Nonparametric Online Monitoring of Dynamic Networks [1]

Yipeng Wang[1], Xiulin Xie[2] and Peihua Qiu[1*]

[1]Department of Biostatistics, University of Florida
2004 Mowry Road, Gainesville, FL 32610

[2]Department of Statistics, Florida State University
117 North Woodward Ave, Tallahassee, FL 32306

[*]Corresponding author, E-mail: pqiu@ufl.edu

Abstract

Network sequence has been commonly used for describing the longitudinal pattern of a dynamic system. Proper online monitoring of a network sequence is thus important for detecting temporal structural changes of the system. To this end, there have been some discussions in the statistical process control (SPC) literature to first extract some features from the observed networks and then apply an SPC chart to monitor the extracted features sequentially over time. However, the features used in many existing methods are insensitive to some important network structural changes, and the control charts used cannot accommodate the complex structure of the extracted features properly. In this paper, we suggest using four specific features to describe the structure of an observed network, and their combination can reflect most network structural changes that we are interested in detecting in various applications. After the four features are extracted from the observed networks, we suggest using a multivariate nonparametric control chart to monitor the extracted features online. Numerical studies show that our proposed network monitoring method is more reliable and effective than some representative existing methods in various cases considered.

Key Words: Dynamic networks; Features; Hurdle model; Multivariate nonparametric control charts; Network surveillance; Statistical process control.

1 Introduction

Many economic, biological, electrical, and social systems can be described by networks to show pairwise interactions among system entities (e.g., Harrison 1992, Leskovec *et al.* 2007, Lotker 2021).

[1] The data that support the findings of this study are available from the corresponding author upon reasonable request.

As an example, in the enterprise network data collected by the Los Alamos National Laboratory (Turcotte *et al.* 2019), communications among 27,436 enterprise computers running the Microsoft Windows operating system over a period of 89 days can be described by a network (Lee *et al.* 2022). When the network structure can change over time, the related network is called dynamic network in the literature (Newmann 2018). Dynamic networks have been routinely used in practice since they can describe the complex longitudinal structural change trajectories of dynamic systems in concern. One fundamental task to analyze dynamic network data is to monitor the sequence of observed networks at different time points and detect any network structural changes in a timely manner. This paper aims to develop a flexible and effective method for this purpose.

In the literature, there have been many discussions about online monitoring of dynamic networks. Most existing methods first extract certain features from the observed networks and then monitor the extracted features over time (e.g., Leskovec *et al.* 2007, Perry 2020, Flossdorf and Jentsch 2021). Some existing methods first fit dynamic network models and then monitor the model-based metrics about the observed networks (Zhang *et al.* 2017, Kim *et al.* 2018). The dynamic network models considered include the stochastic block models (Holland *et al.* 1983, Karrer and Newman 2011, Xu and Hero 2014), the latent space models (Hoff *et al.* 2002, Sewell and Chen 2015), the temporal exponential random graph model (Hanneke *et al.* 2010), and more. Wilson *et al.* (2019) used the Shewhart and the exponentially weighted moving average (EWMA) control charts to monitor the estimated parameters of stochastic block models for dynamic network monitoring. Dong *et al.* (2020) suggested a multilayer weighted stochastic block modeling scheme for online monitoring of multilayer dynamic networks. Ebrahimi *et al.* (2021) integrated hurdle models with state-space models to capture the temporal dynamics of the edge formation process of a network sequence, and then monitored the sequence by an EWMA chart based on the estimated state-space hurdle models. Lee *et al.* (2022) recently developed a method for online monitoring of directed activities in large-scale networks based on latent network space modeling with latent nodal attributes. For overviews on network modeling and/or monitoring, see papers such as Savage *et al.* (2014), Ranshous *et al.* (2015), Woodall *et al.* (2017), Jeske *et al.* (2018), Yu *et al.* (2022), and the references cited therein.

Although dynamic network modeling approaches can provide us an analytic tool to approximate or predict the dynamic patterns of network sequences, their model assumptions are often restrictive and hard to justify in practice. For instance, the assumption that the set of nodes does not change over time required by many existing methods may not be realistic in many dynamic network applications (e.g., Liu *et al.* 2021). Specifically, in the message network example that will be discussed in Subsection 4.1, all active members of an online community constitute the set of nodes of the related network. In this example, because new members can join the community and existing members can leave the community at any time, the node set could change over time. Some existing methods based on stochastic block models require the number of communities within a network does not change over time (e.g., Zhao *et al.* 2012, Nguyen *et al.* 2014, Rossetti and Cazabet 2018), which may not be realistic in many applications either. In addition, many dynamic network models (e.g., the hurdle models) have parametric forms, and a proper justification of the parametric forms is often challenging (Hunter *et al.* 2008).

After certain network features are extracted based on dynamic network modeling and/or task-specific intuition (e.g., using the total number of nodes of a network if we want to monitor the network size over time), traditional statistical process control (SPC) charts, such as the Shewhart, cumulative sum (CUSUM) and EWMA charts, are routinely used to monitor the extracted features sequentially over time (McCulloh and Carley 2011, Flossdorf and Jentsch 2021, Yu *et al.* 2022). However, these charts are usually designed for cases when the in-control (IC) process observations at different time points are assumed independent and identically distributed with a pre-specified parametric distribution form (cf., Qiu 2014). In the SPC literature, it has been well demonstrated that these assumptions are rarely valid in practice and the traditional SPC charts would be unreliable to use when one or more of their model assumptions are violated (Apley and Tsung 2002, Capizzi and Masarotto 2008, Jones-Farmer 2009, Qiu and Xiang 2014, Qiu 2018). To circumvent the "data independence" assumption, Ofori-Boateng *et al.* (2021) proposed a change-point detection method for monitoring a network sequence, based on time series modeling for a network summary statistic. Their proposed method derived the sequential decision rule by approximating the IC distribution of a test statistic at each observation time using the sieve bootstrap procedure. This method is thus hard to use for online monitoring of a network sequence because bootstrapping at each observation time until a signal is time-consuming. For monitoring dynamic networks with a fixed node set, Salmasnia *et al.* (2020) used the multivariate EWMA chart to monitor four summary statistics (or features) of individual networks in a network sequence. To use this approach, a considerable amount (e.g., 150) of networks needs to be collected at each observation time, so that the batch means of the summary statistics at each observation time would be roughly normally distributed. However, combining networks into batches can delay the detection of a distributional shift in the summary statistics, as discussed in Zwetsloot and Woodall (2021).

To overcome the major limitations discussed above about the existing dynamic network monitoring methods, this paper aims to make two contributions. First, instead of using a dynamic network model with restrictive model assumptions to extract network features, we suggest monitoring the following four features of an individual network in a network sequence: i) the number of nodes in the network that represents the network size, ii) the average degree of all nodes in the network that represents the overall connectivity of the network, iii) the number of connected components in the network that represents the quantity of connected local communities in the network, and iv) the average diameter of all connected components in the network that represents the overall size of connected local communities. Their formal definitions will be given in Section 2. The combination of these four features should be sensitive to most structural changes of interest in a network sequence. As a special case, if the node set of a dynamic network does not change over time in an application, then only the last three features need to be monitored. Second, instead of using a conventional SPC chart with restrictive model assumptions, we suggest using a multivariate nonparametric CUSUM chart to monitor the four extracted features. Possible serial correlation among the values of the extracted features at different time points can be accommodated by this CUSUM chart. Numerical studies show that our proposed method is effective for online monitoring of dynamic networks in various cases considered.

The remainder of the paper is organized as follows. Section 2 describes the proposed dynamic network monitoring procedure, including the four network features mentioned above and a sequential network monitoring scheme, in detail. Section 3 presents some simulation results for evaluating the numerical performance of the proposed method, in comparison with some representative existing methods. Section 4 demonstrates the proposed method using three real data examples about sequential monitoring of three dynamic social systems. Some remarks conclude the paper in Section 5.

2 Proposed Method

Description of the proposed method for online monitoring of dynamic networks is given in three parts in this section. First, dynamic network processes are discussed in Subsection 2.1. Then, the four proposed network features are described in Subsection 2.2. Online monitoring of these network features is discussed in Subsection 2.3.

2.1 Dynamic network processes

Assume that a sequence of networks is under online monitoring and denoted as $\{G_t, t \geq 1\}$, where G_t denotes the network observed at the time point t. For each t, G_t consists of a set of nodes (or vertices) denoted as V_t and a set of edges (or links) denoted as E_t. The numbers of nodes and edges are denoted as $n_{v,t}$ and $n_{e,t}$, respectively. In the literature, a dynamic network G_t is often described by its $n_{v,t} \times n_{v,t}$ adjacency matrix $\mathbf{A}_t$ whose (i, j)th element $a_{ij,t}$ denotes the "weight" of the edge between the two nodes in the (i, j)th pair, for each $1 \leq i, j \leq n_{v,t}$, with $a_{ij,t} = 0$ implying no edge between the two related nodes. In this paper, we focus on undirected networks whose edges do not have directions. In such cases, $\mathbf{A}_t$ is a symmetric matrix with non-negative elements. In addition, self-loops are not considered (i.e., $a_{ii,t} = 0$ for each i), and it is assumed that the weights $a_{ij,t}$ are non-negative integers although the proposed method in this paper can be extended easily to cases when $a_{ij,t}$ are non-negative binary or real numbers. In the literature, a network with some weights larger than 1 is called a weighted network, and it is called an unweighted network if all weights equal 0 or 1 only. So, this paper focuses on weighted networks with all $\{a_{ij,t}\}$ values being non-negative integers. In such cases, for each i, j and t, $a_{ij,t}$ can be interpretted as the number of edges between the two nodes in the (i, j)th pair, and we have

$$n_{e,t} = \frac{1}{2} \sum_{i=1}^{n_{v,t}} \sum_{j=1}^{n_{v,t}} a_{ij,t}.$$

Figure 1 illustrates a dynamic network process representing social dynamics among 184 employees at the Enron Corporation during 1998-2002. In the figure, three snapshots of the dynamic network process on three specific days (all Mondays) are shown, in which each little circle denotes an Enron employee (i.e., node) and a thin line (i.e., edge) connecting two nodes implies that the two related employees had an email exchange on the specific day. From the figure, it can be seen that the dynamics of email communications among the Enron employees changed quite significantly over time. For instance, the network on October 22, 2001 has more edges than the other two networks, and it has a quite large connected local community (i.e., its nodes all connect to each other) located in the lower-left part. This phenomenon should be related to the events that Enron reported a \$638 million third quarter loss on October 16, 2001 and announced SEC probe on October 22, 2001 (Yu *et al.* 2018).

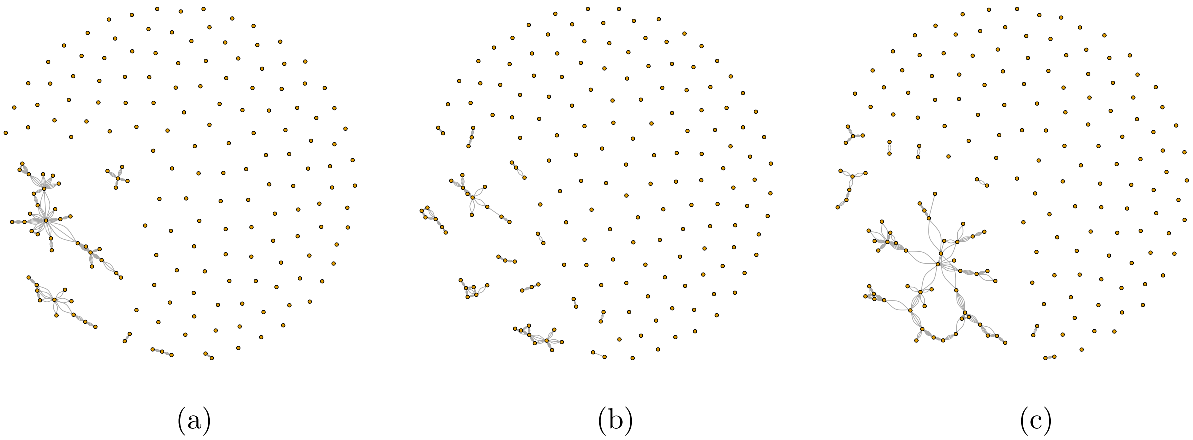

Figure 1: Networks of email communications among Enron employees on August 28, 2000 (plot (a)), December 4, 2000 (plot (b)), and October 22, 2001 (plot (c)).

2.2 Four suggested network features to monitor

To monitor a dynamic network process, it is nature to first extract certain features from the observed networks and then monitor the extracted network features over time. So, many different network features (or summary statistics) have been proposed in the past several decades (Guzman *et al.* 2014, Newman 2018, Sizemore and Bassett 2018). However, if the nature of dynamic networks is not considered carefully, the existing features could be misused, or they cannot effectively capture the structural changes in concern in the underlying dynamic network process. For instance, Segarra and Ribeiro (2015) pointed out that a feature called "betweenness centrality" of a node was unstable because its value is vulnerable to weight perturbation. Friedkin (1981) illustrated that the feature "network density", which is commonly used to measure network cohesion, is inappropriate to compare networks of different sizes. Therefore, proper selection of network features is critically important for online monitoring of dynamic networks.

In this paper, we suggest using the four network features described below for online monitoring of dynamic networks. The *first feature* is the number of nodes $n_{v,t}$ of the network G_t observed at time t. In practice, nodes in a network often represent entities of the underlying system. For a dynamic network system, new entities can enter the system and existing entities can leave the system over time. The quantity $n_{v,t}$ represents the overall size of the system, and the sequence $\{n_{v,t}, t \geq 1\}$ can describe the temporal growth of the system size. See Rossetti and Cazabet (2018) for an example about community growth. Proper monitoring of this feature is thus important. Besides the network size, another important aspect of the network structure is reflected in interactions among the nodes in the network that are represented by the network edges. In practice, edges are used to describe the traffic routes in transportation networks (cf., Rodríguez-Núñez and García-Palomares 2014), disease transmissions in infectious disease networks (cf., Liu *et al.* 2021), communications among a group of people in social networks (cf., Friedkin 1981, Sparks and Wilson 2019), and so forth. In all these examples, it is important to monitor the interactions among the nodes of the dynamic networks over time. In the literature, the number of edges connecting a given node with other nodes in the same network is called the *degree* of the given node. Then, the *second feature* is defined to be the average degree of all nodes in the network G_t, denoted as $\overline{d}_t$. Namely,

$$\overline{d}_t = \frac{2n_{e,t}}{n_{v,t}}.$$

The first two features discussed above reflect the size of a network and the connectivity among all nodes in the network. However, they cannot reflect other important aspects of the network structure. In the literature of network research, structure of connected components is considered to be central for describing network topology (Chung and Lu 2002, von Landesberger *et al.* 2009). For instance, a star-shaped connected component formed by linking isolated nodes to a common node in a network often represents the network structural change that a group of isolated individuals is organized by a leader (Neil *et al.* 2013). Following the task taxonomy discussed in Ahn *et al.* (2013), some typical structural changes of connected components are summarized in Figure 2. In the figure, "Growth" ("Contraction") of a connected component indicates increasing (decreasing) number of edges. "Merging" ("Splitting") connected components can result in less (more) connected components. "Birth" of a connected component implies a new connected component is created, "Death" of a connected component means an existing connected component disappears, and "Shape Change" of a connected component denotes the case when the numbers of nodes and edges do not change but the component structure changes. To describe the structure of connected components in the network G_t, the *third feature* we suggest using is the number of connected components in G_t, denoted as $n_{c,t}$.

