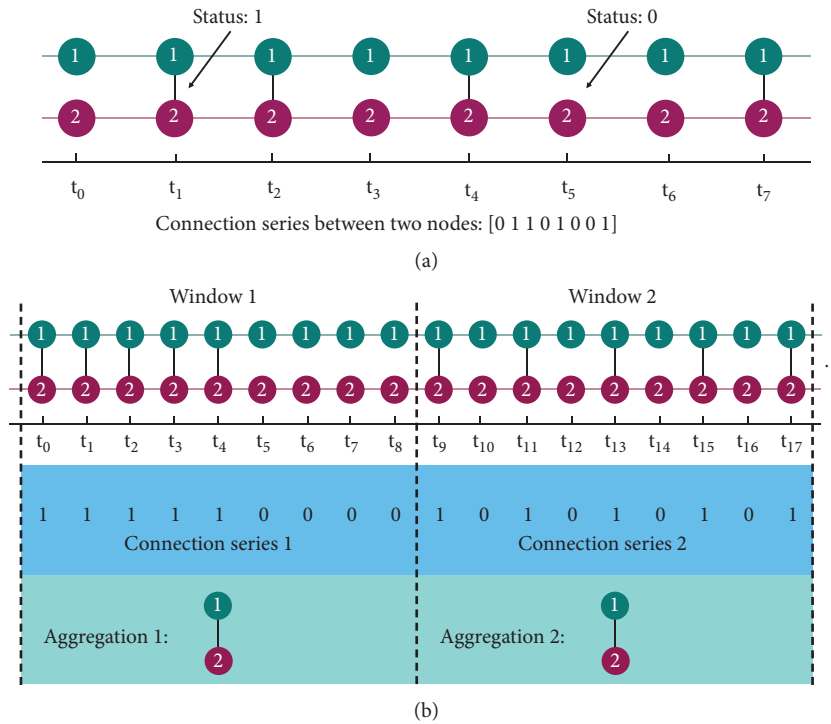


FIGURE 1: Connection series and network aggregation. (a) A simple example of connection series between nodes 1 and 2. (b) Comparison of connection series and network aggregation between two illustrative cases.

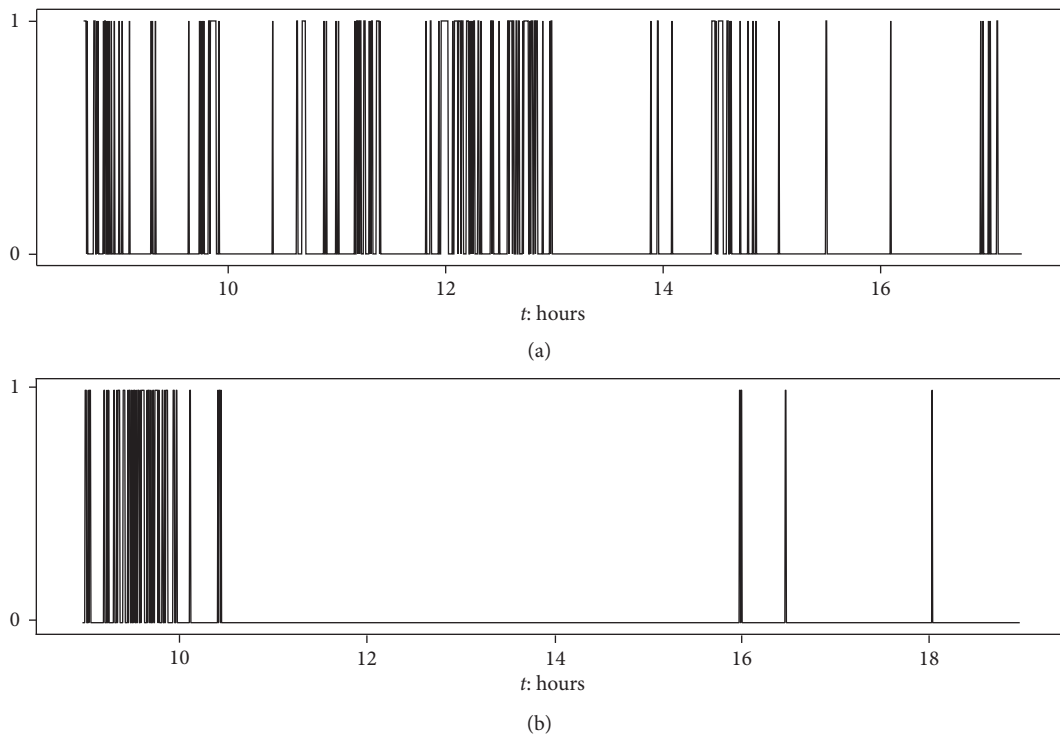


FIGURE 2: Real-world face-to-face contact sequence (i.e., connection series) involving two individuals. Connection status ‘1’ and status ‘0’ (values in y -axis) represent the fact that two nodes are connected and disconnected, respectively, at corresponding time points. (a) Contact sequence between “node 1558” and “node 1567” in the first day’s primary school data [24]. (b) Contact sequence between “node 1080” and “node 1125” in the first day’s conference data [25]. Note that the sampling interval (i.e., time resolution) is 20 seconds in both 2(a) and 2(b).

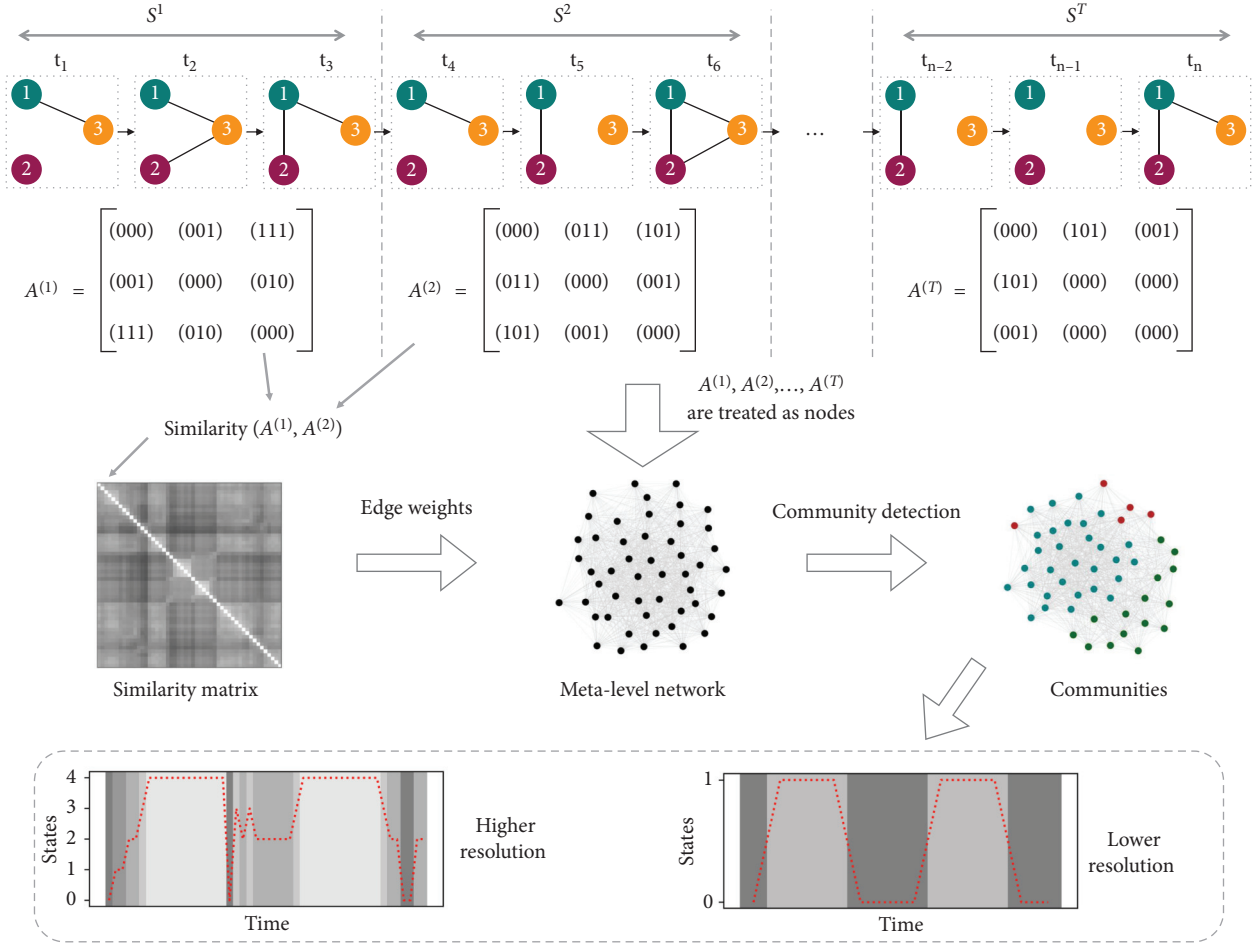


FIGURE 3: A schematic overview of our proposed method. Here we present a temporal network made of three nodes and set $w = 3$.

$$\begin{aligned} \text{Sim}(A^{(1)}, A^{(2)}) &= \text{Sim}(A'^{(1)}, A'^{(2)}) = \frac{1}{m(m-1)} \\ &\times \sum_j^m \sum_{k \neq j}^m \text{sim}(c_{j,k}^{(1)}, c_{j,k}^{(2)}), \end{aligned} \quad (2)$$

where m denotes the number of nodes in $V^{(1)}$ and $V^{(2)}$. Note that $c_{j,k}^{(i)}$ does not contain self-connection series $c_{j,j}^{(i)}$.

