Computers & Industrial Engineering 145 (2020) 106507

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Performance evaluation method for network monitoring based on separable temporal exponential random graph models with application to the study of autocorrelation effects

Panpan Zhou[a], Dennis K.J. Lin[b], Xiaoyue Niu[c], Zhen He[d,*]

[a] School of Management Science and Engineering, Nanjing University of Finance and Economics, Nanjing, China
[b] Department of Statistics, Purdue University, West Lafayette, IN, USA
[c] Department of Statistics, Pennsylvania State University, PA, USA
[d] College of Management and Economics, Tianjin University, Tianjin, China

ARTICLE INFO

Keywords:
Network monitoring
Autocorrelation
Control charts
Performance evaluation
Temporal exponential random graph models

ABSTRACT

With the quick development of sensors and information technologies enabling Industry 3.5, network data, which represent the interactions among related entities, have extensively emerged in manufacturing and service industries. Statistical process monitoring serves as an efficient tool for supporting accurate and timely decision-making in Industry 3.5. Applying statistical process monitoring approaches for monitoring networks significantly facilitates the early detection of potential failures in complex relational systems, and therefore has been increasingly studied in recent years. Selection of an effective network monitoring method relies on the evaluation of performances of candidate methods. However, researches on systematically evaluating and comparing network monitoring methods are very few. Especially, the capability of frequently collecting data with the assistance of modern measuring devices tends to induce autocorrelations among networks. Yet performance evaluation methods for autocorrelated networks are severely lacked. This paper proposes a performance evaluation method for network monitoring based on the separable temporal exponential random graph models, which is applicable to both independent and autocorrelated networks. Further, the effects of neglecting autocorrelations on the detection power of network control charts are studied as an application of the proposed method. The simulation results show the adverse effects of autocorrelations on performances of Shewhart, EWMA, CUSUM control charts for network density, and the residual control chart is suggested in the high autocorrelation scenarios. Following the guide, a residual control chart is applied to the analysis of the Enron email networks, and anomalous events are effectively detected.

1. Introduction

Industry 3.5 is a hybrid strategy between existing Industry 3.0 and to-be Industry 4.0 to facilitate the manufacturing upgrade of emerging countries (Chien, Hong, & Guo, 2017; Chien, Lin, & Lin, 2020; Jamrus, Wang, & Chien, 2020). Digital decision making based on the analysis of information of the operational processes is a critical component of Industry 3.5 as compared with decisions making based on domain knowledge and experience in traditional Industry 3.0 (Chien et al., 2017; Hsu, Chen, & Chien, 2020; Ku, Chien, & Ma, 2020). As one application of big data analytics, statistical process monitoring (SPM) plays a crucial role in promptly identifying change-points in time from real-time data to achieve a fault-free and cost efficient running of the process (He & Wang, 2017; Megahed & Jones-Farmer, 2015; Yin & Kaynak, 2015). It serves as an efficient tool for supporting accurate and timely decision-making in Industry 3.5, which can enhance intelligent analysis capability of manufacturing industries in process control.

With the quick development of sensors and information technologies enabling Industry 3.5, large amount of complex data are generated from the production processes in manufacturing and service industries. Among these "big data", network data, which represent the interactions among related entities, play an important role. Networks are ubiquitous such as sensors transmitting data, computers exchanging information, and employees emailing on corporate operations. Machines in the factories are also connected as a network for information exchange and collaboration (Chien et al., 2017). Networks are constantly undergoing changes through time, and therefore are referred to as dynamic, or time-varying networks (Yu, Woodall, & Tsui, 2018; Wilson, Stevens, &

[*] Corresponding author at: College of Management and Economics, Tianjin University, No.92 Weijin Road, Nankai District, Tianjin, 300072, China.
E-mail address: zhhe@tju.edu.cn (Z. He).

https://doi.org/10.1016/j.cie.2020.106507
Received 31 July 2019; Received in revised form 21 April 2020; Accepted 23 April 2020
Available online 11 May 2020

Woodall, 2019). Through network monitoring, anomalous changes in dynamic networks are detected, which significantly facilitates the early detection of potential failures in complex relational systems. Network monitoring has wide applications in computer, biological and social networks such as fraud detection, intrusion detection, pathological diagnosis and corporate surveillance. Applying SPM approaches for monitoring dynamic networks has attracted rapidly growing research interests in recent years.

SPM approaches are to construct control charts for statistics summarizing the sample features. For network data, features can be characterized by summary metrics or model parameters. For metric-based SPM approaches, control charts are used to monitor network structural metrics such as centrality, distance, and transitivity measures (e.g. McCulloh & Carley, 2011; Park, Priebe, & Youssef, 2013; Priebe, Conroy, Marchette, & Park, 2005; Perry, 2020; Wang, Tang, Park, & Priebe, 2014). For model-based SPM approaches, the idea is similar to profile monitoring (Maleki, Amiri, & Castagliola, 2018). First, statistical models are built to explain the network data as realizations of random networks dependent on certain covariates, and then the SPM methods are employed to monitor model parameters (e.g. Dong, Chen, & Wang, 2020; Farahani & Kazemzadeh, 2019; Fotuhi, Amiri, & Maleki, 2018; Wilson et al., 2019; Yu et al., 2018; Zou & Li, 2017).