From Figure 2, the number of connected components $n_{c,t}$ alone cannot describe the structure of all connected components in G_t well. For instance, in the case of "Shape Change" shown in the figure, although the two connected components have the same numbers of nodes and edges, their structure is quite different. The one at time t is star-shaped while the one at time $t+1$ is dumbbell-shaped. To describe a main structure of a connected component, we suggest using its *diameter*, defined to be the length of the longest geodesic path between any pair of nodes in the connected component. More specifically, the diameter of the jth connected component C_j in the

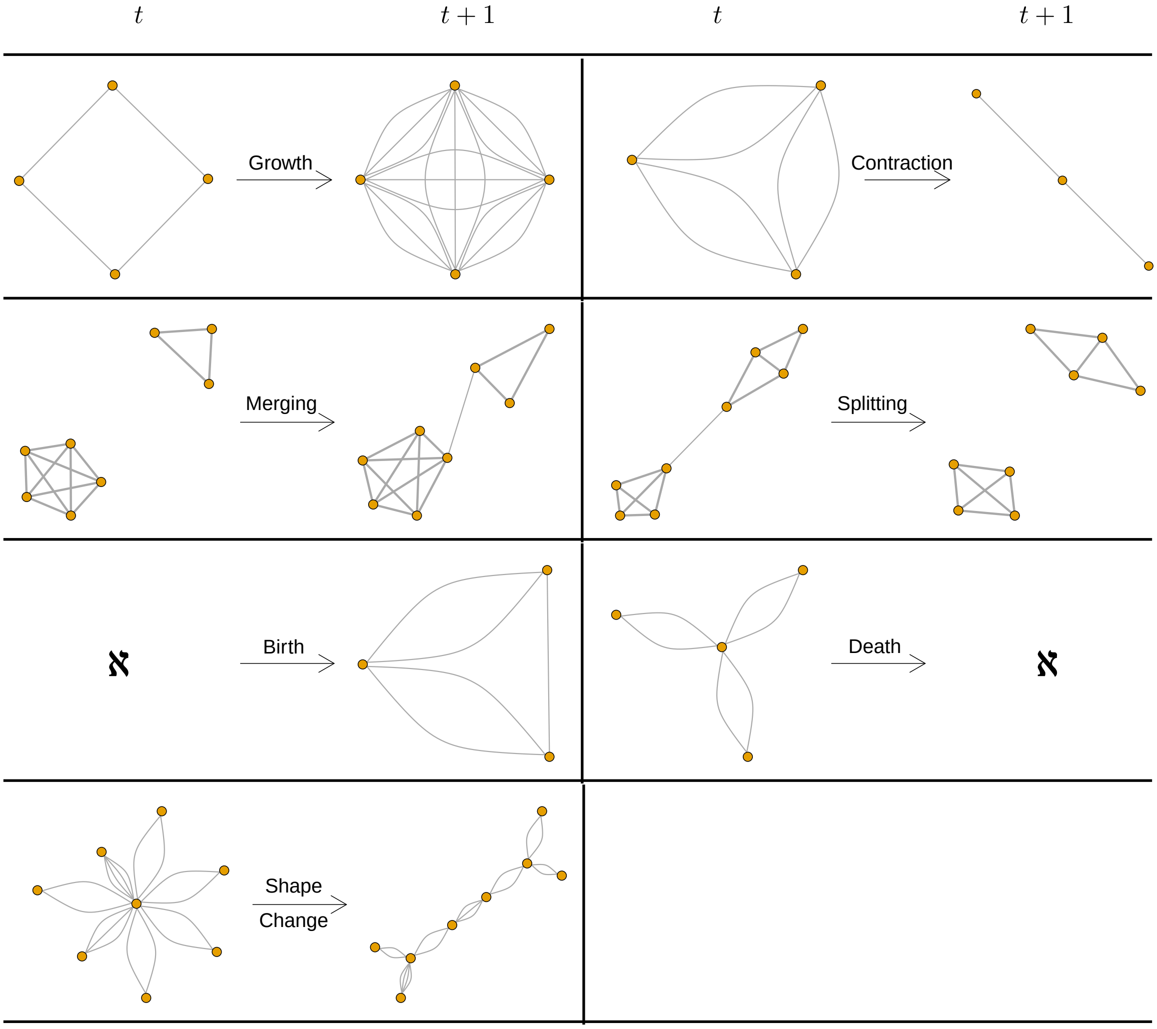

Figure 2: Typical structural changes of connected components. For birth or death of a connected component, the empty graph with no nodes is denoted as $\aleph$. In the demonstrated shape change of a connected component, the numbers of nodes and edges remain the same. For plots in the second row, the unchanged connected components at two time points are shown with thicker edges.

network G_t can be calculated by

$$\mathsf{l}_j = \max_{u,v \in C_j} \text{dis}(u, v),$$

where $\text{dis}(u, v)$ denotes the length of the shortest path between nodes u and v. Thus, each isolated node has the diameter of 0. In the example of “Shape Change” shown in Figure 2, the connected component at time t has the diameter of 2, and the connected component at time $t + 1$ has the diameter of 5. Then, the *fourth feature* that we suggest using is the average diameter of all connected components in G_t, denoted as $\bar{l}_t$. Namely,

$$\bar{l}_t = \frac{1}{n_{c,t}} \sum_{j=1}^{n_{c,t}} \mathsf{l}_j.$$

It can be checked that joint use of $n_{c,t}$ and $\bar{l}_t$ would be sensitive to all seven types of structural changes shown in Figure 2.

As a summary, we suggest using four features to monitor dynamic networks. The first feature (i.e., the number of nodes $n_{v,t}$) reflects the size of the network G_t, the second feature (i.e., the average degree $\overline{d}_t$ of all nodes in G_t) reflects the connectivity among all nodes in G_t, and the third and fourth features (i.e., the number of connected components $n_{c,t}$ and the average diameter $\bar{l}_t$ of all connected components in G_t) reflect the number and average size of the connected components of G_t. Most temporal changes in the structure of a dynamic network can be captured by one or more of these features. For instance, changes in communications among people in a social network would be reflected in the values of $\overline{d}_t$, $n_{c,t}$ and $\bar{l}_t$. A communication outbreak would increase the average degree $\overline{d}_t$, decrease the number of connected components $n_{c,t}$, and increase the average diameter $\bar{l}_t$. It should be pointed out that structural changes in a dynamic network are often reflected in multiple features discussed above. As an example, Figure 3 shows three snapshots of a dynamic network. It can be seen that the number of nodes, the average degree of all nodes, the number of connected components, and the average diameter of connected components all increase from the first network to the third network. In addition, the four features are usually correlated with each other. For instance, if the number of nodes $n_{v,t}$ is stable over time, then $n_{c,t}$ and $\bar{l}_t$ would be negatively correlated. If $n_{e,t}$ remains stable over time, then $\overline{d}_t$ and $n_{v,t}$ would be negatively correlated as well. Based on these considerations, the four features $(n_{v,t}, \overline{d}_t, n_{c,t}, \bar{l}_t)$ should be monitored together in order to detect structural changes in a dynamic network effectively.

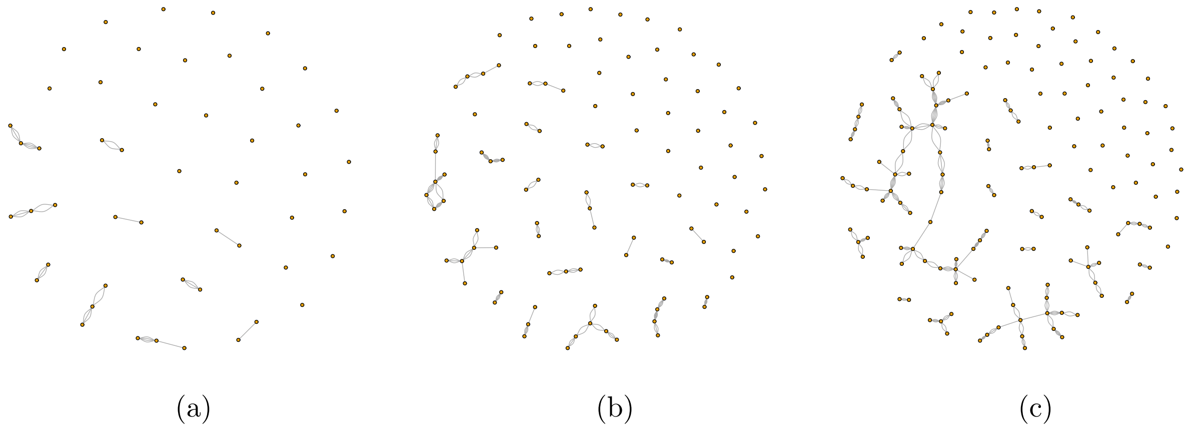

Figure 3: Three snapshots of a dynamic network at three different time points. The number of nodes, the average degree of all nodes, the number of connected components, and the average diameter of connected components all increase from plot (a) to plot (c).

In practice, we often do not know what type of structural change would occur in a dynamic network process, and joint monitoring of the four suggested features can capture main structural changes in many dynamic network process monitoring appllications, as discussed above. However, we would also like to point out that some structural changes can be left undetected by monitoring the four features alone. For instance, in the dumbbell-shaped connected component shown in the bottom row of Figure 2, if one edge of the two central nodes changes its location to become an edge of two nodes located at the upper-right corner of the connected component, then this small structural change cannot be detected by using the four suggested features. Thus, if such structural changes are also of our interest to detect in a given research, then the proposed method needs to be generalized, which is left for future research.

2.3 Online monitoring of a dynamic network

Based on the above discussions, for a dynamic network sequence $\{G_t, t \geq 1\}$, we suggest monitoring the following four features:

$$\mathbf{X}_t = \left(n_{v,t}, \bar{d}_t, n_{c,t}, \bar{l}_t\right)', \quad \text{for } t \geq 1.$$

In certain applications (e.g., the airline traffic network in US), the number of nodes (i.e., airports) may not change in a given period of time. In such applications where $n_{v,t}$ does not change over time, only the last three features need to monitor. Without loss of generality, our discussion below is for monitoring all four features, and the method for monitoring the last three features only can be discussed in a similar way.

As discussed in Subsection 2.2, the four features are usually correlated with each other. In addition, their observations at different time points would be serially correlated as well, although most existing methods for network monitoring assume that observed networks and their metrics at different time points are independent (e.g., Dong et al. 2020). In addition, it is rare in practice that the four features would follow a joint normal distribution. See Section 4 for three real-data examples where serial data correlation and non-normality are confirmed. To monitor the multivariate process $\{\mathbf{X}_t, t \geq 1\}$ properly, we suggest using the multivariate nonparametric CUSUM chart suggested by Xue and Qiu (2021). Its major steps are briefly described below.

To use the method in Xue and Qiu (2021), an initial IC data $\mathcal{X}_{\text{IC}}^{(0)} = \{\mathbf{X}_1, \ldots, \mathbf{X}_{m_0}\}$ is assumed to be available in advance, where m_0 is the sample size of the initial IC data. That method assumes that serial data correlation in the observed data is stationary and short-ranged when the process under monitoring is IC. More specifically, the covariance matrix $\boldsymbol{\gamma}(s) = \text{Cov}(\mathbf{X}_t, \mathbf{X}_{t+s})$, for any $t, s \geq 1$, is assumed to depend on s only when t changes. In addition, it is assumed that $\boldsymbol{\gamma}(s) = 0$ when $s > b_{\max}$, where $b_{\max} \geq 1$ denotes the range of serial correlation. This latter assumption implies that the correlation between two process observations decreases gradually when their observation times get farther away and the serial correlation can be ignored if the two observation times are at least $b_{\max}$ points away, which should be reasonable in practice. By the way, if the stationarity assumption is violated, then the alternative monitoring procedure discussed in Qiu and Xie (2022) can be considered.

Then, the IC mean $\boldsymbol{\mu}_0$ and the IC covariance matrices $\{\boldsymbol{\gamma}(s), 0 \leq s \leq b_{\max}\}$ of $\{\mathbf{X}_t, t \geq 1\}$ can be estimated from the initial IC data by the moment estimates as follows:

$$\begin{aligned} \widehat{\boldsymbol{\mu}}_0^{(0)} &= \frac{1}{m_0} \sum_{t=1}^{m_0} \mathbf{X}_t, \\ \widehat{\boldsymbol{\gamma}}^{(0)}(s) &= \frac{1}{m_0 - s} \sum_{t=1}^{m_0 - s} \left(\mathbf{X}_{t+s} - \widehat{\boldsymbol{\mu}}_0^{(0)}\right) \left(\mathbf{X}_t - \widehat{\boldsymbol{\mu}}_0^{(0)}\right)', \text{ for } 0 \leq s \leq b_{\max}. \end{aligned} \tag{1}$$

To sequentially monitor the multivariate observations $\{\mathbf{X}_t, t \geq m_0+1\}$, the observation $\mathbf{X}_t$ at the current time point t should be decorrelated with its previous observations $\{\mathbf{X}_n, t - b_{\max} \leq n \leq t-1\}$ in advance, because most existing online monitoring procedures are designed for monitoring processes with serially uncorrelated observations (cf., Qiu 2014). To further reduce computation, Xue and Qiu (2021) suggested using the *spring length* concept suggested originally in Chatterjee and Qiu (2009), where the spring length l_t at time t is defined to be the number of observation times between t and the last time when the CUSUM charting statistic (see definition below) is reset to 0. Because the CUSUM chart has the restarting mechanism that all observations collected before the time $t - l_t$ are ignored in the subsequent process monitoring, $\mathbf{X}_t$ only needs to be decorrelated with observations collected at the previous $b_t = \min\{l_{t-1}, b_{\max}\}$ time points, where l_{t-1}, instead of l_t, is used here since l_t is unavailable yet before a decision is made about the process status at time t. The resulting recursive algorithm for data decorrelation used in online process monitoring is briefly described below:

- When $t = m_0 + 1$, the standardized observation is defined to be $\mathbf{X}_t^* = [\widehat{\boldsymbol{\gamma}}^{(0)}(0)]^{-1/2}(\mathbf{X}_t - \widehat{\boldsymbol{\mu}}_0^{(0)})$. Stop the data decorrelation procedure if the CUSUM chart gives a signal at t. Otherwise, set $t = m_0 + 2$.

- When $t > m_0+1$, if $l_{t-1} = 0$, then define $\mathbf{X}_t^* = [\widehat{\boldsymbol{\gamma}}^{(t-m_0-1)}(0)]^{-1/2}(\mathbf{X}_t - \widehat{\boldsymbol{\mu}}_0^{(t-m_0-1)})$. Otherwise, the estimated covariance matrix of $(\mathbf{X}'_{t-l_{t-1}}, \ldots, \mathbf{X}'_t)'$ can be defined to be

$$\widehat{\mathbf{\Sigma}}_{t,t} = \begin{pmatrix} \widehat{\boldsymbol{\gamma}}^{(t-m_0-1)}(0) & \ldots & \widehat{\boldsymbol{\gamma}}^{(t-m_0-1)}(l_{t-1}) \\ \vdots & \ddots & \vdots \\ \left[\widehat{\boldsymbol{\gamma}}^{(t-m_0-1)}(l_{t-1})\right]' & \ldots & \widehat{\boldsymbol{\gamma}}^{(t-m_0-1)}(0) \end{pmatrix} = \begin{pmatrix} \widehat{\mathbf{\Sigma}}_{t-1,t-1} & \widehat{\boldsymbol{\sigma}}_{t-1} \\ \left[\widehat{\boldsymbol{\sigma}}_{t-1}\right]' & \widehat{\boldsymbol{\gamma}}^{(t-m_0-1)}(0) \end{pmatrix},$$

where $\widehat{\boldsymbol{\sigma}}_{t-1} = ([\widehat{\boldsymbol{\gamma}}^{(t-m_0-1)}(l_{t-1})]', \ldots, [\widehat{\boldsymbol{\gamma}}^{(t-m_0-1)}(1)]')'$, $\widehat{\boldsymbol{\mu}}_0^{(t-m_0-1)}$ and $\{\widehat{\boldsymbol{\gamma}}^{(t-m_0-1)}(s), 0 \le s \le l_{t-1}\}$ can be calculated from Equation (3) below. Then, the decorrelated and standardized observation at time t is defined to be

$$\mathbf{X}_t^* = \widehat{\boldsymbol{D}}_t^{-1/2}\left[\mathbf{X}_t - \widehat{\boldsymbol{\mu}}_0^{(t-m_0-1)} - \widehat{\boldsymbol{\sigma}}'_{t-1}\widehat{\mathbf{\Sigma}}_{t-1,t-1}^{-1}\widehat{\mathbf{e}}_{t-1}\right],$$

where $\widehat{\boldsymbol{D}}_t = \widehat{\boldsymbol{\gamma}}^{(t-m_0-1)}(0) - \widehat{\boldsymbol{\sigma}}'_{t-1}\widehat{\mathbf{\Sigma}}_{t-1,t-1}^{-1}\widehat{\boldsymbol{\sigma}}_{t-1}$, and $\widehat{\mathbf{e}}_{t-1} = [(\mathbf{X}_{t-l_{t-1}} - \widehat{\boldsymbol{\mu}}_0^{(t-m_0-1)})', \ldots, (\mathbf{X}_{t-1} - \widehat{\boldsymbol{\mu}}_0^{(t-m_0-1)})']'$. If the CUSUM charting statistic (cf., Equation (2) below) at t is 0, then set $l_t = 0$. Otherwise, define $l_t = \min(l_{t-1}+1, b_{\max})$. Stop the iterative procedure if the chart gives a signal. Otherwise, set $t = t+1$, and repeat this step until the control chart gives a signal.