To compute the similarity between two connection series, $\text{sim}(c_{j,k}^{(1)}, c_{j,k}^{(2)})$, we developed a simple method informed by the well-developed similarity measures of time series [30–32]. Our similarity measure is based on the principle of maximizing the number of matched items. A schematic illustration of our proposed method is shown in Figure 4. The formula of similarity between two connection series is shown as follows:

$$\text{sim}(c_{j,k}^{(1)}, c_{j,k}^{(2)}) = \left(\frac{\max_{1 \leq i \leq w} M_i}{w} \right), \quad (3)$$

where M_i represents the number of matched elements in case i , while w is the length of time window. Note that the maximum length of the connection series equals the length of time window w .

2.2. Community Detection on Metalevel Network. Figure 5 gives a simple example of the metalevel network. We apply community detection to the metalevel network to assign each node (= a connection series tensor, or a subsequence of the original temporal network) with a distinct dynamic state label. Many community detection algorithms have been developed and employed with varying levels of success [29, 33–35]. Here we use the Louvain method [29], one of the most popular modularity maximization algorithms. In the case of a metalevel network, the modularity is defined as

$$Q = \left(\frac{1}{2n} \right) \sum_{ij} \left[\text{Sim}(A^{(i)}, A^{(j)}) - \left(\frac{k_i k_j}{2n} \right) \right] \delta(\text{cm}_i, \text{cm}_j), \quad (4)$$

where $\text{Sim}(A^{(i)}, A^{(j)})$ is the edge weight between tensors $A^{(i)}$ and $A^{(j)}$, $k_i = \sum_j \text{Sim}(A^{(i)}, A^{(j)})$ is the sum of the weights of the edges connected to node i , cm_i is the community label node i is assigned to, $\delta(\text{cm}_i, \text{cm}_j)$ is Kronecker's delta function, and n is the sum of all of the edge weights in the network. The Louvain method is beneficial to our work mainly for two reasons. First, it can take edge weights into account. Two nodes connected by an edge with greater edge weight (i.e., higher similarity between

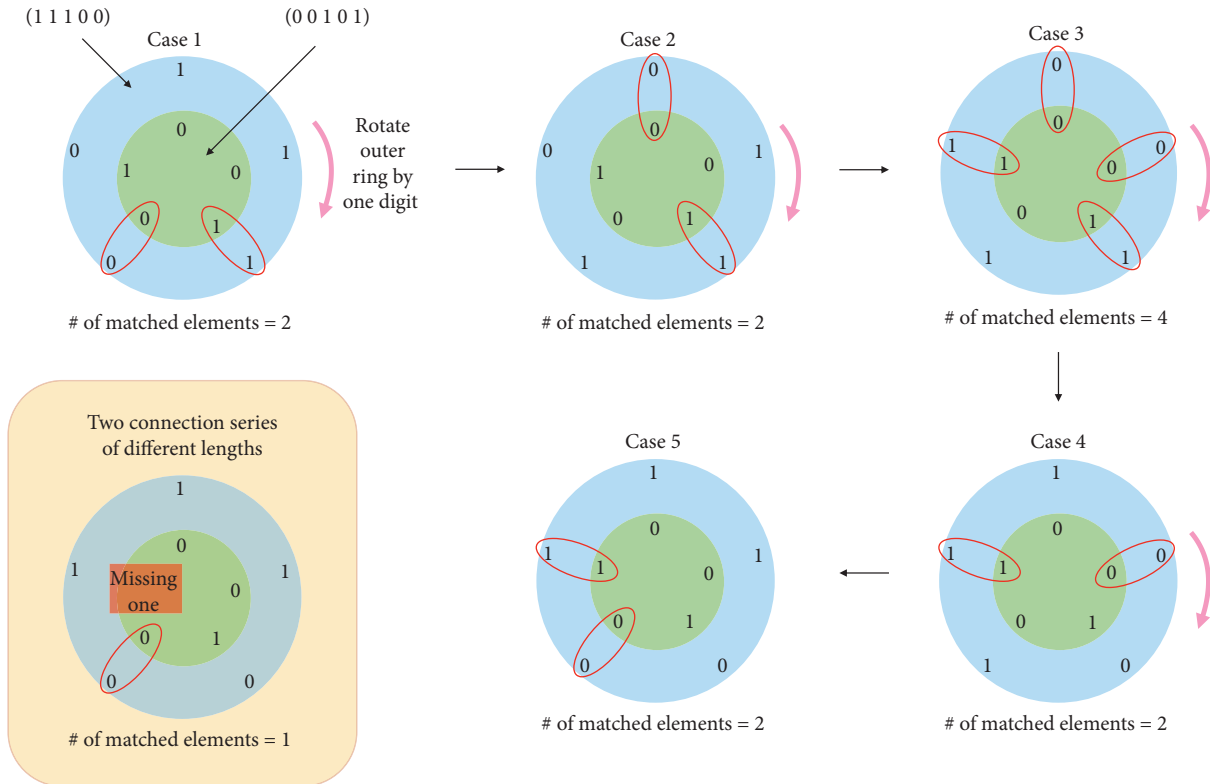


FIGURE 4: A schematic illustration of our proposed similarity measure for connection series. Given two connection series, say (1, 1, 1, 0, 0) and (0, 0, 1, 0, 1), we keep the sequential order of each series unchanged and find the best matching between them. To do this, we roll them into rings and fix one of them, say (0, 0, 1, 0, 1), as the inner ring, and the other one, say (1, 1, 1, 0, 0), as the outer ring. Then we rotate the outer ring clockwise element by element until we make a full rotation and obtain multiple cases regarding one-to-one correspondence between elements in the respective rings. In this example, we can obtain 5 matching cases. Note that the number of matching cases is equal to the length of connection series (i.e., length of time window). Then we count the number of matched elements in each case and choose the one that has the most matched elements as the best matching between these two given connection series. Here, case 3 provides the best matching between (1, 1, 1, 0, 0) and (0, 0, 1, 0, 1), in which the number of matched elements is 4. Finally, we divide the number of matched elements in the best matching case (case 3) by the length connection series w (length of the window), where $w = 5$ in the example shown here. The final similarity between (1, 1, 1, 0, 0) and (0, 0, 1, 0, 1) is 0.8. The bottom left panel in a light-yellow frame gives a special case when two connection series are different in length, say (1, 1, 1, 0, 0) and (0, 0, 1, 0). In this case, we fix the shorter one as the inner ring and the longer one as the outer ring, respectively. Then we still keep the one-to-one correspondence of elements in the respective rings and leave the missing digits in the inner ring empty (the red square in the bottom left panel). The rest of the process of finding the best matching is the same as previously. For the final step, we divide the max number of matched elements in the best matching case by the length of the longer connection series and then obtain the similarity between the two given connection series of different lengths.

connection series tensors) in the metalevel network are more likely to be assigned to the same dynamic state. Second, it also provides a tunable parameter of community resolution that allows for exploration of dynamic states at different spatial/temporal resolutions of interest, which is especially helpful for unknown temporal networks.

3. Data

We used the primary school and conference data sets downloaded from SocioPattern.org to run experiments. We chose them because there were known “ground truth” states to evaluate the performance of our method. Both the data sets represent the physical proximity between people. The basic properties of the two data sets are listed in Table 1.