In practice, the IC distribution of the decorrelated data $\{\mathbf{X}_t^*, t \ge m_0+1\}$ could be substantially different from normal in cases when the distribution of the original data $\{\mathbf{X}_t, t \ge m_0+1\}$ is substantially different from normal. Thus, for the p-dimensional decorrelated data $\{\mathbf{X}_t^* = (X_{t1}^*, \ldots, X_{tp}^*)', t \ge m_0+1\}$ with $p = 4$, the multivariate nonparametric CUSUM chart suggested originally by Qiu (2008) can be considered. To this end, let $\{m_j^*, j = 1, \ldots, p\}$ be the IC medians of the p components of $\mathbf{X}_t^*$, and $\mathbf{Y}_t^* = (Y_{t1}^*, \ldots, Y_{tp}^*)'$ be the categorized version of $\mathbf{X}_t^*$, where

$$Y_{tj}^* = I(X_{tj}^* > m_j^*), \qquad \text{for } 1 \le j \le p,\ t \ge m_0+1,$$

and $I(u)$ is the indicator function that equals 0 and 1, respectively, when u is FALSE and TRUE. Let the IC distribution of $\mathbf{Y}_t^*$ be $\mathbf{f}^{(0)}$, which is a long vector with the elements

$$f_{j_1,\ldots,j_p}^{(0)} = P(Y_{t1}^* = j_1, \ldots, Y_{tp}^* = j_p), \quad \text{for } j_1, \ldots, j_p = 0, 1,$$

and $\mathbf{g}(t)$ be the corresponding long vector with elements

$$g_{j_1,\ldots,j_p}(t) = I(Y_{t1}^* = j_1, \ldots, Y_{np}^* = j_p), \quad \text{for } j_1, \ldots, j_p = 0, 1,$$

arranged in the same order as that of $\mathbf{f}^{(0)}$. Then, $\mathbf{g}(t)$ is the empirical distribution of $\mathbf{Y}_t^*$. By comparing the empirical distribution $\mathbf{g}(t)$ and the IC distribution $\mathbf{f}^{(0)}$ in a cumulative manner, the multivariate nonparametric CUSUM charting statistic is defined to be

$$C_t = \left(\mathbf{S}_t^{\text{obs}} - \mathbf{S}_t^{\text{exp}}\right)' [diag(\mathbf{S}_t^{\text{exp}})]^{-1} \left(\mathbf{S}_t^{\text{obs}} - \mathbf{S}_t^{\text{exp}}\right), \tag{2}$$

where

$$\begin{cases} \mathbf{S}_0^{\text{obs}} = \mathbf{S}_0^{\text{exp}} = \mathbf{0}, & \text{if } U_t \leq k, \\ \mathbf{S}_t^{\text{obs}} = \left[\mathbf{S}_{t-1}^{\text{obs}} + \mathbf{g}(t)\right](U_t - k)/U_t, & \text{if } U_t > k, \\ \mathbf{S}_t^{\text{exp}} = \left[\mathbf{S}_{t-1}^{\text{exp}} + \mathbf{f}^{(0)}\right](U_t - k)/U_t, & \text{if } U_t > k, \end{cases}$$

$$U_t = \left[\left(\mathbf{S}_{t-1}^{\text{obs}} - \mathbf{S}_{t-1}^{\text{exp}}\right) + \left(\mathbf{g}(t) - \mathbf{f}^{(0)}\right)\right]' \left[diag\left(\mathbf{S}_{t-1}^{\text{exp}} + \mathbf{f}^{(0)}\right)\right]^{-1} \left[\left(\mathbf{S}_{t-1}^{\text{obs}} - \mathbf{S}_{t-1}^{\text{exp}}\right) + \left(\mathbf{g}(t) - \mathbf{f}^{(0)}\right)\right],$$

$diag(\mathbf{A})$ denotes a diagonal matrix in which the diagonal elements being those of $\mathbf{A}$, and $k > 0$ is an allowance constant. Then, the chart gives a signal when

$$C_t > h,$$

where $h > 0$ is a control limit. Since different components of $\mathbf{X}_t^*$ have been decorrelated, the components of $\mathbf{Y}_t^*$ would be roughly independent. Thus, each component of the IC distribution $\mathbf{f}^{(0)}$ can be specified to be 2^{-p}. For a given value of the IC average run length, denoted as ARL_0, once the value of k is pre-specified, the control limit h can be determined by a Monte Carlo simulation as described in Appendix A.1 of the supplementary file.

When the CUSUM chart (2) does not give a signal at time t, the process under monitoring would be claimed to be IC. Thus, the observation $\mathbf{X}_t$ can be combined with the IC dataset $\{\mathbf{X}_1, \ldots, \mathbf{X}_{t-1}\}$ at the time $t-1$. Then, the estimates of the IC parameters $\boldsymbol{\mu}_0$ and $\{\boldsymbol{\gamma}(s), 0 \leq s \leq b_{\max}\}$ can be updated recursively using the combined IC data as follows: for $t \geq m_0 + 1$ and $0 \leq s \leq b_{\max}$,

$$\begin{aligned} \widehat{\boldsymbol{\mu}}_0^{(t-m_0)} &= \frac{1}{t}\mathbf{X}_t + \frac{t-1}{t}\widehat{\boldsymbol{\mu}}_0^{(t-m_0-1)}, \\ \widehat{\boldsymbol{\gamma}}^{(t-m_0)}(s) &= \frac{1}{t-s}\left(\mathbf{X}_t - \widehat{\boldsymbol{\mu}}_0^{(t-m_0)}\right)\left(\mathbf{X}_{t-s} - \widehat{\boldsymbol{\mu}}_0^{(t-m_0)}\right)' + \frac{t-s-1}{t-s}\widehat{\boldsymbol{\gamma}}^{(t-m_0-1)}(s). \end{aligned} \tag{3}$$

In Equation (3), $\widehat{\boldsymbol{\mu}}_0^{(0)}$ and $\{\widehat{\boldsymbol{\gamma}}^{(0)}(s), 0 \leq s \leq b_{\max}\}$ are defined in Equation (1). The IC medians $\{m_j^*, j = 1, \ldots, p\}$ used in computing the charting statistic C_t in Equation (2) can be estimated by the sample medians of the IC dataset, and the sample medians $\{\widehat{m}_j^*, j = 1, \ldots, p\}$ at the current time point t can be obtained recursively by updating the sample medians at the time $t-1$, after comparing their values with the decorrelated observations at the times $t-2$ and t. Regarding the parameter $b_{\max}$, based on an extensive numerical study, Xue and Qiu (2021) suggested choosing it in the interval $[10, 20]$. Thus, in all numerical examples in Sections 3 and 4, $b_{\max}$ is chosen to be 20.

Because the estimates of the IC parameters have been updated during the online process monitoring process, the entire process monitoring procedure is a self-starting procedure (cf., Hawkins 1987). For this reason, our proposed process monitoring procedure described in this section is denoted as SS-MCUSUM, where SS denotes "self-starting" and MCUSUM implies that the major charting scheme (2) is a multivariate CUSUM chart.

3 Simulation Studies

In this section, we investigate the numerical performance of the proposed method for dynamic network monitoring using Monte Carlo simulations, in comparison with six representative existing methods that are discussed in Subsection 3.1. The method to generate a sequence of networks is described in Subsection 3.2, and the simulation results are presented in Subsection 3.3.

3.1 Six representative existing network monitoring methods

One straightforward network monitoring strategy is to use a summary statistic X_t to describe the observed network G_t, and then sequentially monitor the sequence $\{X_t, t \geq 1\}$. To this end, a series of papers used the EWMA chart to detect anomalous changes in a dynamic network process by using various summary statistics (e.g., Hosseini and Noorossana 2018; Flossdorf and Jentsch 2021, Yu *et al.* 2022). More specifically, the charting statistic is defined to be

$$E_t = \lambda(X_t - \mu_0) + (1 - \lambda)E_{t-1}, \quad \text{for } t \geq m_0 + 1,$$

where $\lambda \in (0, 1]$ is a weighting parameter, $E_{m_0} = 0$, and μ_0 is the IC mean of X_t. In all the papers mentioned above, their EWMA charts are designed for cases when IC process observations are assumed to be independent and identically distributed (i.i.d.) at different observation times with a normal IC distribution. These assumptions are rarely valid in practice. But, some researchers think that the EWMA chart is robust to some of these assumptions if the weighting parameter λ is chosen properly (e.g., Borror *et al.* 1999, Horng Shiau and Ya-Chen 2005). Under the assumptions mentioned above, the control limit of the above EWMA chart can be determined by Monte Carlo simulations to reach a given ARL_0 value. See Table 5.1 in Qiu (2014) for the control limit values corresponding to some commonly used ARL_0 and λ values. In this section, we consider three such EWMA charts. Two were suggested by Yu *et al.* (2022) for monitoring the maximum degree ($d_{\max}$) and the average degree ($\bar{d}$), respectively, of a dynamic network, and the other was discussed in Flossdorf and Jentsch (2021) for monitoring the spectral norm (SN) of the adjacency matrix of the dynamic network. These three methods are denoted as EWMA-$d_{\max}$, EWMA-$\bar{d}$, and EWMA-SN, respectively.

In cases when the node set of a dynamic network is fixed over time, a number of researchers have proposed scan-based network monitoring schemes. One well-known method was proposed by Priebe *et al.* (2005), which used the temporal moving-window idea. More specifically, at the current time t, let us consider all previous networks in the time window $\{t-L,\ldots,t-1\}$ of length L. For a given node $i \in \{1,\ldots,n_v\}$ (note: the notation $n_{v,t}$ has been simplified to n_v here since the node set does not change with t), the total number of edges of all nodes within its lth neighborhood at time t is denoted as $O^l_{t,i}$, for $l = 0,1,2$, where the lth neighborhood of a given node is defined to be the set of nodes in a network whose shortest paths to the given node are at most l. Then, the quantities $\{O^l_{t,i}\}$ are first normalized as follows:

$$\widetilde{O}^l_{t,i} = \frac{O^l_{t,i} - \widehat{\mu}^l_{O,t,i}}{\max\left(\widehat{\sigma}^l_{O,t,i}, 1\right)},$$

where $\widehat{\mu}^l_{O,t,i} = \sum_{k=1}^{L} O^l_{t-k,i}/L$ and $\widehat{\sigma}^l_{O,t,i} = \sqrt{\sum_{k=1}^{L}(O^l_{t-k,i} - \widehat{\mu}^l_{O,t,i})^2/(L-1)}$. The denominator in the above expression sets a lower bound of 1 to avoid fragility for nodes with little or no variation in activity over time. Then, we define

$$S^l_t = \max\left\{\widetilde{O}^l_{t,1},\ldots,\widetilde{O}^l_{t,n_v}\right\}, \quad \text{for } l = 0,1,2.$$

For the quantities $\{S^l_t\}$, they are standardized as follows:

$$\widetilde{S}^l_t = \frac{S^l_t - \widehat{\mu}^l_{S,t}}{\max\left(\widehat{\sigma}^l_{S,t}, 1\right)},$$

where $\widehat{\mu}^l_{S,t} = \sum_{k=1}^{L} S^l_{t-k}/L$ and $\widehat{\sigma}^l_{S,t} = \sqrt{\sum_{k=1}^{L}(S^l_{t-k} - \widehat{\mu}^l_{S,t})^2/(L-1)}$. Then, the charting (scan) statistic is

$$S_t = \max\left\{S^0_t, S^1_t, S^2_t\right\},$$

and the chart gives a signal when $S_t > h_p$. Priebe *et al.* (2005) pointed out that the IC distribution of the charting scan statistic S_t was usually unknown and its values at different time points could be serially correlated. So, a block bootstrap procedure is used in this paper to determine the control limit h_p, which is described in Appendix A.2 of the supplementary file. The above network monitoring method is denoted as PSCAN.

When the node set of a dynamic network does not change over time, Chen (2019) proposed another online network monitoring procedure based on change-point detection, which is briefly described below. At time t, consider a time window $\{t-L+1,\ldots,t\}$ of length L. For a potential change-point at $t-L+1 \le n \le t$ in the time window, let us compare the two groups of networks:

$\{G_{t-L+1}, \ldots, G_n\}$ and $\{G_{n+1}, \ldots, G_t\}$. For a given network, if another network is one of its ρ nearest neighbors judged by a similarity measure between the two networks, then we claim that there is an edge connecting the two networks. This graphic structure is called the K-nearest-neighbor (KNN) graph in the literature (cf., Schilling 1986, Henze 1988). Let $Z_L(n,t)$ be a pre-specified decreasing function of the number of edges connecting the networks in $\{G_{t-L+1}, \ldots, G_n\}$ and the networks in $\{G_{n+1}, \ldots, G_t\}$. Then, the likelihood that n is a change-point would be relatively large if the value of $Z_L(n,t)$ is large. Based on this intuition, the change-point detection (CPD) chart gives a signal when

$$\max_{t-t_1 \leq n \leq t-t_0} Z_L(n,t) > b_Z,$$

where t_1 and t_0 are two pre-specified small positive integers, and b_Z is a control limit. To determine the control limit b_Z to reach a given ARL_0 value, Chen (2019) recommended using an analytic formula. However, the integrand in the formula can be infinite and thus a reasonable b_Z value would not be well defined in such cases. In this paper, we suggest an alternative approach to determine b_Z from the set of IC networks. See Appendix A.3 of the supplementary file for details. The above network monitoring method is denoted as CPD-KNN hereafter. In CPD-KNN, the similarity measure between two networks at times t and s is defined to be

$$\frac{\|\mathbf{A}_t - \mathbf{A}_s\|_F^2}{\|\mathbf{A}_t\|_F \|\mathbf{A}_s\|_F},$$

where $\mathbf{A}_t$ and $\mathbf{A}_s$ are the adjacency matrices of the related networks, and $\|\cdot\|_F$ is the Frobenius norm of a matrix. Chen (2019) suggested choosing $\rho = 5$, $L = 50$, $t_0 = 3$, and $t_1 = L - t_0$, which has been adopted in all numerical studies in this paper.

If the fixed node set of a dynamic network is assumed to follow a two-community structure (Lancichinetti *et al.* 2008), Yu *et al.* (2022) (also see Wilson *et al.* (2019)) suggested to use a Shewhart chart to monitor the unique entries of the estimated propensity matrix $\widehat{\mathfrak{P}}_t$ (cf., Section B in the supplementary file for a related discussion) for detecting structural changes in the network. More specifically, for communities c_1 and c_2, the number of nodes in the two communities are denoted as n_{c_1} and n_{c_2}, respectively. At time t, the estimated entries of $\widehat{\mathfrak{P}}_t$ are defined to be $\widehat{\mathfrak{P}}_{c_1,c_2} = w_{c_1,c_2}/(n_{c_1} n_{c_2})$, for $c_1, c_2 = 1, 2$, where w_{c_1,c_2} is the sum of all edges between the communities c_1 and c_2. Then, each one of the three unique entries in $\widehat{\mathfrak{P}}_t$ is used as the charting statistic of the Shewhart chart and it gives a signal if the statistic value is beyond the control limits $\widehat{\mu} \pm 3\widehat{\sigma}$, where $\widehat{\mu}$ and $\widehat{\sigma}$ are the estimated mean and standard deviation of the charting statistic obtained from the m_0 IC networks. A structural change of the network is detected if any one of the three individual charts gives a signal. This method is denoted as Shewhart.

3.2 Generation of network sequences for Monte Carlo simulations

In this paper, a dynamic network model is used when generating a network sequence for simulation studies, by which the network topology can be controlled by changing parameter values in the model (cf., Snijders 2005, Peel and Clauset 2015). First, the following Poisson hurdle model is used to specify the edge distribution of a network: for $1 \leq i \neq j \leq n_{v,t}$ and $t \geq 1$,

$$P(a_{ij,t} = u) = \begin{cases} \pi_t, & \text{if } u = 0, \\ \frac{(1-\pi_t)\exp(-\omega_t){\omega_t}^u}{u![1-\exp(-\omega_t)]}, & \text{if } u > 0, \end{cases} \tag{4}$$

where π_t is the probability of no edge between the two nodes in the (i, j)th pair, and the connectivity parameter ω_t controls the edge weight. Equation (4) is the mixture of a Bernoulli distribution with the parameter π_t and a Positive Poisson distribution with the parameter ω_t (Grogger and Carson 1991), and the value of π_t is large (or small) for a sparse (or dense) network.

Then, the network sequence $\{G_t, 1 \leq t \leq T\}$ is generated as follows. First, let $\overline{n}_v$ be a pre-specified integer (e.g., 100) representing the expected IC network size. The number of nodes of G_t at time t is then determined by $n_{v,t} = \max\{10, \langle \overline{n}_v + \xi_t \rangle\}$, where $\langle \cdot \rangle$ rounds a value to the nearest integer, and ξ_t is generated by the AR(1) model

$$\xi_t = 0.5\xi_{t-1} + \epsilon_{t1},$$

where $\xi_0 = 0$ and $\{\epsilon_{t1}\}$ are i.i.d. random numbers generated from the $N(0, 5^2)$ distribution. Second, determine the value of π_t used in Equation (4) as follows. At time t, the expected number of nodes that are not isolated is denoted as $\mathfrak{n}_{v,t}$. Then, $\mathfrak{n}_{v,t}$ is determined by $\mathfrak{n}_{v,t} = \langle p_t n_{v,t} \rangle$, where p_t is randomly generated from the uniform distribution on the interval $[p_1, p_2]$, and p_1 and p_2 represent the lower and upper proportions of nodes that are expected to have edges in the network. The number of expected binary edges is denoted as $\mathfrak{n}_{e,t}$. Then, its maximum value is $\mathfrak{n}_e^{\max} = \mathfrak{n}_{v,t}(\mathfrak{n}_{v,t} - 1)/2$, and its minimum value is $\mathfrak{n}_e^{\min} = (\mathfrak{n}_{v,t} + 1)/2$ if $\mathfrak{n}_{v,t}$ is an odd number and $\mathfrak{n}_e^{\min} = \mathfrak{n}_{v,t}/2$ if $\mathfrak{n}_{v,t}$ is an even number. Then, $\mathfrak{n}_{e,t}$ is chosen to be

$$\mathfrak{n}_{e,t} = \begin{cases} \langle \mathfrak{n}_{v,t} + \epsilon_{t2} \rangle, & \text{if } \mathfrak{n}_e^{\min} \leq \langle \mathfrak{n}_{v,t} + \epsilon_{t2} \rangle \leq \mathfrak{n}_e^{\max}, \\ \mathfrak{n}_e^{\min}, & \text{if } \mathfrak{n}_e^{\min} > \langle \mathfrak{n}_{v,t} + \epsilon_{t2} \rangle, \\ \mathfrak{n}_e^{\max}, & \text{if } \mathfrak{n}_e^{\max} < \langle \mathfrak{n}_{v,t} + \epsilon_{t2} \rangle, \end{cases} \tag{5}$$

where $\{\epsilon_{t2}\}$ are i.i.d. random numbers generated from the $N(0, 2^2)$ distribution. Equation (5) has made use of the linear relationship between the number of non-isolated nodes and the number of binary edges observed in real temporal networks (cf., Leskovec and Krevl 2014). After $\mathfrak{n}_{e,t}$ is determined, the value of the sparsity parameter is chosen to be

$$\pi_t = 1 - \frac{2\mathfrak{n}_{e,t}}{n_{v,t}(n_{v,t} - 1)}.$$

Third, determine the value of ω_t used in Equation (4). To this end, let μ_e be a pre-specified value (e.g., 2) of the expected degree of a node. Based on the Poisson hurdle model, this number can be calculated to be $(n_{v,t} - 1)E(a_{ij,t})$, which is the sum of the expected edge weights between node i and all other nodes in G_t. Then, from the equation $\mu_e = (n_{v,t} - 1)E(a_{ij,t})$ and (4), we have

$$\frac{\mu_e}{(n_{v,t} - 1)(1 - \pi_t)} = \frac{\omega_t}{1 - \exp(-\omega_t)}.$$

Since $n_{v,t}$ and π_t in the above expression have been determined in advance, the value of ω_t can be determined by solving the above equation, which is unique when $\pi_t > 1 - \mu_e/(n_{v,t} - 1)$.

After the values of $\{n_{v,t}, \pi_t, \omega_t\}$ are determined as described above, the adjacency matrix $\mathbf{A}_t$ can be generated by Equation (4) at each observation time t, and then a network sequence can be generated afterwards. From the above description, it can be seen that the structure of the network sequence generated based on the Poisson hurdle model is uniquely determined by the pre-specified values of $\{\overline{n}_v, p_1, p_2, \mu_e\}$.

Following the network generation framework in Yu *et al.* (2022), we also use the hurdle degree corrected stochastic block model (HDCSBM) to generate the network sequence in some examples. The simulated networks are assumed to follow the two-community structure, and the community structure is controlled by the tuning parameters in a propensity matrix. See Appendix B of the supplementary file for details.

3.3 Simulation results

In this part, we present some simulation examples, in which the nominal ARL_0 values of all network monitoring methods are fixed at 200. If there is no further specification, the allowance constant k in the SS-MCUSUM chart is chosen to be 0.1, the weighting parameter λ in the EWMA-$d_{\max}$, EWMA-$\bar{d}$, and EWMA-SN charts is fixed at 0.05 as suggested by Yu *et al.* (2022), the moving window size L in PSCAN is chosen to be 20 as suggested by Priebe *et al.* (2005), and the number of nearest neighbors ρ used in CPD-KNN is chosen to be 5 as suggested by Chen (2019).

To evaluate the IC performance of the related methods, the following eleven cases are considered, among which the Poisson hurdle model is used for generating network sequences in the first seven cases and the HDCSBM model is used in the remaining cases.

Case I: The expected network size is set to be $\overline{n}_v = 100$, the expected proportion of non-isolated nodes in the network is chosen between $p_1 = 0.2$ and $p_2 = 0.4$, and the expected degree of a node in the network is set to be $\mu_e = 2$.

Case II: The four parameters are chosen to be $\overline{n}_v = 100$, $p_1 = 0.6$, $p_2 = 0.8$, and $\mu_e = 4$.

Case III: Same as Case I, except that the distribution of $\{\epsilon_{t1}\}$ used in generating $n_{v,t}$ is the one of $5(\xi - 3)/\sqrt{6}$, and the distribution of $\{\epsilon_{t2}\}$ used in generating $\mathfrak{n}_{e,t}$ is the one of $2(\xi - 3)/\sqrt{6}$, where ξ is a random variable with the distribution χ^2_3.

Cases IV–VI: Same as Cases I–III, respectively, except that the node set is fixed with $n_{v,t} \equiv 100$.

Cases VII–XI: Each network has the two-community structure with the propensity matrix $\mathfrak{P}$ specified as $\begin{pmatrix} 0.12 & 0.04 \\ 0.04 & 0.12 \end{pmatrix}$ in all Cases VII, IX, and XI, and $\begin{pmatrix} 0.08 & 0.08 \\ 0.08 & 0.08 \end{pmatrix}$ and $\begin{pmatrix} 0.16 & 0.08 \\ 0.08 & 0.04 \end{pmatrix}$ in Cases VIII and X, respectively. Other simulation settings in these cases can be found in Appendix B.

In Case I, the network size varies around 100 over time, and the generated networks are quite sparse since all p_1, p_2 and μ_e are quite small. Compared to Case I, the generated networks in Case II are relatively dense since the values of p_1, p_2 and μ_e are chosen larger than those in Case I. Case III is the same as Case I, except that the distributions of $n_{v,t}$ and $\mathfrak{n}_{e,t}$ in the former case are quite skewed. Dynamic networks with fixed node sets are considered in Cases IV–VI. Cases VII–XI consider various scenarios when the network sequence has the fixed node set that has the two-community structure.

Evaluation of the IC performance: We first evaluate the IC performance of the related network monitoring methods. In cases when the node set can change over time (i.e., Cases I–III), we compare the proposed method SS-MCUSUM with EWMA-$d_{\max}$, EWMA-$\bar{d}$, and EWMA-SN only, since the other methods PSCAN, CPD-KNN, and Shewhart are designed for cases with fixed node set. When the IC sample size m_0 changes among $\{200, 300, 400, 500, 1,000\}$, the estimated actual ARL$_0$ values of the four methods SS-MCUSUM, EWMA-$d_{\max}$, EWMA-$\bar{d}$, and EWMA-SN are presented in Table 1. For each method, the actual ARL$_0$ value is estimated as follows. First, after m_0 IC networks are generated, some IC parameters used by the method are estimated from the IC networks. Second, the monitoring method is then applied to a sequence of 2,000 IC networks for online process monitoring, and its run length (RL) value is recorded. The online process monitoring is then repeated for 1,000 times, and the average of the 1,000 RL values is used as an estimate of the actual conditional ARL_0 value, conditional on the m_0 IC networks. Finally, to obtain an estimate of the actual (unconditional) ARL_0 value, all steps described above, starting from the generation of the m_0 IC networks to the computation of the estimated actual conditional ARL_0 value, are repeated for 100 times. The actual ARL_0 value of the monitoring method is then estimated by the average of the 100 estimates of the actual conditional ARL_0 value.

Table 1: Estimated actual ARL_0 values and their standard errors (in parentheses) of four control charts in Cases I-III when their nominal ARL_0 values are fixed at 200, and the number of IC networks m_0 in the initial IC data changes among 200, 300, 400, 500, and 1,000.

Case	Methods	$m_0 = 200$	300	400	500	1000
I	SS-MCUSUM	218 (3.36)	216 (3.58)	216 (3.22)	211 (2.89)	204 (2.42)
	EWMA-$d_{\max}$	176 (5.75)	177 (4.99)	181 (4.30)	183 (4.06)	192 (2.91)
	EWMA-$\bar{d}$	181 (5.48)	180 (4.20)	182 (3.67)	183 (3.03)	193 (2.42)
	EWMA-SN	174 (5.22)	180 (4.65)	181 (3.93)	186 (3.97)	193 (2.71)
II	SS-MCUSUM	225 (3.21)	220 (2.71)	222 (2.77)	218 (2.72)	209 (2.61)
	EWMA-$d_{\max}$	179 (5.70)	184 (5.68)	186 (3.91)	190 (3.91)	195 (3.15)
	EWMA-$\bar{d}$	174 (5.36)	180 (4.66)	178 (3.37)	187 (3.73)	199 (2.66)
	EWMA-SN	173 (5.48)	176 (4.76)	181 (3.58)	184 (3.11)	193 (2.31)
III	SS-MCUSUM	204 (3.89)	205 (3.53)	205 (3.68)	202 (3.20)	201 (2.21)
	EWMA-$d_{\max}$	175 (5.57)	185 (6.15)	187 (4.73)	186 (4.40)	194 (2.78)
	EWMA-$\bar{d}$	178 (5.65)	186 (4.76)	186 (3.48)	185 (2.99)	193 (2.37)
	EWMA-SN	179 (6.13)	183 (4.61)	186 (4.21)	189 (3.86)	194 (2.42)

For Table 1, we can have the following conclusions. (i) The IC performance of all four charts is quite reliable in Cases I–III when $m_0 \geq 500$ since their estimated actual ARL_0 values are within 10% of the nominal ARL_0 value of 200 in all cases considered. (ii) It seems that SS-MCUSUM is bit better than the other three charts in Cases I and III when $m_0 \geq 500$ while its advantage disappears in Case II when the networks are dense. This example shows that in cases when the node set can change over time, all the charts SS-MCUSUM, EWMA-$d_{\max}$, EWMA-$\bar{d}$, and EWMA-SN seem to have a quite reliable IC performance when $m_0 \geq 500$ in the simulation settings considered.

Note that the weighting parameter is chosen to be 0.05 in EWMA-$d_{\max}$, EWMA-$\bar{d}$, and EWMA-SN. In cases considered in Table 1, if the weighting parameter is chosen to be one of $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ in these three charts, then their estimated actual ARL$_0$ values when $m_0 = 500$ are presented in Table 2. From the table, it can be seen that the IC performance of the three charts are quite sensitive to the weighting parameter value, and their IC performance becomes unreliable when the weighting parameter is chosen to be larger than or equal to 0.3.

Table 2: Estimated actual ARL$_0$ values and their standard errors (in parentheses) of the three EWMA charts when their nominal ARL$_0$ values are fixed at 200, $m_0 = 500$, and the weighting parameter changes among 0.1, 0.2, 0.3, 0.4, and 0.5.

Case	Methods	$\lambda = 0.1$	0.2	0.3	0.4	0.5
I	EWMA-$d_{\max}$	179 (4.62)	157 (4.36)	132 (3.51)	113 (2.80)	99 (2.28)
	EWMA-$\bar{d}$	184 (3.65)	180 (4.17)	174 (4.26)	166 (4.18)	158 (3.97)
	EWMA-SN	185 (4.78)	175 (5.05)	158 (4.57)	141 (3.93)	126 (3.32)
II	EWMA-$d_{\max}$	188 (4.62)	166 (5.36)	145 (4.67)	126 (3.84)	110 (3.15)
	EWMA-$\bar{d}$	193 (3.96)	185 (3.70)	184 (3.90)	182 (4.08)	179 (4.06)
	EWMA-SN	190 (4.51)	182 (5.48)	173 (5.45)	161 (5.09)	149 (4.63)
III	EWMA-$d_{\max}$	183 (4.74)	166 (5.43)	142 (4.60)	120 (3.59)	105 (2.92)
	EWMA-$\bar{d}$	186 (4.24)	184 (5.18)	179 (5.35)	172 (5.47)	164 (5.22)
	EWMA-SN	189 (3.78)	188 (5.58)	177 (6.23)	160 (5.82)	144 (5.15)

Next, we consider Cases IV–VI when the node set is fixed. In such cases, we compare the proposed method SS-MCUSUM with the two existing methods PSCAN and CPD-KNN, since these two existing methods were designed for cases with a fixed node set. The estimated actual ARL$_0$ values of the three completing methods are presented in Table 3, along with their standard errors, when m_0 changes among $\{150, 200, 250, 300, 500\}$. From the table, we can have the following conclusions. (i) The SS-MCUSUM chart has a quite reliable IC performance in all Cases IV–VI when $m_0 \geq 200$ since its estimated actual ARL$_0$ values are within 10% of the nominal ARL$_0$ value of 200 in these cases. (ii) The PSCAN chart performs reasonably well in Cases IV–VI when $m_0 = 500$. (iii) In all cases considered, the estimated actual ARL$_0$ values of the CPD-KNN chart are well below the nominal ARL$_0$ value of 200. So, we can conclude that its IC performance is unreliable in these cases. So, from this example, it seems that SS-MCUSUM has a more reliable IC performance than that of PSCAN and CPD-KNN in all cases considered.

Table 3: Estimated actual ARL_0 values and their standard errors (in parentheses) of three network monitoring methods in Cases IV-VI when their nominal ARL_0 values are fixed at 200, and the number of IC networks m_0 changes among 150, 200, 250, 300, and 500.

Case	Methods	$m_0 = 150$	200	250	300	500
IV	SS-MCUSUM	174 (3.65)	180 (3.45)	189 (4.04)	190 (3.49)	192 (3.15)
	PSCAN	258 (5.25)	240 (4.59)	224 (3.31)	221 (3.33)	213 (2.45)
	CPD-KNN	126 (3.31)	132 (2.94)	131 (2.76)	133 (2.91)	128 (3.11)
V	SS-MCUSUM	186 (3.50)	193 (3.70)	194 (3.44)	198 (3.84)	197 (3.60)
	PSCAN	269 (8.24)	253 (7.83)	234 (6.78)	235 (5.30)	216 (3.98)
	CPD-KNN	128 (3.44)	128 (3.37)	130 (2.71)	131 (3.13)	126 (3.14)
VI	SS-MCUSUM	174 (4.22)	181 (3.24)	188 (3.88)	194 (3.93)	197 (3.15)
	PSCAN	256 (4.98)	241 (4.15)	233 (3.87)	225 (2.93)	214 (2.56)
	CPD-KNN	129 (2.76)	130 (2.80)	139 (2.43)	128 (3.42)	127 (3.00)

In the network monitoring method CPD-KNN, the parameter ρ which determines the maximum number of nearest neighbors in that method is chosen to be 5 in the above example, as recommended by Chen (2019). To study its IC performance when ρ is chosen to be other values, we consider Cases IV–VI when $m_0 = 300$ and ρ varies among $\{6, 7, 8, 9, 10\}$. The estimated actual ARL_0 values of CPD-KNN are presented in Table 4 in such cases. From the table, it can be seen that the estimated actual ARL_0 values of CPD-KNN are much less than the nominal ARL_0 value of 200 in all cases considered. Thus, the recommended value of ρ would indeed improve the IC performance of CPD-KNN, although the IC performance of CPD-KNN in such cases is still unsatisfactory.

Table 4: Estimated actual ARL_0 values and their standard errors (in parentheses) of CPD-KNN in Cases IV–VI when its nominal ARL_0 value is fixed at 200, $m_0 = 300$, and ρ changes among 6, 7, 8, 9, and 10.

Case	$\rho = 6$	7	8	9	10
IV	100 (2.68)	104 (2.81)	99 (2.99)	94 (2.98)	91 (2.81)
V	98 (2.67)	89 (2.67)	87 (2.69)	84 (2.64)	81 (2.52)
VI	94 (2.97)	98 (3.14)	93 (3.09)	89 (3.01)	85 (2.99)

Next, we consider Cases VII–XI when the fixed node set is assumed to follow the two-community structure and the networks are generated by the HDCSBM model. In such cases, we compare the proposed method SS-MCUSUM with the existing method Shewhart since this existing method was designed to monitor dynamic networks. The estimated actual ARL$_0$ values of the two completing methods when m_0 changes among $\{150, 200, 250, 300, 500\}$, along with their standard errors, are presented in Table B.2 of the supplementary file. From the table, we can have the following conclusions. (i) The SS-MCUSUM chart has a reliable IC performance in all Cases VII–XI when $m_0 \geq 200$ since its estimated actual ARL$_0$ values are within 10% of the nominal ARL$_0$ value of 200 in these cases. (ii) In all cases considered, the estimated actual ARL$_0$ values of the Shewhart chart is much smaller than 200. Thus, we can conclude that the IC performance of the SS-MCUSUM chart is reliable and the IC performance of the Shewhart chart is unreliable in this example.

Evaluation of the OC performance: Let us first consider Cases I–III when the node set of the dynamic network can change over time. In such cases, we compare the OC performance of the four methods SS-MCUSUM, EWMA-$d_{\max}$, EWMA-$\bar{d}$, and EWMA-SN when m_0 is fixed at 500 and a network structural change occurs at the beginning of online process monitoring. The value 500 is chosen for m_0 because all four methods have quite reliable IC performance in such a case according to Table 1. From the discussion in Subsection 3.2, the network structural change is determined by the four parameters $\{\overline{n}_v, p_1, p_2, \mu_e\}$, whose IC values are denoted as $\{\overline{n}_v^{(IC)}, p_1^{(IC)}, p_2^{(IC)}, \mu_e^{(IC)}\}$ and OC values are denoted as $\{\overline{n}_v^{(OC)}, p_1^{(OC)}, p_2^{(OC)}, \mu_e^{(OC)}\}$. Then, the following three types of network structural changes are considered in this study:

Network size changes: $\overline{n}_v^{(OC)} = \overline{n}_v^{(IC)} + 2.5\gamma$, where $\overline{n}_v^{(IC)} = 100$ in Cases I–III, $\gamma = \pm 1, \pm 2, \pm 3$ or ± 4, and the other three parameters $\{p_1, p_2, \mu_e\}$ do not change;

Communication changes: $(p_1^{(OC)}, p_2^{(OC)}, \mu_e^{(OC)}) = (p_1^{(IC)}, p_2^{(IC)}, \mu_e^{(IC)}) + (0.05, 0.05, 0.125)\gamma$, where $(p_1^{(IC)}, p_2^{(IC)}, \mu_e^{(IC)}) = (0.2, 0.4, 2)$ in Cases I and III, $(p_1^{(IC)}, p_2^{(IC)}, \mu_e^{(IC)}) = (0.6, 0.8, 4)$ in Case II, $\gamma = \pm 1, \pm 2, \pm 3$ or ± 4, and the parameter $\overline{n}_v$ does not change;

Evolution changes: $(\overline{n}_v^{(OC)}, p_1^{(OC)}, p_2^{(OC)}, \mu_e^{(OC)}) = (\overline{n}_v^{(IC)}, p_1^{(IC)}, p_2^{(IC)}, \mu_e^{(IC)}) + (2.5, 0.05, 0.05, 0.125)\gamma$, where $\overline{n}_v^{(IC)} = 100$ in Cases I–III, $(p_1^{(IC)}, p_2^{(IC)}, \mu_e^{(IC)}) = (0.2, 0.4, 2)$ in Cases I and III, $(p_1^{(IC)}, p_2^{(IC)}, \mu_e^{(IC)}) = (0.6, 0.8, 4)$ in Case II, and $\gamma = \pm 1, \pm 2, \pm 3$ or ± 4.