3.1. Primary School Data. The primary school data was collected in a primary school in Lyon, France. In the school, each of all the five grades was divided into two classes [24]. The schedule of a school day was shown in Table 2. Note that different classes took turns to take breaks in a playground and to have lunch in a canteen because the playground or the canteen could not accommodate all the students at the same time [24]. The face-to-face contacts between 232 children and 10 teachers in the school were measured and recorded by body-mounted RFID devices. Two individuals were joined when they faced each other in a close range (about 1 m to 1.5 m). The data were collected from 8:45 to 17:20 on Thursday, October 1st, 2009, and from 8:30 to 17:05 on Friday, October 2nd, 2009. We used only the first day’s data in this paper.

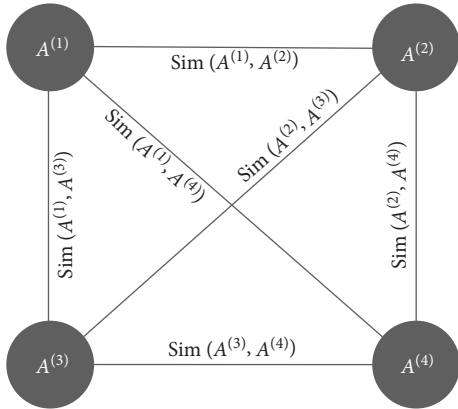


FIGURE 5: A schematic example of the generated metalevel network of connection series tensors. Here we consider only four tensors (i.e., $A^{(1)}$, $A^{(2)}$, $A^{(3)}$, $A^{(4)}$) as an example. The edge weight between node i (i.e., $A^{(i)}$) and node j (i.e., $A^{(j)}$) is defined by the similarity between $A^{(i)}$ and $A^{(j)}$ $\text{Sim}(A^{(i)}, A^{(j)})$.

TABLE 1: The properties of the temporal network data sets used in this paper.

Data	Primary school data [24]	Conference data [25]
Number of nodes	242	110
Selected period	First day, 8:40~17:20	First day, 9:00~19:00
Sampling interval	20 seconds	20 seconds

TABLE 2: Schedule of a school day in the primary school in Lyon, France [24].

Time	Event
8:30~10:30	Class time
10:30~10:55 (approximate time)	Break time
10:55~12:00	Class time
12:00~14:00	Lunchtime
14:00~15:30	Class time
15:30~15:55 (approximate time)	Break time
15:55~16:30	Class time

3.2. *Conference Data.* This data set is named ‘‘Hypertext 2009 dynamic contact network’’ on the website of SocioPattern.org, which we call ‘‘conference data’’ for short. The data set represents the temporal network of face-to-face contacts of about 110 attendees at an academic conference. It was collected during the ACM Hypertext 2009 conference (<http://www.ht2009.org/>) hosted by the Institute for Scientific Interchange Foundation in Turin, Italy, from June 29th to July 1st, 2009 [25]. The data collection method was the same as that used for the primary school data. We used only the first day’s (Monday, June 29, 2009) data in this paper, whose program was given in Table 3.

TABLE 3: The first day’s program of ACM hypertext 2009 conference (<http://www.ht2009.org/>).

Time	Event
9:00~10:30	Setup time for posters and demos
10:30~11:45	Workshops 1
11:45~12:00	Coffee break 1
12:00~13:30	Workshops 2
13:30~15:00	Lunch break
15:00~16:30	Workshops 3
16:30~16:45	Coffee break 2
16:45~18:05	Workshops 4
18:05~18:10	Short break
18:10~19:00	Wine and cheese welcome reception

4. Experiments

We applied our proposed method to the two real-world temporal networks to demonstrate how this approach can be used to detect meaningful insights regarding complex interactions among elements in time-varying complex systems. We used the event information shown in Tables 2 and 3 as the ground truth for our results. In the experiments, we varied the community resolution parameter in the Louvain method from 1.0 to smaller values (decreasing 0.01 in each variation) to scan the hierarchical temporal structure and uncover dynamic states at different resolutions. Note that a smaller community resolution parameter in the Louvain method indicates a higher resolution of system states. For comparison, we also implemented the approach using network aggregation and graph similarity proposed in [19]. Here we choose DeltaCon [36] from multiple graph similarity measures used in [19], because it takes the node identity into account that is compatible with our proposed method, and also because it is a relatively new, computationally scalable method.

4.1. *Results for Primary School Data.* We partitioned the primary school data into subsequences through time windows of length of 20 minutes, which was the same as that used for the same data set in [19]. Figure 6 presents the results for the primary school data obtained by our method. Figure 6(b) exhibits the detected two dynamic states, state 0 (the approximated periods are 8:40~11:50 and 14:10~17:20) and state 1 (the approximated period is 11:50~14:10) at the community resolution of 1.0. By comparing the results with the primary school’s schedule, state 1 corresponds to the lunchtime, while state 0 can be simply assumed as class time. However, the morning and afternoon breaks were not revealed with such a large community resolution parameter value (low resolution of system states).

Figure 7 provides more representative results that obtained at distinct community resolutions. The results at community resolution of 0.92 are shown in Figure 7(b). There are three detected dynamic states, state 0, state 1, and state 2, which are consistent with class time, break time (morning and afternoon), and lunchtime, respectively. The detected two breaks are longer than the break time in the schedule of a school day, which may be led by the fact that

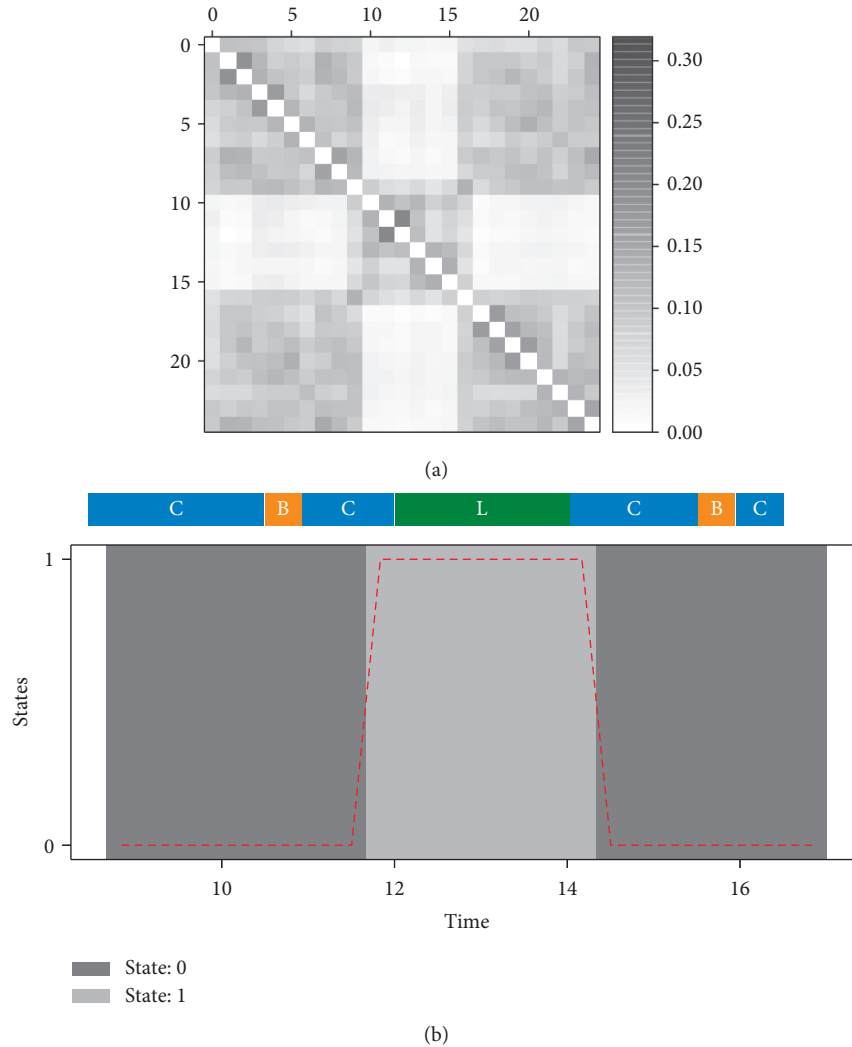


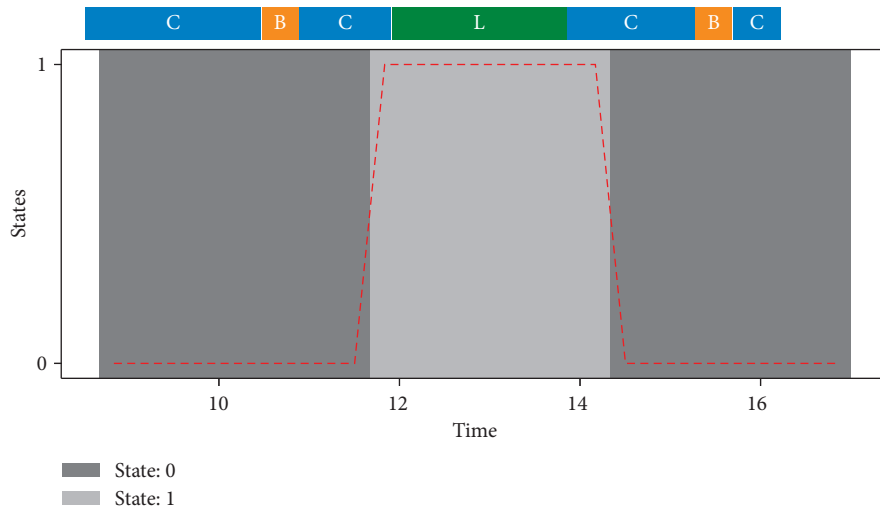
FIGURE 6: Results for the primary school data using our proposed method. (a) Similarity matrix. (b) Detected states obtained by setting community resolution as 1.0. We use the red dotted line and spans with distinct greyscales to represent the dynamic states. The schedule of a school day is displayed in a horizontal bar on the top of the figure. The letters ‘C’, ‘B’, and ‘L’ in the bar correspond to the class time, the break time, and the lunchtime in the schedule, respectively. Note that the time span (8:45~17:20) of the collected data is not aligned with the schedule of a school day (8:30~16:30).

students in different classes took turns to take breaks due to the limitation of the playground [24]. The results in Figure 7(c) that were obtained by using an even smaller community resolution (0.91) show that the morning break can be broken into two dynamic states, state 1 and state 2, which may suggest a distinction in collective behaviors of different classes. If we continue to decrease the community resolution to 0.89 (Figure 7(d)), we can even find the subtle differences between morning class time (state 0 in Figure 7(d)) and afternoon class time (state 4 in Figure 7(d)). Additionally, the results in Figure 7 indicate that smaller community resolution parameter values (higher resolution of system states) are beneficial to discover more subtle dynamic states.

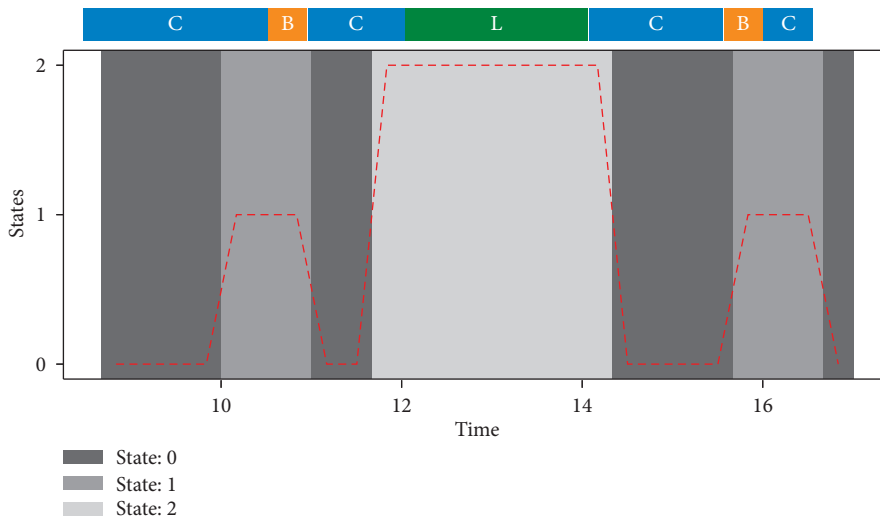
As a comparison, Figure 8 gives the results for the primary school data obtained by the method using network aggregation and graph similarity in [19]. This method discovered two optimal system states shown in Figure 8(b)

through hierarchical clustering and Dunn’s index [20]. Though it detected the lunchtime and class time, it failed to recognize the two breaks between classes.

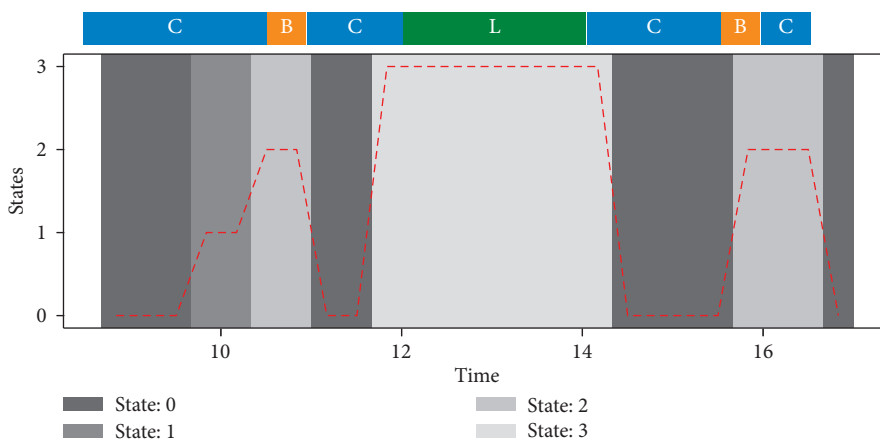
4.2. Results for Conference Data. The conference temporal network was divided into subsequences by nonoverlapping time windows of length of 5 minutes, which were chosen according to the shortest event (short break (18:05~18:10) in Table 3). Figure 9 shows the results obtained by our method, which are mainly aligned with the first day’s program of ACM hypertext 2009 conference. For example, the results in Figure 9(b) suggest that coffee break 1, coffee break 2, wine and cheese welcome reception, and part of lunch break (approximately from 13:30~15:15) are recognized as the same dynamic state 2. Workshops 1, 2, 3, and 4 correspond to state 0, state 3, state 1, and state 4, respectively. Figure 9(c) provides more subtle insights at



(a)



(b)



(c)

FIGURE 7: Continued.

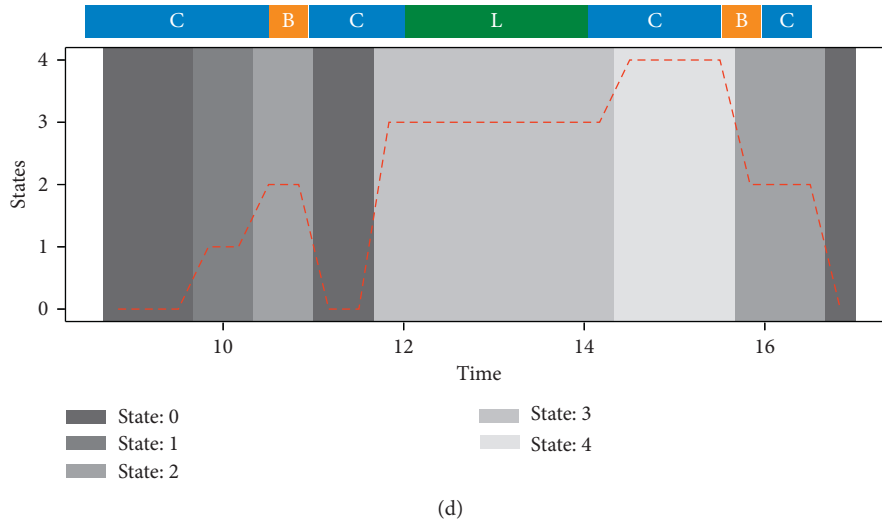


FIGURE 7: More representative results for the primary school data using our method. (a) Detected states obtained by setting community resolution as 1.0, which is the same as the results in Figure 6(b). (b) Detected states at community resolution of 0.92. (c) Detected states at community resolution of 0.91. (d) Detected states at community resolution of 0.89. Note that the time span (8:45~17:20) of the collected data is not aligned with the schedule of a school day (8:30~16:30).

relatively smaller community resolution (0.99), in which a new period (state 5) emerges at the end of the whole day’s program. This new period may be interpreted as the “ending time” of a conference day.

On the contrary, the results displayed in Figure 10 were obtained by using the method in [19]. As shown in Figure 10(b), the only two detected system states, state 0 and state 1, failed to match the first day’s conference program. Therefore, it performed poorly in revealing meaningful system states in the conference data.

5. Discussion

In this paper, we developed a new method for detecting dynamic states in temporal networks. We transformed a given temporal network into a sequence of tensors that consisted of connection series between each pair of nodes. These connection series can help capture the collective dynamics regarding temporal and spatial interactions between elements in time-varying complex systems. We also proposed a simple method to evaluate the similarity between two connection series tensors, which can be potentially extended to a similarity measure of two temporal networks. The results with empirical temporal network data demonstrated the effectiveness of our method in detecting dynamic states. Our method also outperformed the previous approach in [19] in revealing actual events in real-world temporal network data, which suggests that incorporating timelines of interactions between pairs of nodes within time windows helps detecting dynamic system states.