Network size changes are related to changes in the number of nodes only, communication changes represent changes in network connectivity and/or communications among nodes, and evolution changes happen when the dynamic network system under monitoring grows or shrinks over time. To make the comparison among different network monitoring methods fair, the control limits of the competing methods have all been adjusted properly so that their actual ARL_0 values equal to the nominal ARL_0 value of 200. In addition, the OC performance of different methods depends on the selection of their procedure parameters (e.g, the allowance constant k of the SS-MCUSUM chart (2)). Therefore, the OC performance of different methods may not be comparable when their procedure parameters are specified in advance (Qiu 2008). To avoid this limitation, Qiu (2008) suggested comparing the optimal OC performance of different methods by choosing the procedure parameters of each method such that its OC average run length (denoted as ARL_1) reached the minimum for detecting a given process structural change. This approach to compare the optimal OC performance of different methods has been adopted here.

In the same setups as those in Table 1, the optimal ARL_1 values of the four competing methods SS-MCUSUM, EWMA-$d_{\max}$, EWMA-$\bar{d}$, and EWMA-SN are shown in Figure 4. From the figure, we can have the following conclusions. (i) For detecting network size changes, SS-MCUSUM outperforms the methods EWMA-$d_{\max}$, EWMA-$\bar{d}$, and EWMA-SN in big margins in all cases considered. (ii) For detecting communication and evolution changes, the SS-MCUSUM chart outperforms the other three methods in most cases considered, while its OC performance is close to that of EWMA-SN in Cases I and III when γ is negative.

Next, we consider Cases IV–VI when the node set of the dynamic network is fixed over time. In such cases, we would like to compare the OC performance of the network monitoring methods SS-MCUSUM, PSCAN and CPD-KNN. From the results in Table 3, m_0 is fixed at 300 so that SS-MCUSUM has a reliable IC performance. Because the network size does not change over time in Cases IV–VI, only the communication changes are considered here. To make the comparison among different methods fair, their control limits have been adjusted properly so that their actual ARL_0 values all equal to the nominal ARL_0 value of 200. Other setups are the same as those in Table 3. The optimal ARL_1 values of the three methods are shown in Figure 5. From the figure, we can have the following conclusions. (i) SS-MCUSUM outperforms both PSCAN and CPD-KNN in quite large margins in all cases considered. (ii) Both PSCAN and CPD-KNN perform poorly in Cases IV–VI, and thus are insensitive to communication changes.

For detecting changes in community structure, we consider Cases VII–XI when the fixed node set is assumed to have the two-community structure. In such cases, the OC performance of the methods SS-MCUSUM and Shewhart are compared. From Table B.2 in the supplementary file, m_0

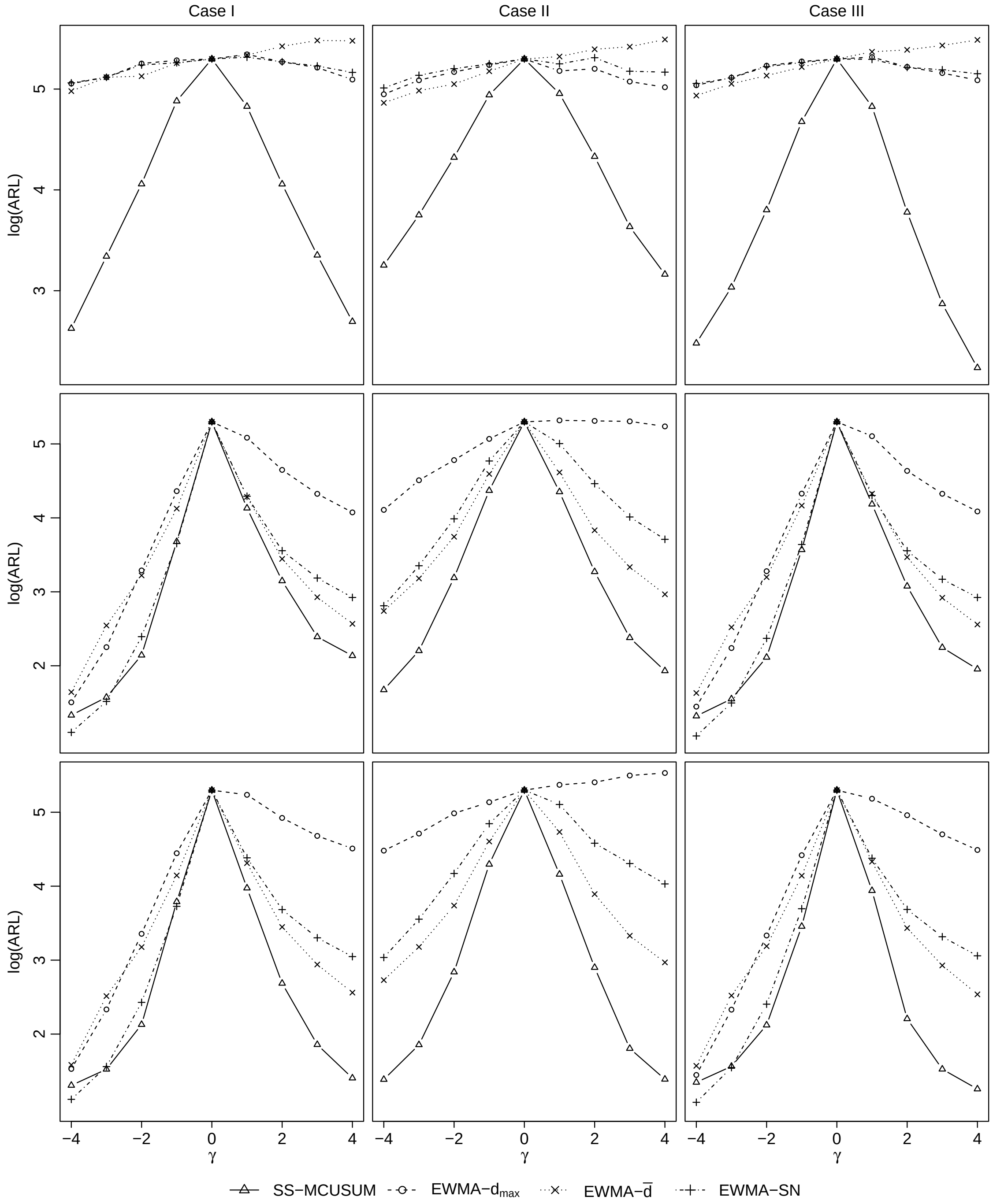

Figure 4: Optimal ARL_1 values of the four charts SS-MCUSUM, EWMA-$d_{\max}$, EWMA-$\bar{d}$ and EWMA-SN when their nominal ARL_0 values are fixed at 200, and $m_0 = 500$. Plots in the first row show the optimal ARL_1 values for detecting network size changes, plots in the second row show the optimal ARL_1 values for detecting communication changes, and plots in the third row show the optimal ARL_1 values for detecting evolution changes. In each plot, the y-axis denotes the optimal ARL_1 values in natural log scale.

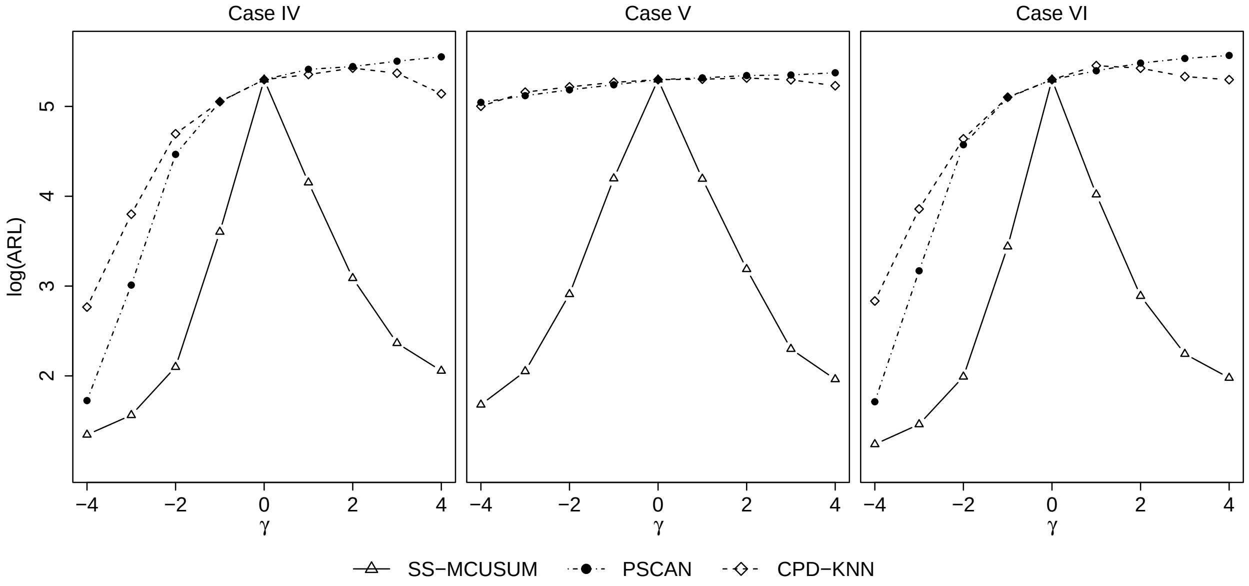

Figure 5: Optimal ARL_1 values of the three charts SS-MCUSUM, PSCAN and CPD-KNN for detecting communication changes in Cases IV–VI when their nominal ARL_0 values are fixed at 200 and $m_0 = 300$. In each plot, the y-axis denotes the optimal ARL_1 values in natural log scale.

is fixed at 300 so that SS-MCUSUM has a reliable IC performance. Different community structure changes considered in this example and the related optimal ARL_1 values of the two methods are presented in Table B.1 in the supplementary file. As in previous examples, the control limits of the two methods are adjusted properly so that their actual ARL_0 values are both 200 in all cases. From Table B.1, it can be seen that SS-MCUSUM outperforms Shewhart in all cases considered.

4 Case Studies

In this section, we demonstrate the proposed dynamic network monitoring method using three real-data examples. The first example is about an online community message network in which the node set could change over time, the second example is about the Enron email network where the node set does not change during a specific time period, and the third example is about a rating network of an online Bitcoin trading platform.

4.1 Message networks of an online community

For an online community at the University of California at Irvine, Panzarasa *et al.* (2009) provided a dataset about private messaging communications among community members. In the online community, members can search for other members and then initiate private messaging conversations based on the profile information. In this case, an edge of a dynamic network represents a private message from one member to another member, and members of the online community can join or leave the community any time. So, the node set can change over time. In this example, we use the observed data from July 6, 2004 to September 27, 2004. A total of 504 networks are constructed by aggregating event records in four-hour intervals. So, there are 6 networks every day, starting from the first one in 00:00-04:00 on July 6, 2004. Among the 504 networks, the 38th and 321st networks are excluded because their network sizes are much larger than the other networks. So, a total of 502 networks are actually used in this example.

Figure 6 shows the four features $\{n_{v,t}, \overline{d}_t, n_{c,t}, \overline{l}_t\}$ of the original data. Recall that these features represent the number of nodes, the average degree of all nodes, the number of connected components, and the average diameter of all connected components in a network. From the figure, the data from July 6 to September 10 (i.e., the first 400 observations), which are located on the left side of the vertical dotted lines in the plots, seem quite stable. Thus, they are used as the initial IC data. Figure 7 shows the first 42 original observations of the four features (i.e., the observed data during the first week), where the number attached to each observation denotes the time period of that observation within a specific day. From the figure, it can be seen that there is a quite obvious hour-of-day effect. For instance, for the first feature $n_{v,t}$ shown in the top panel of the figure, its values in the 5th and 6th time periods (i.e., during 16:00-24:00) are always quite large, and its values in the 2nd time periods (i.e., during 04:00-08:00) are always quite small. To eliminate the hour-of-day effect, we standardize all observations in the following way. First, for each time period, the sample mean and sample standard deviation are computed for each feature from the IC data. Second, for each feature, its observations in a given time period are standardized using the corresponding sample mean and sample standard deviation of that time period. Then, the standardized data are shown in Figure 8.

Next, the Durbin-Waston (DW) test is used to check the autocorrelation in the standardized IC data. The p-values of the DW test for the four features are computed to be 3.341×10^{-21}, 2.465×10^{-3}, 8.604×10^{-17}, and 2.538×10^{-3}, respectively, which imply significant autocorrelation in

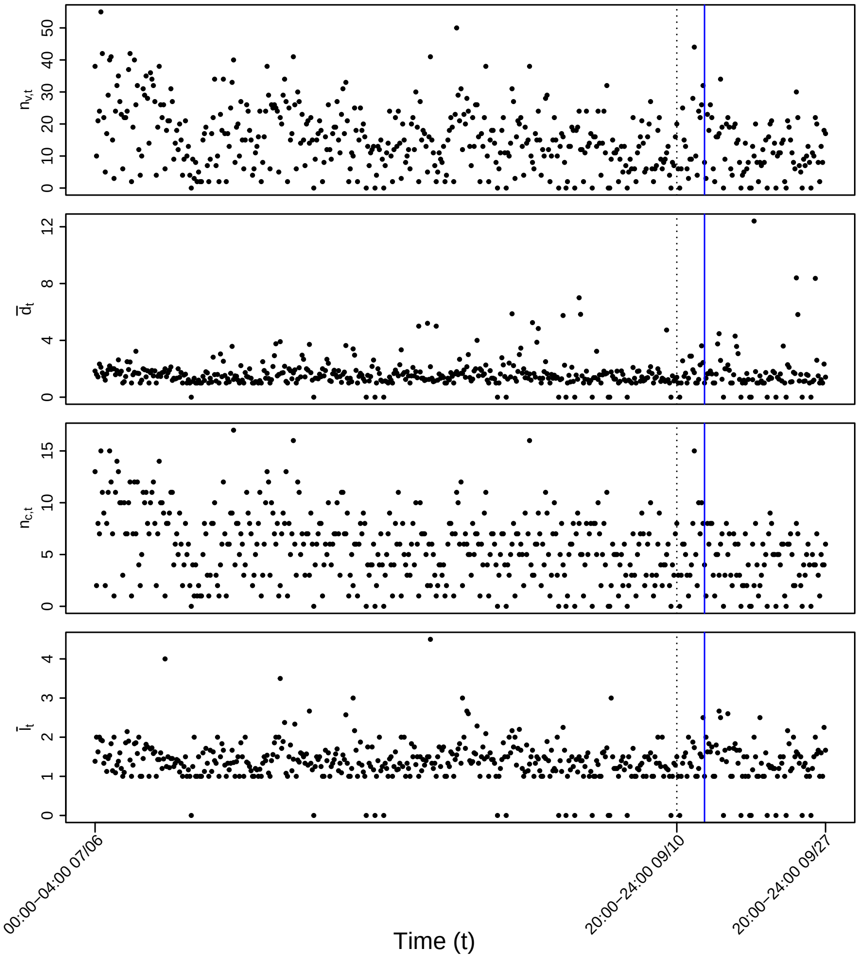

Figure 6: Original observations of the four features $\{n_{v,t}, \overline{d}_t, n_{c,t}, \overline{l}_t\}$ for the 502 networks in the online community example. The vertical dotted line in each panel separates the IC data from the data for online process monitoring, and the vertical solid line in each panel indicates the first signal time of the SS-MCUSUM chart.

the time series of all four features. To check the stationarity of the autocorrelation, the augmented Dickey-Fuller (ADF) test is used, and its p-values for the four features are all < 0.01, implying that the stationary assumption is valid for the four features in the IC data. To check the normality assumption for the IC data, the Shapiro-Wilk (SW) test is used for each feature, and its p-values for the four features are 1.159×10^{-8}, 2.429×10^{-18}, 1.551×10^{-6}, and 7.482×10^{-18}, respectively. Thus, the normality assumption is significantly violated for the four features. Based on these results, the proposed network monitoring method SS-MCUSUM is appropriate to use, since it can accommodate stationary serial correlation in the observed data and a nonparametric IC data distribution.

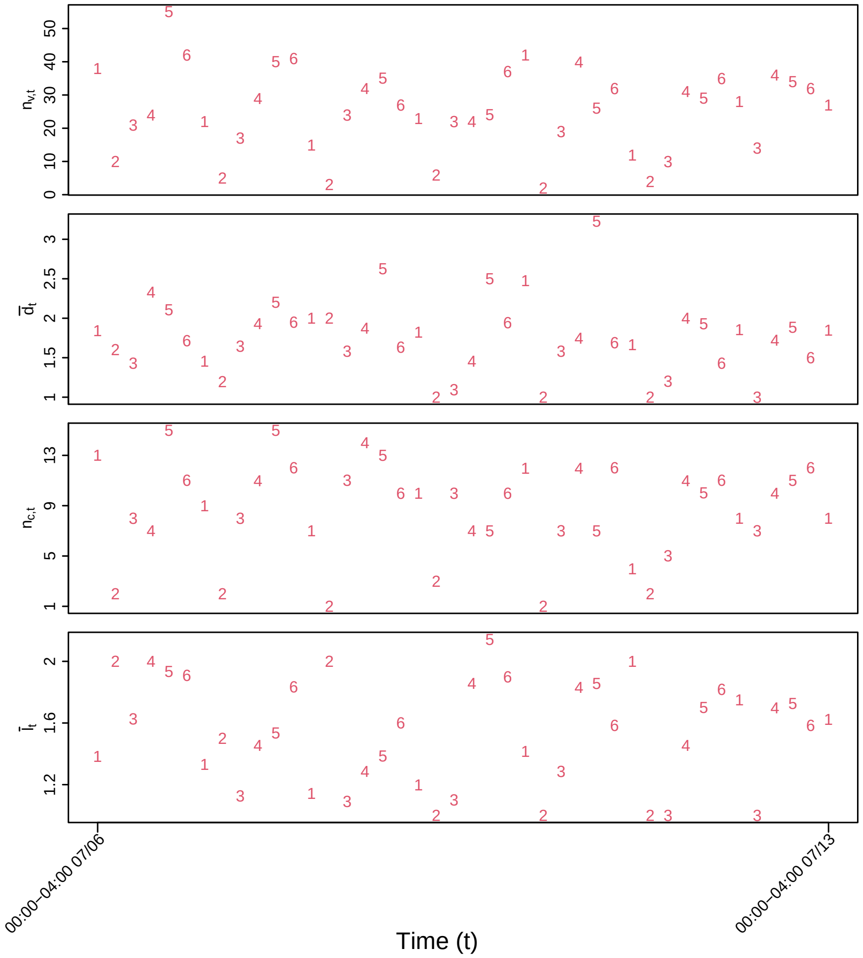

Figure 7: First 42 observations of the four features $\{n_{v,t}, \bar{d}_t, n_{c,t}, \bar{l}_t\}$ in the online community example. The number attached to each observation denotes the time period of that observation within a specific day.