As demonstrated in Figure 7, the tunable parameter of community resolution in the Louvain method is a useful tool to detect the system states at various spatial/temporal resolutions. Users can choose the appropriate community resolution parameter according to their research interests. For temporal network data for which no ground truth or

underlying processes are known a priori, one possible way to find the “right” level of system state detection is to scan the parameter space gradually from high to low values and then choose the most robust, persistent division of system states as the final result. Users may also benefit from using several community validation metrics [37–39] for this purpose.

We note that the choice of the length of nonoverlapping time windows would influence the results of detected states in our method. In general, a shorter time window helps discover more subtle system states, whereas it may also make the results less robust since the temporal sparseness of interactions between nodes in many temporal networks [1] could lead to insufficient topological information in each time window. Using a large time window obviously can avoid this problem, but the results may be too coarse-grained because a large time window may include multiple system states. Figure 11 gives an example of how the time window length may influence the results of state detection, showing results of our method applied to the primary school data with varying lengths of time windows. The shorter the time windows were, the more subtle the system states were uncovered. We consider it is important to choose the length of the time windows according to the underlying dynamics of the time-varying complex systems (e.g., the duration of possible shortest events). If no such information is available a priori, one should systematically vary the length of time windows until a robust result of system states is identified.

Finally, we note several limitations of our work. First, the proposed similarity measure for the connection series tensors is limited in temporal networks with given node labels. Further methodological exploration and development would be needed to handle unlabeled temporal network data. Second, consideration of all the connection series in temporal networks may become computationally very expensive when the size of the temporal network data is very large. Third, we have used only the Louvain method and have not

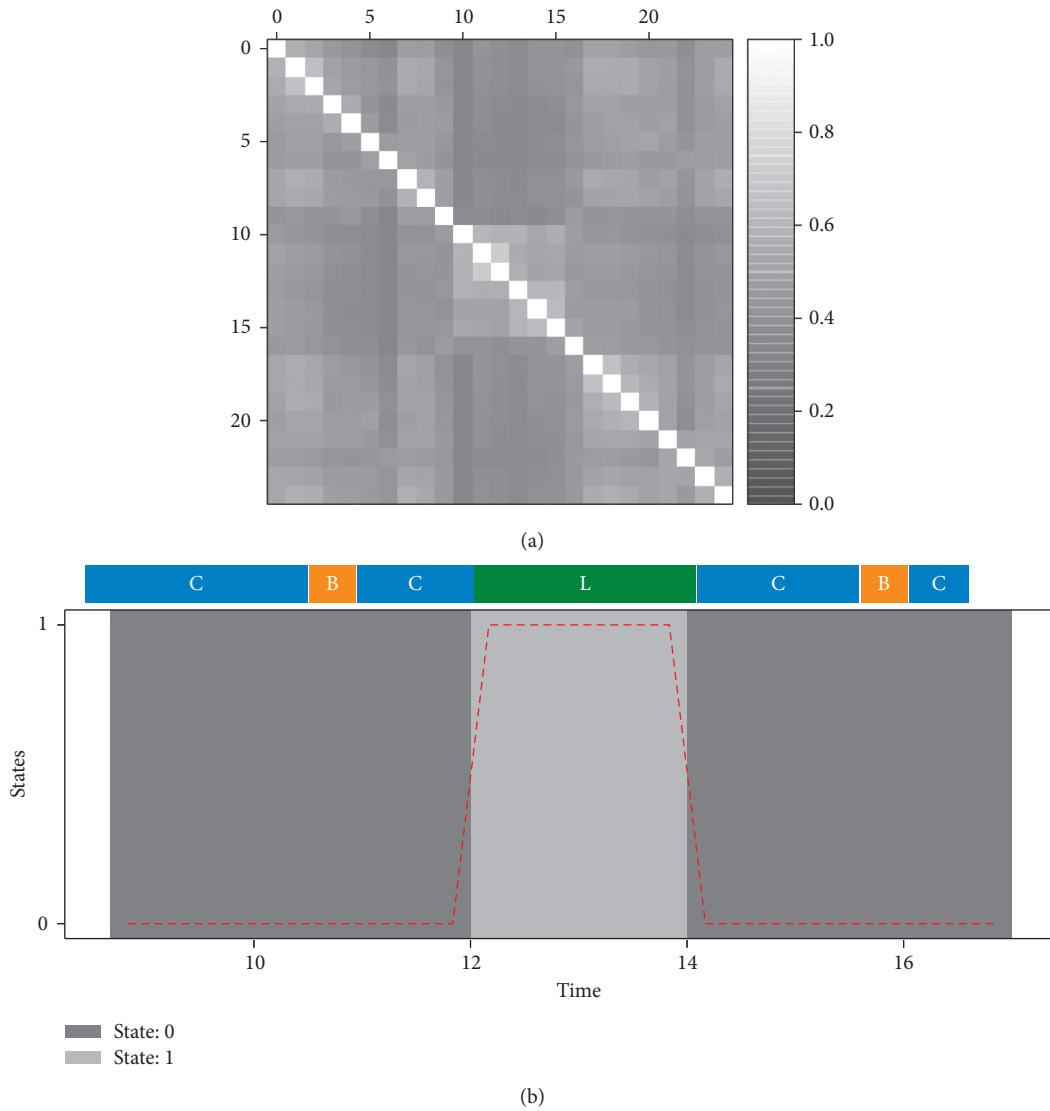


FIGURE 8: The results for the primary school data obtained by the method using network aggregation and graph similarity in [19]. (a) Similarity matrix. (b) Detected system states. Note that the time span (8 : 45~17 : 20) of the collected data is not aligned with the schedule of a school day (8 : 30~16 : 30).

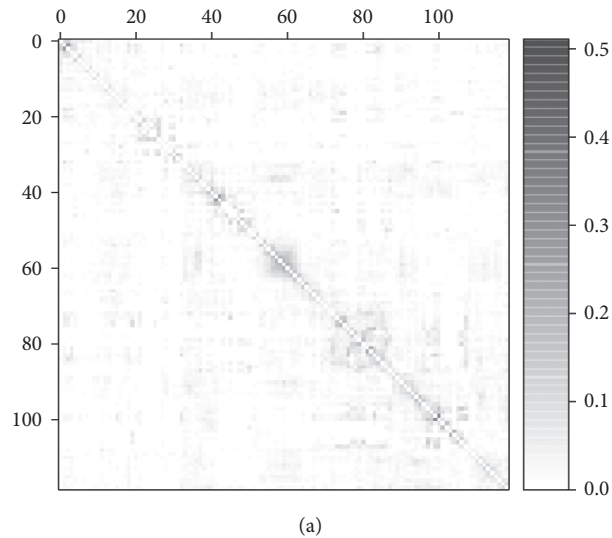


FIGURE 9: Continued.