In this example, because the node set changes over time, we consider applying the four network monitoring methods SS-MCUSUM, EWMA-$d_{\max}$, EWMA-$\bar{d}$, and EWMA-SN to the standardized data for online network monitoring. For the methods EWMA-$d_{\max}$, EWMA-$\bar{d}$, and EWMA-SN, they use the features $d_{\max,t}$, $\bar{d}_t$, and SN_t, respectively. The original observations of these three features are shown in Figure 9, and their standardized observations after the hour-of-day effect is excluded are shown in Figure 10. For their standardized IC data, the p-values of the DW test are 1.279×10^{-2}, 2.465×10^{-3}, and 5.814×10^{-3}, implying significant autocorrelation in the IC data. The Shapiro-Wilk (SW) test for checking the normality assumption gives the p-values $< 2.2 \times 10^{-16}$ for the three features, implying that the distributions of their IC data are significantly different from normal.

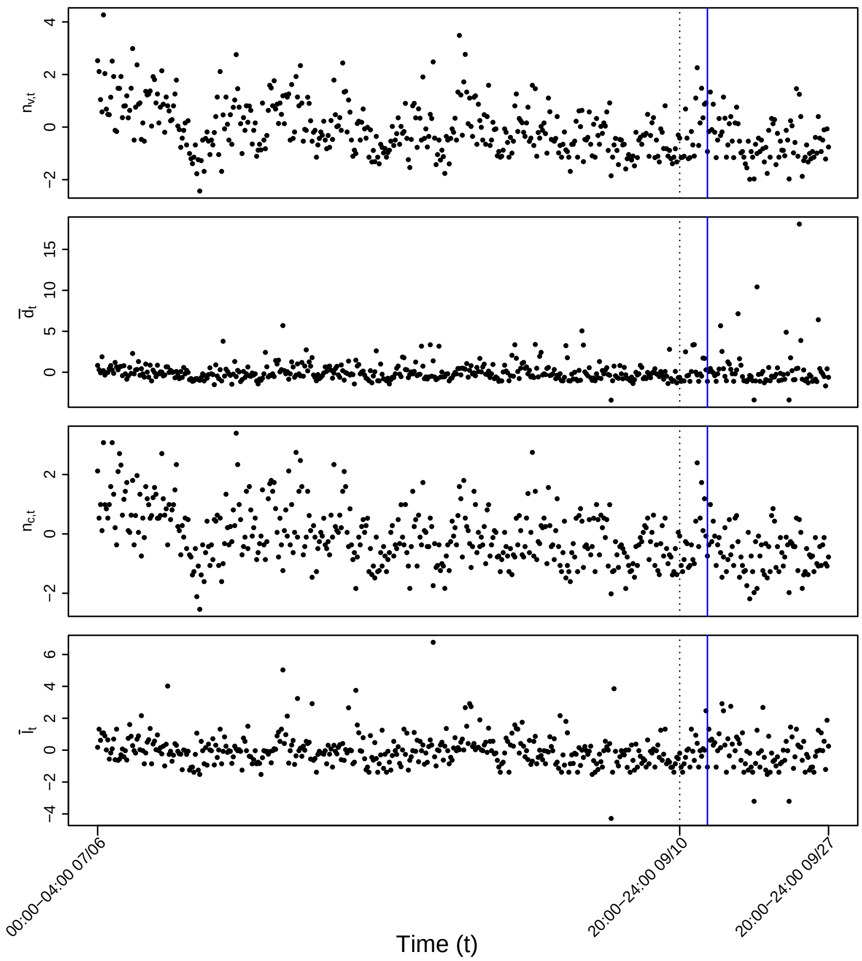

Figure 8: Standardized observations of the four features $\{n_{v,t}, \overline{d}_t, n_{c,t}, \overline{l}_t\}$ for the 502 networks in the online community example. The vertical dotted line in each panel separates the IC data from the data for online process monitoring, and the vertical solid line in each panel indicates the first signal time of the SS-MCUSUM chart.

For the four methods SS-MCUSUM, EWMA-$d_{\max}$, EWMA-$\bar{d}$, and EWMA-SN, their control limits are computed as in Section 3. The four charts are shown in Figure 11. From the figure, it can be seen that the charts SS-MCUCUM, EWMA-$d_{\max}$, EWMA-$\bar{d}$, and EWMA-SN give their first signals on September 14, September 17, September 15, and September 15, respectively. So, the signal by SS-MCUSUM is the earliest among the four methods in this example. To further interpret structural changes related to the signal given by SS-MCUSUM, we apply a modified local linear kernel estimator (cf., Brabanter *et al.* 2011) to obtain estimated mean curves of the standardized observations of the four network features. The estimated mean curves around the signal time of SS-MCUSUM are presented in Figure 12. From the figure, it can be seen that the signal is mainly

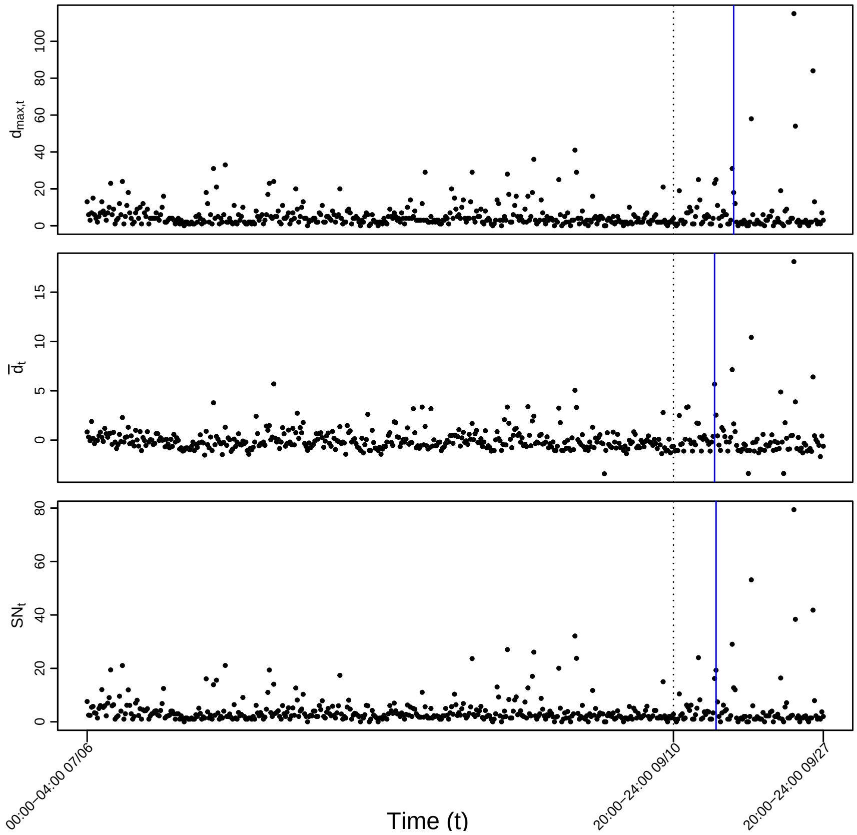

Figure 9: Original observations of the features $d_{\max,t}, \bar{d}_t$, and SN_t for the 502 networks in the online community example. The vertical dotted line in each panel separates the IC data from the data for online process monitoring, and the vertical solid line in each panel indicates the first signal time of the related control chart.

related to the increase of active community members (i.e., both $n_{v,t}$ and $n_{c,t}$ increase before the signal time) and the increase of local connected communities (i.e., $\bar{l}_t$ increases before the signal time).

4.2 Email networks of the Enron corporation

The Enron email corpus is a well-known network dataset reflecting communications in a real energy trading company. The Enron scandal, publicized in October 2001, eventually led to the bankruptcy of the company on December 21 of the same year. For the 184 Enron employees, the dataset contains their email communications over a time period from 1998 to 2002. In this example, we use the observed data from August 21, 2000 to November 24, 2001, with a total of 461 networks obtained by

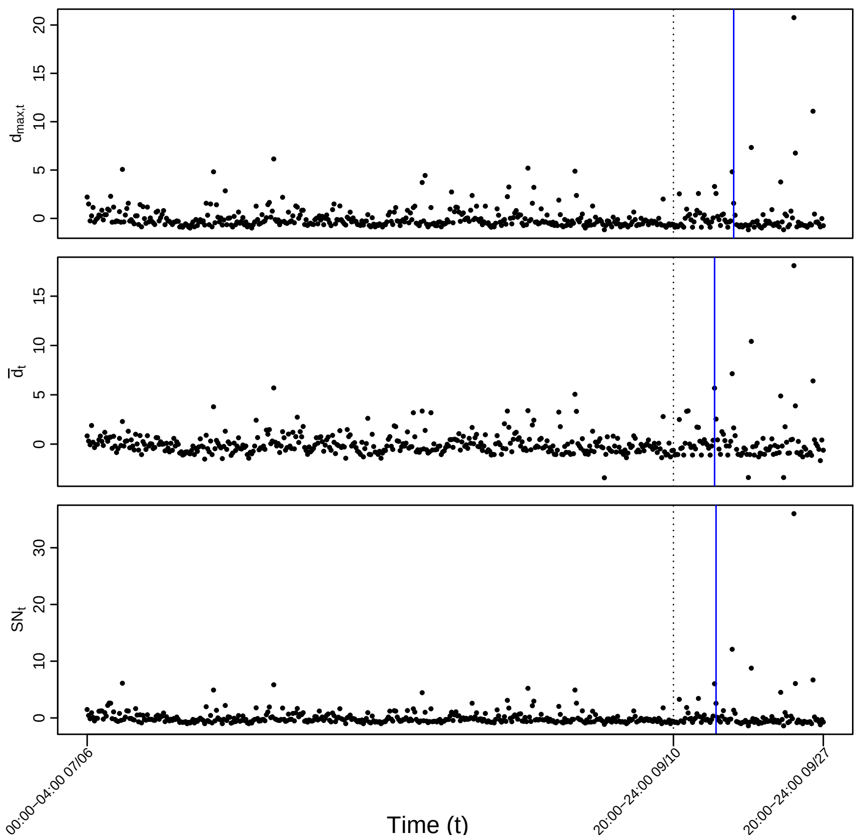

Figure 10: Standardized observations of the features $d_{\max,t}$, $\bar{d}_t$, and SN_t for the 502 networks in the online community example. The vertical dotted line in each panel separates the IC data from the data for online process monitoring, and the vertical solid line in each panel indicates the first signal time of the related control chart.

aggregating daily email records during that time period. For the 461 networks, the 275th network is excluded because it has an extremely large number of edges compared to the other networks, resulting in a total of 460 networks used in this example.

In this example, the number of nodes did not change in the time period under consideration. So, we only monitor the three features $\{\bar{d}_t, n_{c,t}, \bar{l}_t\}$ related to the average degree of all nodes, the number of connected components, and the average diameter of all connected components in a network. Figure 13 shows the observed data of these three features. From the figure, it can be seen that the data on the left-hand-side of the vertical dotted lines in the plots, which correspond to the data from August 21, 2000 (Monday) to May 27, 2001 (Sunday), are quite stable. Thus, this part of the observed data is used as the initial IC data. As in the example in Subsection 4.1, there is an obvious day-of-week pattern in the observed data. After that pattern is deleted, the standardized

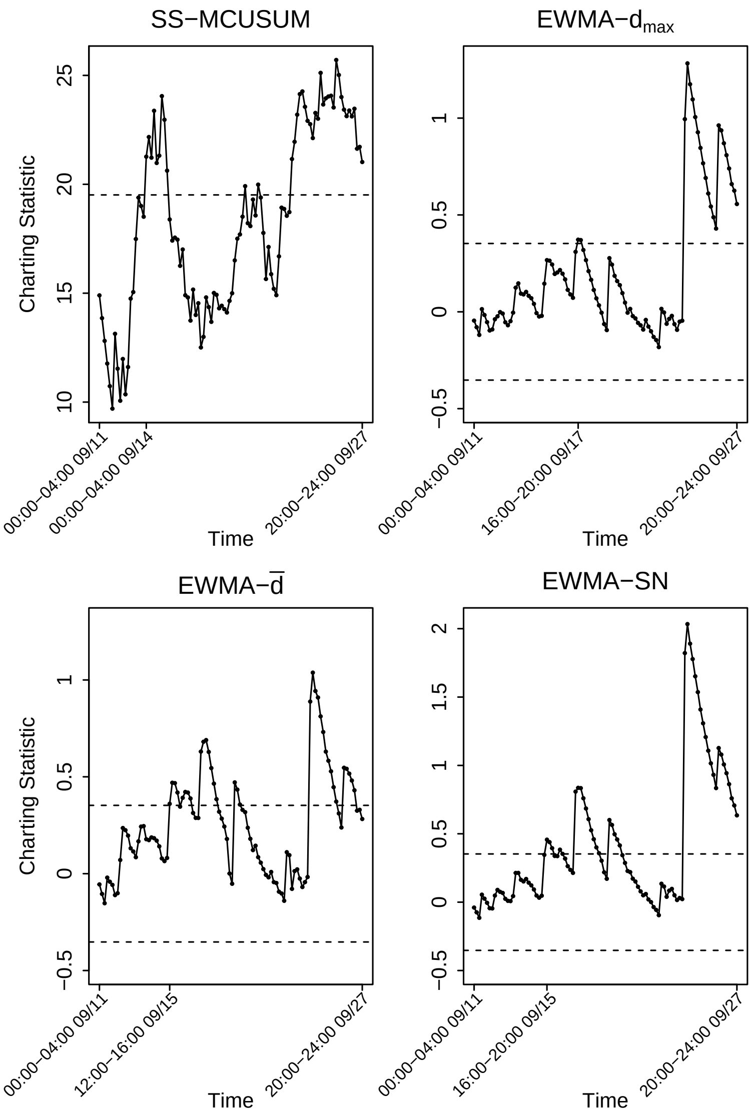

Figure 11: Control charts SS-MCUCUM, EWMA-$d_{\max}$, EWMA-$\bar{d}$, and EWMA-SN for monitoring the dynamic messaging network of the online community at the University of California at Irvine. The horizontal dashed line in each panel denotes the control limit of the related control chart.

data are shown in Figure 14.

To check the significance of autocorrelation in the standardized IC data, the DW test is used and its p-values for the three features are 3.823×10^{-4}, 6.147×10^{-7}, and 1.355×10^{-7}, respectively. So, there is significant autocorrelation in the time series of the three features. In addition, the p-values of the ADF test for the three features are all < 0.01, implying that the stationary assumption is valid for the three features in the IC data. The SW test is then used to check the normality assumption in

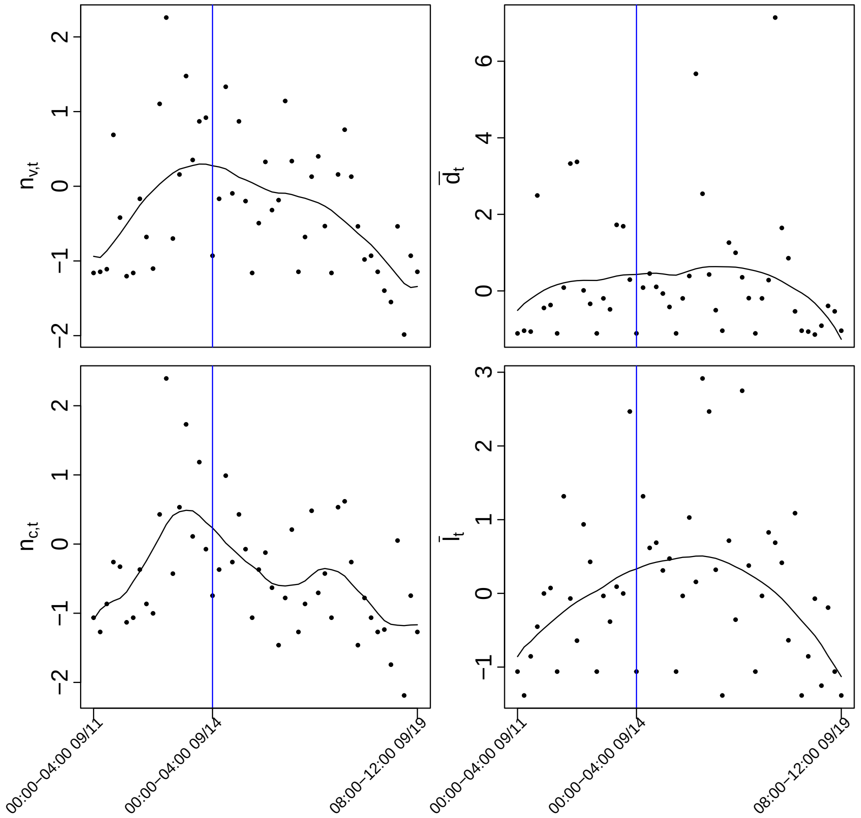

Figure 12: Estimated mean curves of the standardized observations of the four features $\{n_{v,t}, \bar{d}_t, n_{c,t}, \bar{l}_t\}$ around the signal time (vertical solid line in each plot) of SS-MCUSUM in the online community example.

the standardized IC data for each feature, and its p-values for the three features are 3.632×10^{-11}, 0.012, and 0.014, respectively. Thus, the normality assumption is significantly violated for the three features in this example, and the multivariate nonparametric chart SS-MCUSUM should be appropriate to use here.

We then apply the three monitoring methods SS-MCUSUM, PSCAN and CPD-KNN to this dataset to monitor the networks starting from May 28, 2001 (i.e., the first observation after the vertical dotted lines in Figure 14). To use the method CPD-KNN, the similarity measure between two networks cannot be calculated if there are no edges in either network. In such cases, the network with no edges is always merged into the next network with edges in order to use that method properly. The control limits of the three charts are computed as in Section 3. Then, the three charts are shown in Figure 15. From the figure, it can be seen that SS-MCUSUM, PSCAN, and CPD-KNN give their first signals on June 5, 2001, August 24, 2001,

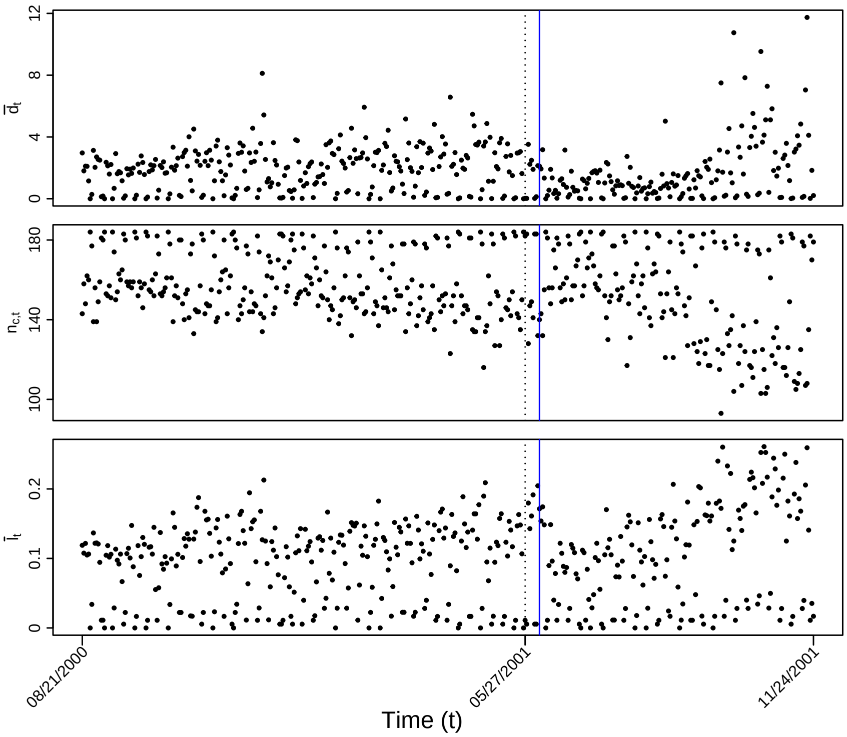

Figure 13: Original observations of the three features $\{\bar{d}_t, n_{c,t}, \bar{l}_t\}$ for the 460 networks in the Enron email example. The vertical dotted line in each panel separates the IC data from the data for online process monitoring, and the vertical solid line in each panel indicates the first signal time of the SS-MCUSUM chart.

and June 13, 2001, respectively. So, the SS-MCUSUM chart gives the earliest first signal among all three methods in this example. As in the previous example, we use a modified local linear kernel smoothing procedure to obtain estimated mean curves of the standardized observations of the three features. The estimated mean curves around the signal time of SS-MCUCUM are presented in Figure 16. From the figure, it can be seen that the signal is related to decreasing email communications (i.e., $\bar{d}_t$ decreases and $n_{c,t}$ increases before the signal time) and decreasing local connected communities (i.e., $\bar{l}_t$ decreases before the signal time) among Enron employees. From the Enron Timeline (https://www.agsm.edu.au/bobm/teaching/BE/Enron/timeline.html), the Federal Energy Regulatory Commission finally instituted price caps across the western states, and the California energy crisis ended in June 2021, which might be related to the reduced email communications among Enron employees.

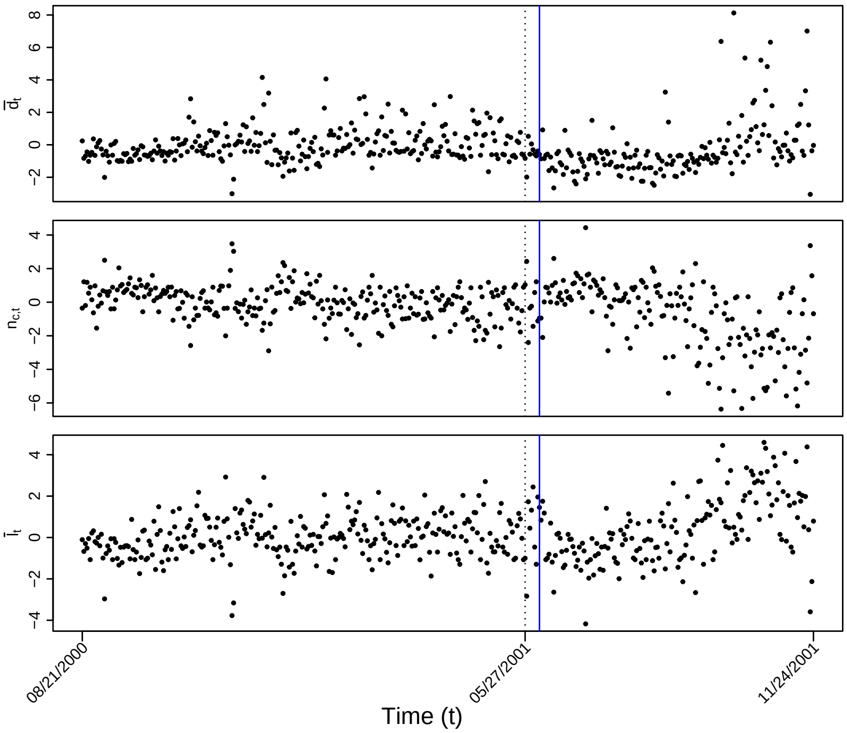

Figure 14: Standardized observations of the three features $\{\bar{d}_t, n_{c,t}, \bar{l}_t\}$ for the 460 networks in the Enron email example. The vertical dotted line in each panel separates the IC data from the data for online process monitoring, and the vertical solid line in each panel indicates the first signal time of the SS-MCUSUM chart.

4.3 Rating network of an online trading platform

Bitcoin Alpha is an online trading platform where users can trade Bitcoin with others. Because Bitcoin users are anonymous, users of the platform often rate other users during or after transactions. In this example, an edge of a dynamic network represents a rating action from one user to another, and we use the observed data from April 10, 2012 to April 25, 2013. For the 1,618 users, a total of 381 networks are constructed by aggregating daily rating records during that time period. For the 381 networks, the 42nd, 44th, 168th, and 183rd networks are removed because they have extremely large numbers of edges compared to other networks, resulting in a total of 377 networks used in our analysis.

In this example, the number of nodes is fixed during the time period considered. So, the three features $\{\bar{d}_t, n_{c,t}, \bar{l}_t\}$ are monitored. After the day-of-week pattern is removed as in the previous two examples, the standardized values of the three features are shown in Figure 17. From the figure, the

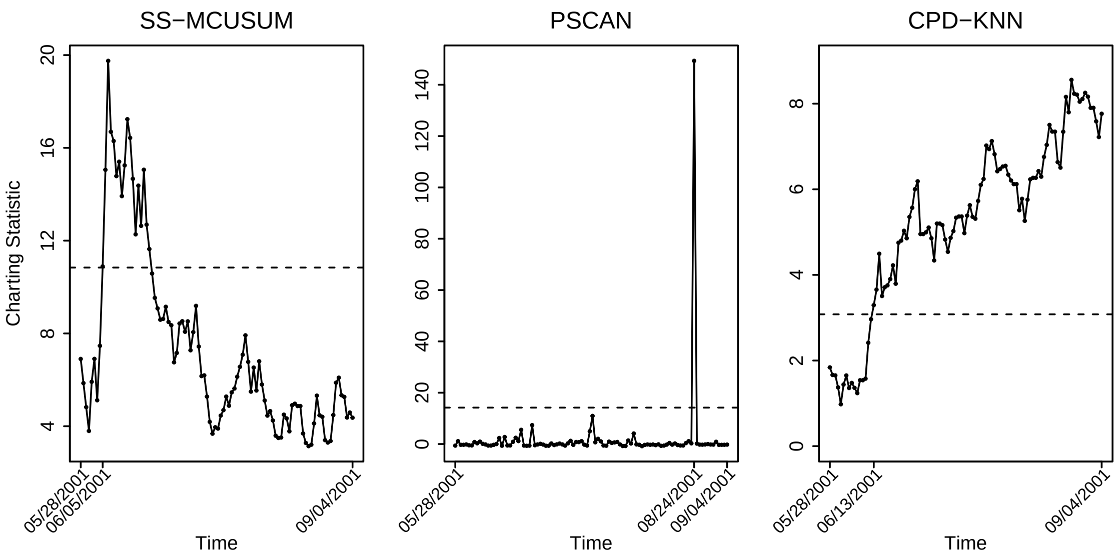

Figure 15: Control charts SS-MCUCUM, PSCAN and CPD-KNN for monitoring the email networks of Enron employees. The horizontal dashed line in each panel denotes the control limit of the related control chart.

data from April 10, 2012 to February 7, 2013 (i.e., the first 300 observations located on the left side of the vertical dotted lines) are quite stable. So, they are used as the initial IC data. As in the previous two examples, the DW, ADF and SW tests confirm that the initial IC data contain significant stationary autocorrelation and the normality assumption would be violated. Thus, the multivariate nonparametric chart SS-MCUSUM should be appropriate to use in this example.

We then apply the three monitoring methods SS-MCUSUM, PSCAN and CPD-KNN to this dataset to monitor the networks starting from February 8, 2013 (i.e., the first observation after the vertical dotted lines in Figure 17). The control limits of the three charts are computed as in Section 3. Plots of the three charts are shown in Figure 18. From the figure, it can be seen that SS-MCUSUM and PSCAN give their first signals on February 20, 2013 and March 25, 2013, respectively, and no signal is given by CPD-KNN. The SS-MCUSUM chart thus gives the earliest first signal among all three methods in this example. As in the previous example, a modified local linear kernel smoothing procedure is used to estimate mean curves of the standardized data, and the estimated mean curves around the signal time of SS-MCUSUM are shown in Figure 19. From the figure, it can be seen that the signal is related to the increasing rating actions (i.e., $\bar{d}_t$ increases and $n_{c,t}$ decreases before the signal time) and the increasing local connected communities (i.e., $\bar{l}_t$ increases around the signal time). In February 2013, the Bitcoin-based payment processor Coinbase

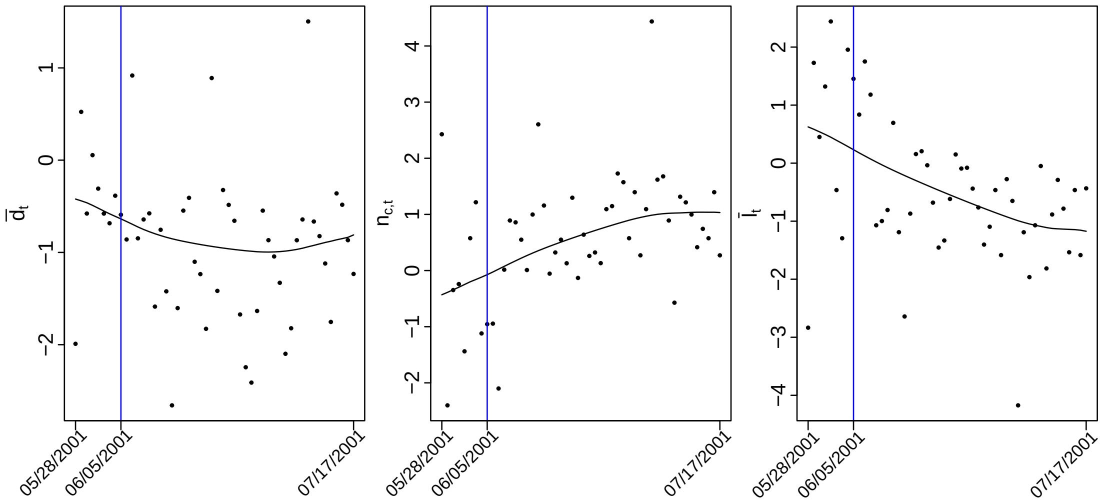

Figure 16: Estimated mean curves of the standardized observations of the three features $\{\bar{d}_t, n_{c,t}, \bar{l}_t\}$ around the signal time of SS-MCUCUM in the Enron email example. The vertical solid line in each plot indicates the first signal time given by the SS-MCUSUM chart.

reported selling $1 million worth of Bitcoins in a single month, and the American digital library (The Internet Archive) announced that it was ready to accept donations as Bitcoins and that it intended to give employees the option to receive portions of their salaries in Bitcoin. These events might stimulate transactions on Bitcoin Alpha, and increasing the rating actions on the platform.

5 Concluding Remarks

We have described a general framework for online monitoring of dynamic networks. By the proposed network monitoring approach, four network features are first extracted from each observed network, which represent the number of nodes, the average degree of all nodes, the number of connected components, and the average diameter of all connected components in a network. Then, a multivariate nonparametric control chart is applied to the extracted features for detecting network structural changes. This method is flexible in the sense that it can accommodate nonparametric data distribution and serial data correlation, and sensitive to various network structural changes. Both simulation studies and real network data examples show that the proposed method provides a reliable and effective tool for dynamic network monitoring. However, the current version of the proposed method still has several limitations. First, the number of connected components and

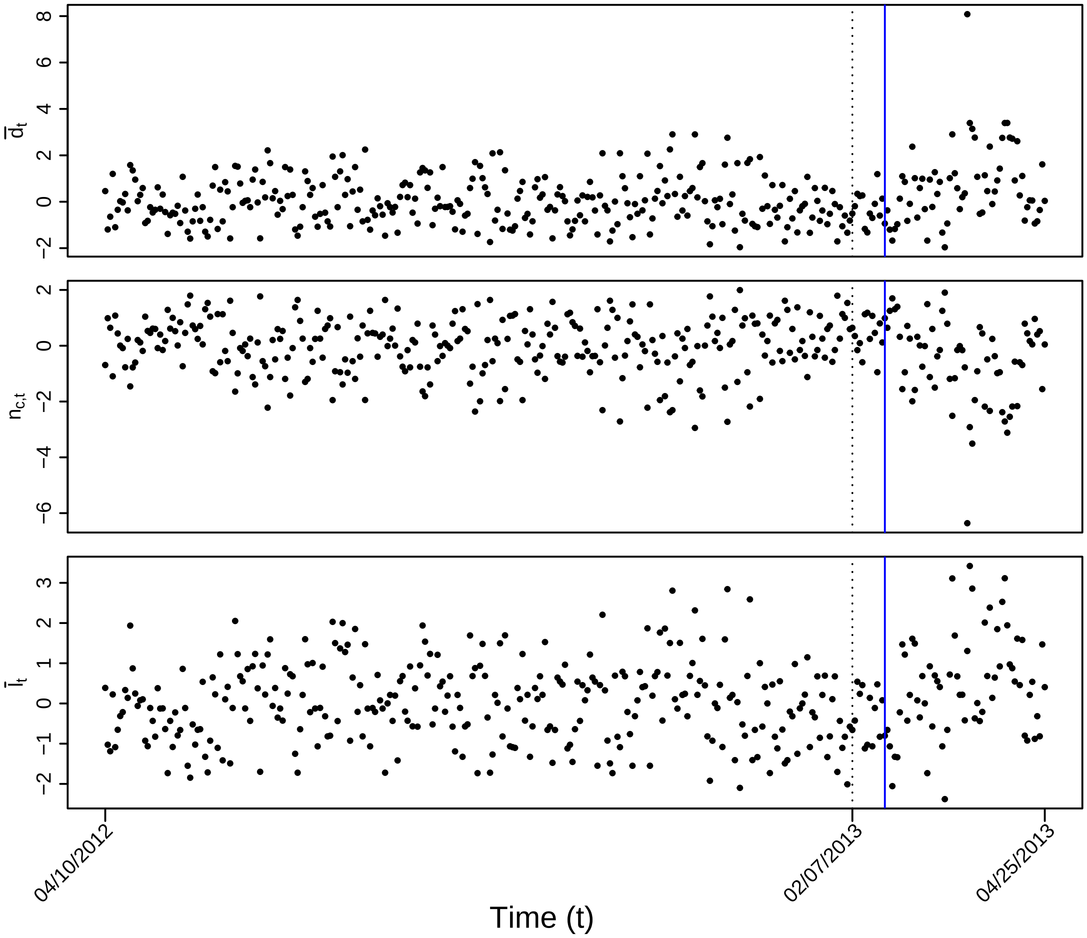

Figure 17: Standardized observations of the three features $\{\bar{d}_t, n_{c,t}, \bar{l}_t\}$ for the 377 networks in the Bitcoin Alpha example. The vertical dotted line in each panel separates the IC data from the data for online process monitoring, and the vertical solid line in each panel indicates the first signal time of the SS-MCUSUM chart.