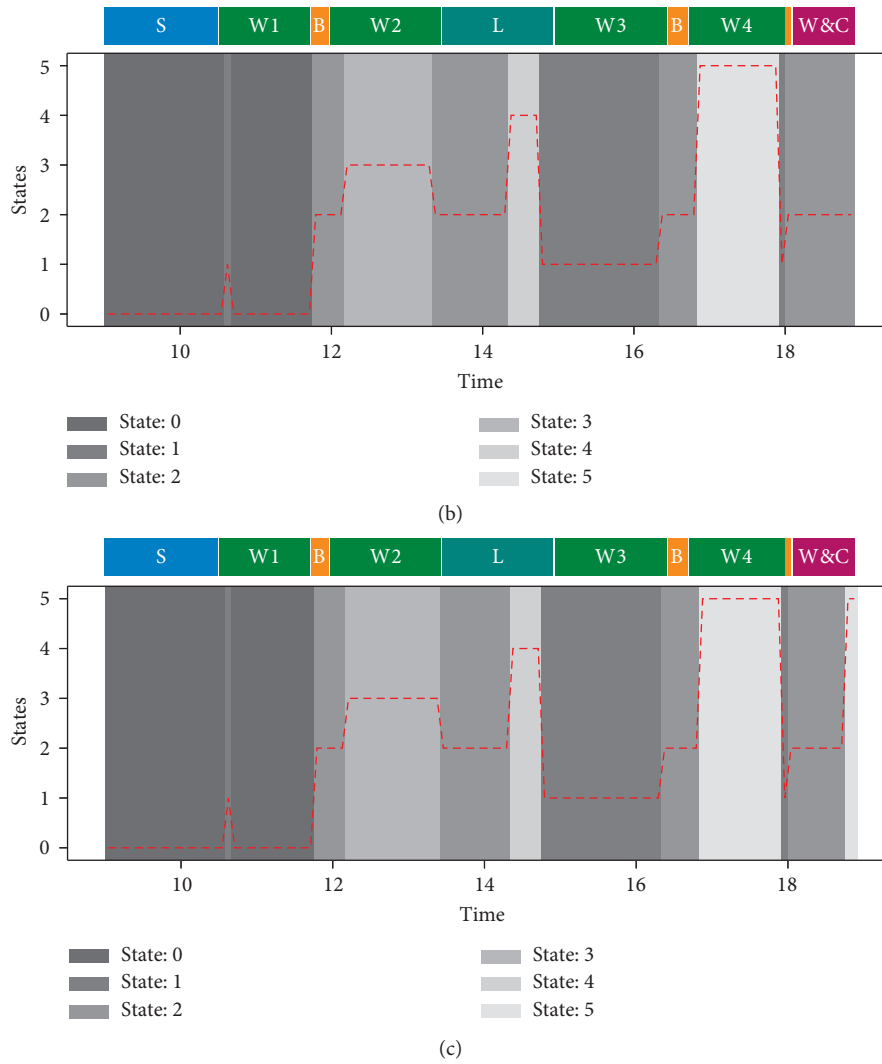


FIGURE 9: Results for the conference data using our proposed method. (a) Similarity matrix. (b) Detected states with setting community resolution as 1.0. (c) Detected states obtained by setting community resolution as 0.99. The horizontal bar on the top of the figures 9(b) and 9(c) shows the first day’s program of the ACM hypertext 2009 conference. The letters ‘S,’ ‘W1,’ ‘B,’ ‘W2,’ ‘L,’ ‘W3,’ ‘W4,’ and ‘W&C’ in the bar correspond to setup time, workshop 1, coffee break, workshop 2, lunchtime, workshop 3, workshop 4, and wine and cheese welcome reception, respectively.

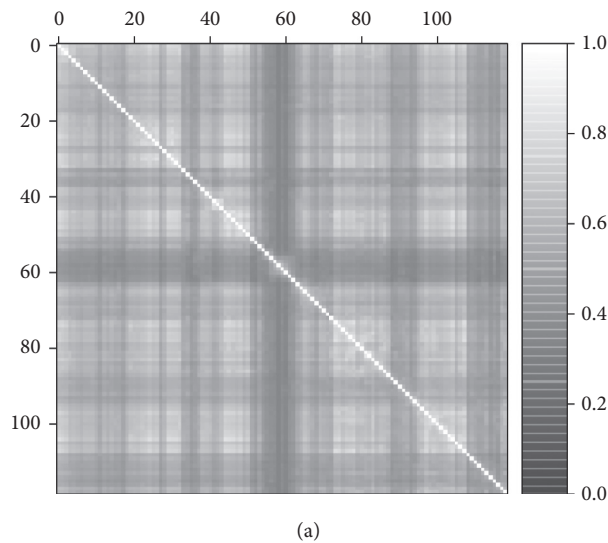
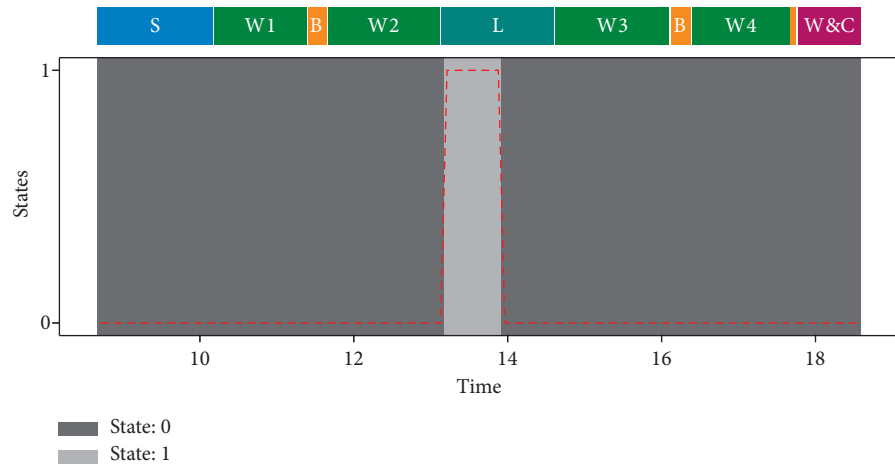
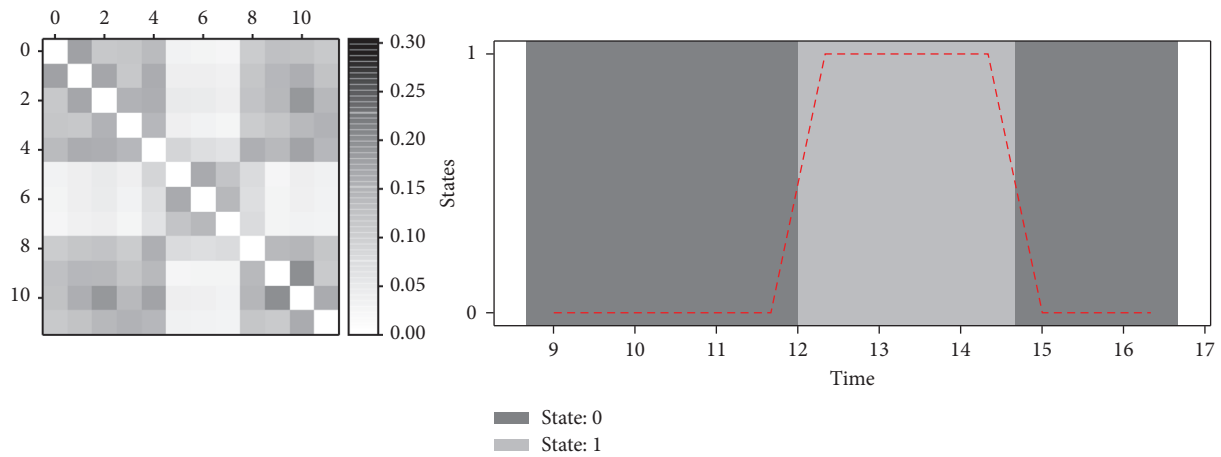


FIGURE 10: Continued.

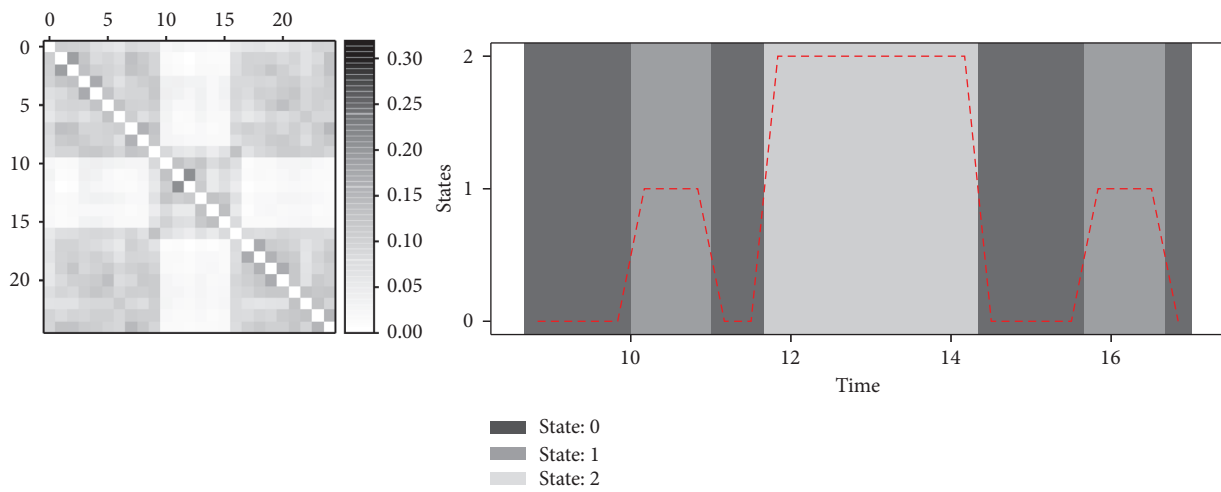


(b)

FIGURE 10: Results for the conference data using the method in [16]. (a) Similarity matrix. (b) Detected system states. The letters ‘S,’ ‘W1,’ ‘B,’ ‘W2,’ ‘L,’ ‘W3,’ ‘W4,’ and ‘W&C’ in the bar correspond to setup time, workshop 1, coffee break, workshop 2, lunchtime, workshop 3, workshop 4, and wine and cheese welcome reception, respectively.

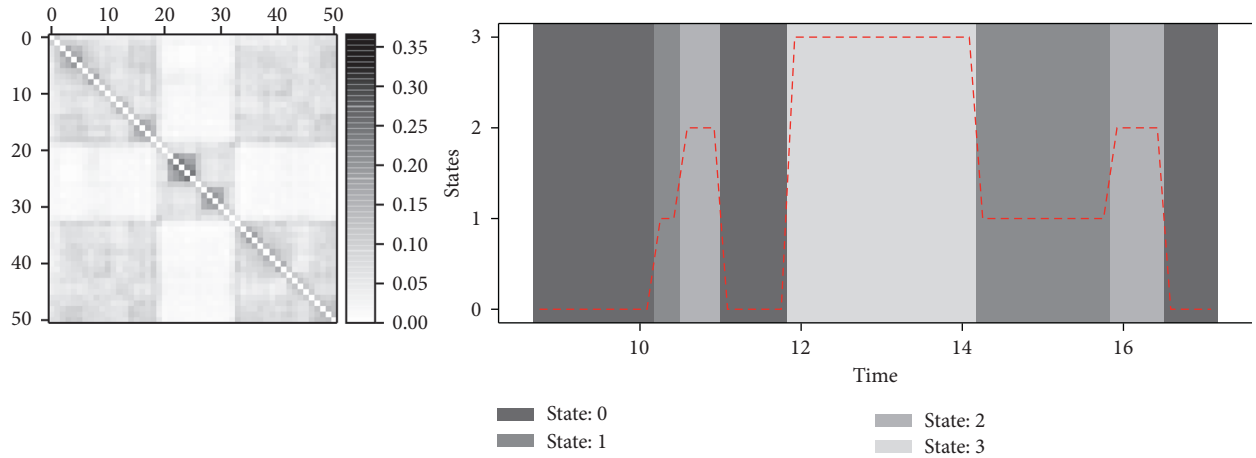


(a)

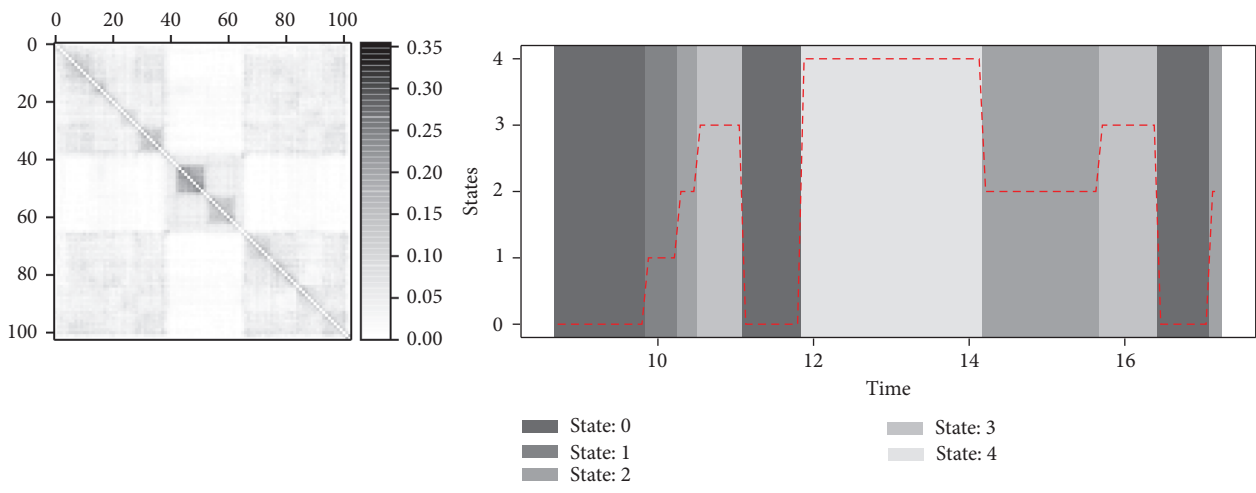


(b)

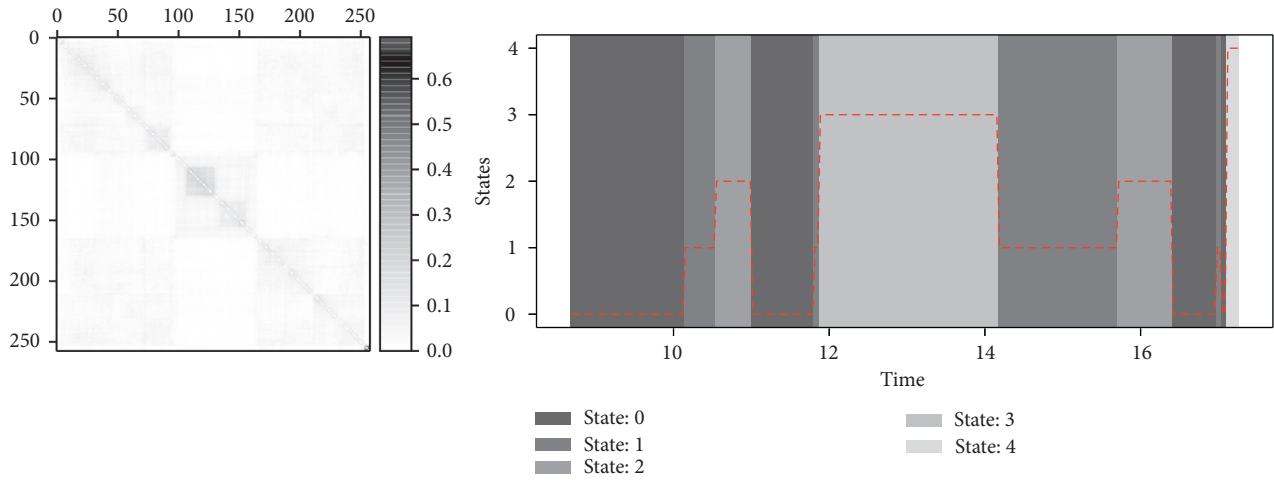
FIGURE 11: Continued.



(c)



(d)



(e)

FIGURE 11: Continued.

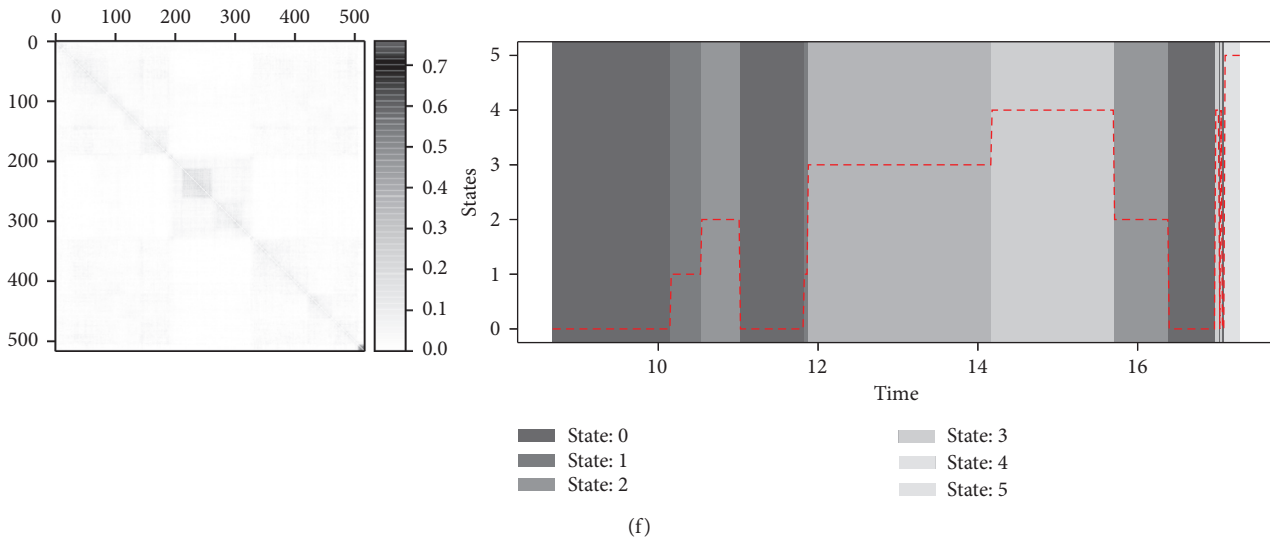


FIGURE 11: Results for the primary school data using distinct sizes of the time window (community resolution parameter = 0.92). (a) Results using windows of 40 minutes. (b) Results using windows of 20 minutes. (c) Results using windows of 10 minutes. (d) Results using windows of 5 minutes. (e) Results using windows of 2 minutes. (f) Results using windows of 1 minute.

explored different community detection methods (although we assume the main results would not be affected much as long as community detection was done to maximize modularity). Finally, our validation of the results remained only qualitative comparison with the presumed “ground truth”, while more objective, quantitative validation requires further study using other empirical temporal network data sets whose underlying system states (ground truth) are rigorously established and available.