the average diameter of all connected components are chosen by considering the network structure of connected components only. They can be quite stable when the observed networks are highly connected (e.g., each network has only one connected component). In such cases, these two features may not be very sensitive to network structural changes. There are two possible ways to overcome this limitation. One is to construct sparser networks by aggregating event records within smaller time intervals, and the other is to use a weighted average diameter of all connected components in a network as an alternative feature. Second, some specific network changes (e.g., membership changes in the example discussed in Subsection 4.1) can be missed by considering undirected networks with no attributes. As a future research topic, the proposed method might be generalized to handle dynamic networks with attributes and directed edges (Savage *et al.* 2014). Third, intrinsic

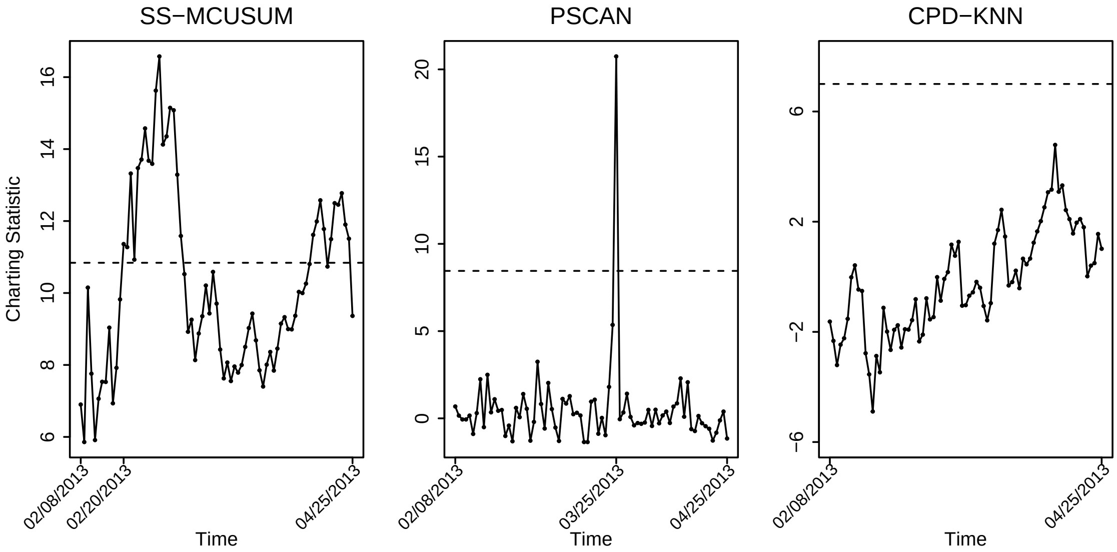

Figure 18: Control charts of SS-MCUCUM, PSCAN and CPD-KNN for monitoring the rating networks of users on Bitcoin Alpha. The horizontal dashed line in each panel denotes the control limit of the related control chart.

hierarchical structure is common in a network system. To monitor such dynamic network sequences, the monitoring method should make use of the intrinsic hierarchical structure in order to detect changes in the hierarchical structure effectively, which has not been discussed in this paper. Last but not the least, after a structure change is detected by the proposed monitoring procedure, a post-signal network analysis should be performed to figure out when and where the detected change occurs in the dynamic network system. All these topics require much future research effort.

Acknowledgements: The authors thank the editor and two referees for many insightful comments and suggestions, which improved the quality of the paper greatly.

References

Ahn, J.-W., Plaisant, C., and Shneiderman, B. (2013), "A task taxonomy for network evolution analysis," *IEEE Transactions on Visualization and Computer Graphics*, **20**, 365–376.

Apley D.W., and Tsung, F. (2002), "The autoregressive T^2 chart for monitoring univariate autocorrelated processes," *Journal of Quality Technology*, **34**, 80–96.

Borror, C. M., Montgomery, D. C., and Runger, G. C. (1999), "Robustness of the ewma control chart to non-normality," *Journal of Quality Technology*, **31**, 309–316.

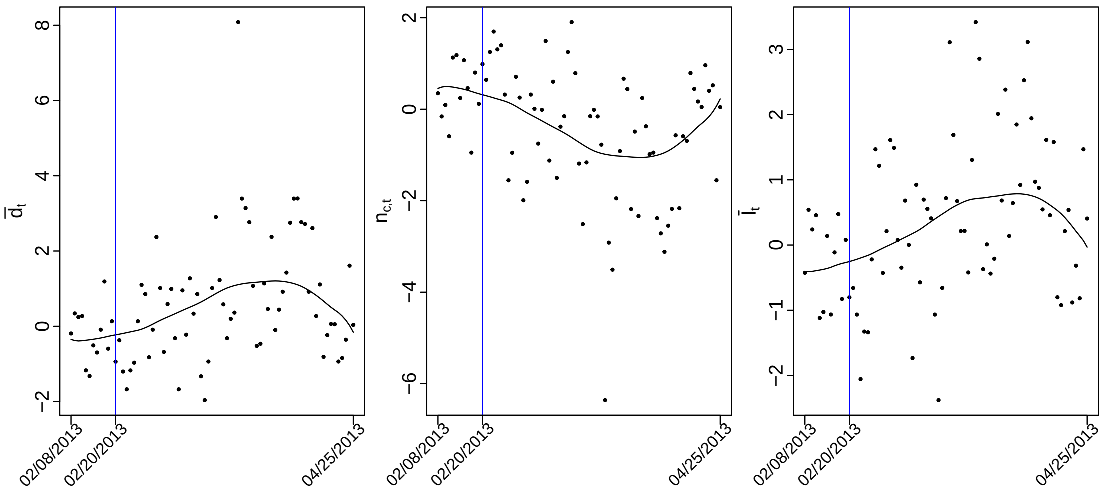

Figure 19: Estimated mean curves of the standardized observations of the three features $\{\bar{d}_t, n_{c,t}, \bar{l}_t\}$ around the signal time of SS-MCUSUM in the Bitcoin Alpha example. The vertical solid line in each panel indicates the first signal time of the SS-MCUSUM chart.

Capizzi, G., and Masarotto, G. (2008), "Practical design of generalized likelihood ratio control charts for autocorrelated data," *Technometrics*, **50**, 357–370.

Chatterjee, S., and Qiu, P. (2009), "Distribution-free cumulative sum control charts using bootstrap-based control limits," *Annals of Applied Statistics*, **3**, 349–369.

Chen, H. (2019), "Sequential change-point detection based on nearest neighbors," *The Annals of Statistics*, **47**, 1381–1407.

Chung, F. and Lu, L. (2002), "Connected components in random graphs with given expected degree sequences," *Annals of Combinatorics*, **6**, 125–145.

De Brabanter, K., De Brabanter, J., Suykens, J. A., and De Moor, B. (2011), "Kernel Regression in the Presence of Correlated Errors," *Journal of Machine Learning Research*, **12**, 1955–1976.

Dong, H., Chen, N., and Wang, K. (2020), "Modeling and change detection for count-weighted multilayer networks," *Technometrics*, **62**, 184–195.

Ebrahimi, S., Reisi-Gahrooei, M., Paynabar, K., and Mankad, S. (2021), "Monitoring sparse and attributed networks with online hurdle models," *IISE Transactions*, **54**, 91–104.

Flossdorf, J. and Jentsch, C. (2021), "Change detection in dynamic networks using network characteristics," *IEEE Transactions on Signal and Information Processing over Networks*, **7**, 451–464.

Friedkin, N. E. (1981), "The development of structure in random networks: an analysis of the effects of increasing network density on five measures of structure," *Social Networks*, **3**, 41–52.

Grogger, J. T. and Carson, R. T. (1991), "Models for truncated counts," *Journal of Applied Econometrics*, **6**, 225–238.

Guzman, J. D., Deckro, R. F., Robbins, M. J., Morris, J. F., and Ballester, N. A. (2014), "An analytical comparison of social network measures," *IEEE Transactions on Computational Social Systems*, **1**, 35–45.

Hanneke, S., Fu, W., and Xing, E. P. (2010), "Discrete temporal models of social networks," *Electronic Journal of Statistics*, **4**, 585–605.

Harrison C. W. (1992), *Identity and Control: A Structural Theory of Social Action,* Princeton, N.J.: Princeton University Press.

Hawkins, D. M. (1987), "Self-starting cusums for location and scale," *The Statistician*, **36**, 299–315.

Henze, N. (1988), "A multivariate two-sample test based on the number of nearest neighbor type coincidences," *The Annals of Statistics*, **16**, 772–783.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002), "Latent space approaches to social network analysis," *Journal of the American Statistical Association*, **97**, 1090–1098.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983), "Stochastic blockmodels: First steps," *Social Networks*, **5**, 109–137.

Horng Shiau, J.-J. and Ya-Chen, H. (2005), "Robustness of the ewma control chart to nonnormality for autocorrelated processes," *Quality Technology & Quantitative Management*, **2**, 125–146.

Hosseini, S. S. and Noorossana, R. (2018), "Performance evaluation of ewma and cusum control charts to detect anomalies in social networks using average and standard deviation of degree measures," *Quality and Reliability Engineering International*, **34**, 477–500.

Hunter, D. R., Goodreau, S. M., and Handcock, M. S. (2008), "Goodness of fit of social network models," *Journal of the American Statistical Association*, **103**, 248–258.

Jeske, D. R., Stevens, N. T., Tartakovsky, A. G., and Wilson, J. D. (2018), "Statistical methods for network surveillance," *Applied Stochastic Models in Business and Industry*, **34**, 425–445.

Jones-Farmer, L.A., Jordan, V., and Champ, C.W. (2009), "Distribution-free phase I control charts for subgroup location," *Journal of Quality Technology*, **41**, 304–317.

Karrer, B. and Newman, M. E. (2011), "Stochastic blockmodels and community structure in networks," *Physical Review E*, **83**, 016107.

Kim, B., Lee, K. H., Xue, L., and Niu, X. (2018), "A review of dynamic network models with latent variables," *Statistics Surveys*, **12**, 105.

Lancichinetti, A., Fortunato, S., and Radicchi, F. (2008), "Benchmark graphs for testing community detection algorithms," *Physical Review E*, **78**, 046110.

Lee, W., McCormick, T. H., Neil, J., Sodja, C., and Cui, Y. (2022), "Anomaly detection in large-scale networks with latent space models," *Technometrics*, **64**, 241–252.

Leskovec, J., Kleinberg, J., and Faloutsos, C. (2007), "Graph evolution: Densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, **1**, 2–es.

Leskovec, J. and Krevl, A. (2014), *Snap datasets: Stanford large network dataset collection.*

Liu, Y., Gu, Z., and Liu, J. (2021), "Uncovering transmission patterns of covid-19 outbreaks: A region-wide comprehensive retrospective study in hong kong," *EClinicalMedicine*, **36**, 100929.

Lotker, Z. (2021), *Analyzing Narratives in Social Networks*, New York: Springer, Cham.

McCulloh, I. and Carley, K. M. (2011), "Detecting change in longitudinal social networks," *Journal of Social Structure*, **12**, 1-37.

Neil, J., Hash, C., Brugh, A., Fisk, M., and Storlie, C. B. (2013), "Scan statistics for the online detection of locally anomalous subgraphs," *Technometrics*, **55**, 403–414.

Newman, M. (2018), *Networks: An Introduction (2nd edition)*, Oxford, UK: Oxford university press.

Nguyen, N. P., Dinh, T. N., Shen, Y., and Thai, M. T. (2014), "Dynamic social community detection and its applications," *PloS One*, **9**, e91431.

Ofori-Boateng, D., Gel, Y. R., and Cribben, I. (2021), "Nonparametric anomaly detection on time series of graphs," *Journal of Computational and Graphical Statistics*, **30**, 756–767.

Panzarasa, P., Opsahl, T., and Carley, K. M. (2009), "Patterns and dynamics of users' behavior and interaction: Network analysis of an online community," *Journal of the American Society for Information Science and Technology*, **60**, 911–932.

Peel, L. and Clauset, A. (2015), "Detecting change points in the large-scale structure of evolving networks," In *Twenty-Ninth AAAI Conference on Artificial Intelligence.*

Perry, M. B. (2020), "An ewma control chart for categorical processes with applications to social network monitoring," *Journal of Quality Technology*, **52**, 182–197.

Priebe, C. E., Conroy, J. M., Marchette, D. J., and Park, Y. (2005), "Scan statistics on enron graphs," *Computational & Mathematical Organization Theory*, **11**, 229–247.

Qiu, P. (2008), "Distribution-free multivariate process control based on log-linear modeling," *IIE Transactions*, **40**, 664–677.

Qiu, P. (2014), *Introduction to Statistical Process Control*, Boca Raton, FL: Chapman Hall/CRC.

Qiu, P. (2018), "Some perspectives on nonparametric statistical process control," *Journal of Quality Technology*, **50**, 49–65.

Qiu, P., and Xiang, D. (2014), "Univariate dynamic screening system: an approach for identifying individuals with irregular longitudinal behavior," *Technometrics*, **56**, 248–260.

Qiu, P. and Xie, X. (2022), "Transparent sequential learning for statistical process control of serially correlated data," *Technometrics*, **64**, 487–501.

Ranshous, S., Shen, S., Koutra, D., Harenberg, S., Faloutsos, C., and Samatova, N. F. (2015), "Anomaly detection in dynamic networks: a survey," *Wiley Interdisciplinary Reviews: Computational Statistics*, **7**, 223–247.

Rodríguez-Núnez, E. and García-Palomares, J. C. (2014), "Measuring the vulnerability of public transport networks," *Journal of Transport Geography*, **35**, 50–63.

Rossetti, G. and Cazabet, R. (2018), "Community discovery in dynamic networks: a survey," *ACM Computing Surveys (CSUR)*, **51**, 1–37.

Salmasnia, A., Mohabbati, M., and Namdar, M. (2020), "Change point detection in social networks using a multivariate exponentially weighted moving average chart," *Journal of Information Science*, **46**, 790–809.

Savage, D., Zhang, X., Yu, X., Chou, P., and Wang, Q. (2014), "Anomaly detection in online social networks," *Social Networks*, **39**, 62–70.

Segarra, S. and Ribeiro, A. (2015), "Stability and continuity of centrality measures in weighted graphs," *IEEE Transactions on Signal Processing*, **64**, 543–555.

Sewell, D. K. and Chen, Y. (2015), "Latent space models for dynamic networks," *Journal of the American Statistical Association*, **110**, 1646–1657.

Schilling, M.F. (1986), "Multivariate two-sample tests based on nearest neighbors," *Journal of the American Statistical Association*, **81**, 799–806.

Sizemore, A. E. and Bassett, D. S. (2018), "Dynamic graph metrics: Tutorial, toolbox, and tale," *NeuroImage*, **180**, 417–427.

Snijders, T. A. (2005), "Models for longitudinal network data," *Models and Methods in Social Network Analysis*, **1**, 215–247.

Sparks, R. and Wilson, J. D. (2019), "Monitoring communication outbreaks among an unknown team of actors in dynamic networks," *Journal of Quality Technology*, **51**, 353–374.

Turcotte, M. J., Kent, A. D., and Hash, C. (2019), "Unified host and network data set," In *Data Science for Cyber-Security*, pages 1–22. World Scientific.

von Landesberger, T., Gorner, M., and Schreck, T. (2009), "Visual analysis of graphs with multiple connected components," In *2009 IEEE Symposium on Visual Analytics Science and Technology*, pages 155–162. IEEE.

Wilson, J. D., Stevens, N. T., and Woodall, W. H. (2019), "Modeling and detecting change in temporal networks via the degree corrected stochastic block model," *Quality and Reliability Engineering International*, **35**, 1363–1378.

Woodall, W. H., Zhao, M. J., Paynabar, K., Sparks, R., and Wilson, J. D. (2017), "An overview and perspective on social network monitoring," *IISE Transactions*, **49**, 354–365.

Xu, K. S. and Hero, A. O. (2014), “Dynamic stochastic blockmodels for time-evolving social networks,” *IEEE Journal of Selected Topics in Signal Processing*, **8**, 552–562.

Xue, L. and Qiu, P. (2021), “A nonparametric cusum chart for monitoring multivariate serially correlated processes,” *Journal of Quality Technology*, **53**, 396–409.

Yu, L., Woodall, W. H., and Tsui, K.-L. (2018), “Detecting node propensity changes in the dynamic degree corrected stochastic block model,” *Social Networks*, **54**, 209–227.

Yu, L., Zwetsloot, I. M., Stevens, N. T., Wilson, J. D., and Tsui, K. L. (2022), “Monitoring dynamic networks: A simulation-based strategy for comparing monitoring methods and a comparative study,” *Quality and Reliability Engineering International*, **38**, 1226–1250.

Zhang, X., Moore, C., and Newman, M. E. (2017), “Random graph models for dynamic networks,” *The European Physical Journal B*, **90**, 1–14.

Zhao, Y., Levina, E., and Zhu, J. (2012), “Consistency of community detection in networks under degree-corrected stochastic block models, *The Annals of Statistics*, **40**, 2266–2292.

Zwetsloot, I. M. and Woodall, W. H. (2021), “A review of some sampling and aggregation strategies for basic statistical process monitoring,” *Journal of Quality Technology*, **53**, 1–16.