Data Availability

All the data used in this study are publicly available on the website of <http://www.sociopatterns.org/>. The codes of our proposed method are available from <https://github.com/shun-cao/Detecting-Dynamic-States-of-Temporal-Networks-Using-Connection-Series-Tensors>.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors greatly thank Naoki Masuda for his valuable suggestions and comments on this work. The authors also thank SocioPatterns.org for sharing the temporal network data sets for public use. This work was supported by the National Science Foundation, under Grant no. 1734147.

References

- [1] P. Holme and J. Saramäki, *Temporal Network Theory*, Springer International Publishing, Berlin, Germany, 2019.
- [2] P. Holme, “Modern temporal network theory: a colloquium,” *The European Physical Journal B*, vol. 88, p. 234, 2015.
- [3] A. Li, S. P. Cornelius, Y.-Y. Liu, L. Wang, and A.-L. Barabási, “The fundamental advantages of temporal networks,” *Science*, vol. 358, no. 6366, pp. 1042–1046, 2017.
- [4] N. Masuda and P. Holme, *Temporal Network Epidemiology*, Springer, Berlin, Germany, 2017.
- [5] J. Stehlé, V. Nicolas, B. Alain, and C. Ciro, “Simulation of an SEIR infectious disease model on the dynamic contact network of conference attendees,” *BMC Medicine*, vol. 9, p. 87, 2011.
- [6] A. Perer and J. Sun, “Matrixflow: temporal network visual analytics to track symptom evolution during disease progression,” in *Proceedings of the AMIA Annual Symposium Proceedings*, San Francisco, CA, USA, November 2012.
- [7] M.-X. Li, P. Vasyl, J. Zhi-Qiang, K. Kaski et al., “Statistically validated mobile communication networks: the evolution of motifs in European and Chinese data,” *New Journal of Physics*, vol. 16, Article ID 083038, 2014.
- [8] Q. Ye, Z. Tian, H. Deyong, and W. Bin, “Cell phone mini challenge award: social network accuracy—exploring temporal communication in mobile call graphs,” in *Proceeding of the IEEE Symposium on Visual Analytics Science and Technology*, IEEE, Chicago, IL, USA, October 2008.
- [9] S. J. Stolfo, G. Creamer, and S. Hershkop, “A temporal based forensic analysis of electronic communication,” in *Proceedings of the 2006 International Conference on Digital Government Research*, San Diego, CA, USA, May 2006.
- [10] Y. Fujiwara, Y. Ikeda, H. Aoyama, and H. Iyetomi, “Structure and temporal change of the credit network between banks and large firms in Japan,” *Economics: The Open-Access Open-Assessment E-Journal*, vol. 3, p. 7, 2009.
- [11] C. W. J. Granger and Z. Ding, “Stylized facts on the temporal and distributional properties of daily data from speculative markets,” *UCSD Department of Economics Discussion Paper*, pp. 94–19, 1994.
- [12] W. Gao, J. H. Gilmore, K. S. Giovanello et al., “Temporal and spatial evolution of brain network topology during the first two years of life,” *PLoS One*, vol. 6, no. 9, Article ID e25278, 2011.
- [13] W. H. Thompson, P. Brantefors, and P. Fransson, “From static to temporal network theory: applications to functional brain

- connectivity,” *Network Neuroscience*, vol. 1, no. 2, pp. 69–99, 2017.
- [14] Z. Neal, “The devil is in the details: differences in air traffic networks by scale, species, and season,” *Social Networks*, vol. 38, pp. 63–73, 2014.
- [15] X. Sun, S. Wandelt, and F. Linke, “Temporal evolution analysis of the European air transportation system: air navigation route network and airport network,” *Transportmetrica B: Transport Dynamics*, vol. 3, no. 2, pp. 153–168, 2015.
- [16] A. Gorovits, E. Gujral, and E. Evangelos, “Papalexakis, and petko bogdanov. Larc: learning activity-regularized overlapping communities across time,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1465–1474, London, UK, August 2018.
- [17] N. Mishra, R. Schreiber, I. Stanton, E. Robert, and Tarjan, “Clustering social networks,” in *International Workshop on Algorithms and Models for the Web-Graph*, pp. 56–67, Warsaw, Poland, September 2007.
- [18] L. Zhang, A. Gorovits, and P. Bogdanov, “PERCeIDs: PE-Riodic Community detection,” in *Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM)*, pp. 816–825, IEEE, New Orleans, LA, USA, November 2019.
- [19] N. Masuda and P. Holme, “Detecting sequences of system states in temporal networks,” *Scientific Reports*, vol. 9, p. 795, 2019.
- [20] J. C. Dunn, “A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters,” *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.
- [21] V. D. Calhoun, R. Miller, G. Pearlson, and T. Adali, “The chronnectome: time-varying connectivity networks as the next frontier in fMRI data discovery,” *Neuron*, vol. 84, no. 2, pp. 262–274, 2014.
- [22] J. M. Shine, P. G. Bissett, P. T. Bell et al., “The dynamics of functional brain networks: integrated network states during cognitive task performance,” *Neuron*, vol. 92, no. 2, pp. 544–554, 2016.
- [23] Q. K. Telesford, M.-E. Lynall, J. Vettel, M. B. Miller, S. T. Grafton, and D. S. Bassett, “Detection of functional brain network reconfiguration during task-driven cognitive states,” *NeuroImage*, vol. 142, pp. 198–210, 2016.
- [24] J. Stehlé et al., “High-resolution measurements of face-to-face contact patterns in a primary school,” *PloS One*, vol. 6, 2011.
- [25] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. Van den Broeck, “What’s in a crowd? Analysis of face-to-face behavioral networks,” *Journal of Theoretical Biology*, vol. 271, no. 1, pp. 166–180, 2011.
- [26] X. Luo, H. Wu, H. Yuan, and M.C. Zhou, “Temporal pattern-aware QoS prediction via biased non-negative latent factorization of tensors,” *IEEE Transactions on Cybernetics*, vol. 50, no. 5, pp. 1798–1809, 2019.
- [27] X. Luo, J. Sun, Z. Wang, S. Li, and M. Shang, “Symmetric and nonnegative latent factor models for undirected, high-dimensional, and sparse networks in industrial applications,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3098–3107, 2017.
- [28] D. Hallac, S. Vare, S. Boyd, and J. Leskovec, “Toeplitz inverse covariance-based clustering of multivariate time series data,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 215–223, Halifax, NS, Canada, August 2017.
- [29] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.
- [30] D. Q. Goldin and P. C. Kanellakis, “On similarity queries for time-series data: constraint specification and implementation,” in *Proceedings of the International Conference on Principles and Practice of Constraint Programming*, Berlin, Heidelberg, 1995.
- [31] T.-c. Fu, “A review on time series data mining,” *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, 2011.
- [32] D. Rafiei and A. Mendelzon, “Similarity-based queries for time series data,” in *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, Portland, OR, USA, June 1997.
- [33] M. E. J. Newman, “Detecting community structure in networks,” *The European Physical Journal B - Condensed Matter*, vol. 38, no. 2, pp. 321–330, 2004.
- [34] A. J. Alvarez, C. E. Sanz-Rodríguez, and J. Luis Cabrera, “Weighting dissimilarities to detect communities in networks,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 373, p. 2056, 2015.
- [35] M. E. J. Newman, “Fast algorithm for detecting community structure in networks,” *Physical Review E*, vol. 69, Article ID 066133, 2004.
- [36] D. Koutra, “DeltaCon: principled massive-graph similarity function with attribution,” *ACM Transactions on Knowledge Discovery from Data (TKDD)* 10, vol. 3, p. 28, 2016.
- [37] K. Steinhaeuser and N. V. Chawla, “Identifying and evaluating community structure in complex networks,” *Pattern Recognition Letters*, vol. 31, no. 5, pp. 413–421, 2010.
- [38] S. Fortunato and D. Hric, “Community detection in networks: a user guide,” *Physics Reports*, vol. 659, pp. 1–44, 2016.
- [39] M. Gustafsson, M. Hörnquist, and A. Lombardi, “Comparison and validation of community structures in complex networks,” *Physica A: Statistical Mechanics and Its Applications*, vol. 367, pp. 559–576, 2006.