Most of the previous methods for monitoring dynamic networks did not consider the potential autocorrelations. The capability of frequently collecting data by the modern measuring devices, however, tends to induce autocorrelations among networks. In such a context, it is risky to directly adopt those monitoring methods since their effectiveness is unknown in the presence of autocorrelations. Simulation-based performance evaluation among candidate methods can provide a guide for selecting an effective SPM approach. Simulation is an efficient tool to mimic the operational environment, which is practical for validating the viability of operational strategies in Industry 3.5 (Chien, Chou, & Yu, 2016). Through well-designed systematic simulations, different anomalous scenarios can be mimicked, and the candidate monitoring methods can be compared under various potentially risky scenarios without requiring their real occurrences. Furthermore, when a new method for monitoring autocorrelated networks is proposed, the detection power can be well assessed through systematic performance evaluation prior to its application to real data monitoring. In addition, optimal control limits for detecting certain specific shifts may be obtained based on evaluated performances by manipulating the simulation scenarios. In summary, since the “true” network processes in practice are very rarely known and usually very complex, it is highly useful to validate the effectiveness of the monitoring methods under predetermined “ideal” conditions through simulation studies in a controllable manner (Sengupta & Woodall, 2018; Woodall, Zhao, Paynabar, Sparks, & Wilson, 2017).

There have been several studies on simulation-based performance evaluation methods, which are designed for different types of networks that can be categorized from three aspects. By the time-dependency, networks are classified into independent and autocorrelated networks. By the edge directionality, networks are classified into undirected and directed networks. Depending on whether the edge represents the interaction existence or the interaction counts, networks are classified into binary and weighted networks. Zhao et al. (2018) simulated independent undirected binary social networks with changes of interaction probability among different proportions of nodes based on the Erdös-Rényе model and evaluated the performance of the scan method proposed by Priebe et al. (2005). Yu et al. (2018) and Wilson et al. (2019) advocated the use of degree-corrected stochastic block models to simulate individual, local and global changes of interaction propensities and the change of degree variability as well as community structural changes for independent undirected weighted networks. Hosseini and Noorossana (2018) compared the performances of EWMA and CUSUM control charts for average degree and standard deviation of degree measures in detecting outbreaks within independent undirected weighted networks based on Poisson models. Komolafe, Quevedo, Sengupta, and Woodall (2019) carried out statistical evaluation of a suite of popular spectral anomaly detection methods through independent undirected binary networks simulated from the Erdös-Rényе model, the recursive matrix model and the Chung Lu model. In the above literature, researches have been conducted on performance evaluation for monitoring independent undirected binary and weighted networks. Evaluation methods for autocorrelated networks and directed networks, however, have been very little studied. In this paper, we provide a systematic framework to evaluate the performance of control charts for autocorrelated undirected binary networks based on the separable temporal exponential random graph model (STERGM) proposed by Krivitsky and Handcock (2014).

Discussions on the adverse effects of autocorrelations and improving strategies have been studied for univariate, multivariate and profile control charts (e.g. Alwan, 1992; Costa & Fichera, 2017; He, Wang, Tsung, & Shang, 2016; Khedmati & Niaki, 2016; Maragah & Woodall, 1992; Noorossana & Vaghefi, 2006; Soleimani, Noorossana, & Amiri, 2009; Vanhatalo & Kulahci, 2015; Wang & Huang, 2017). It is reasonable to conjecture that autocorrelations may have adverse effects on network monitoring as well. However, few efforts were devoted to studying such effects quantitatively. We apply the proposed performance evaluation method to the exploration of the autocorrelation effects on network control charts. Guides are provided for choosing proper control charts in the presence of autocorrelations.

The contributions of this paper are summarized as follows. First, we provide a systematic framework for evaluating network control charts based on the STERGMs with the flexibility of simulating both independent and autocorrelated networks. Second, we apply the proposed method to studying the effects of neglecting autocorrelations on the performances of some commonly-used network control charts. The simulation experiments show that (1) autocorrelations have adverse effects on performances of Shewhart, EWMA, and CUSUM control charts for network density; (2) the CUSUM control chart performs generally better in detecting the small and medium shifts in the low and medium autocorrelation scenarios; (3) a smaller weighting parameter for the EWMA chart contributes to a better performance when network autocorrelation is not high and a larger value is preferred otherwise; (4) the residual control chart is suggested in the presence of high autocorrelations. Following the guide based on the study of autocorrelation effects, we adopt a residual control chart for monitoring the Enron email networks, in which a strong autocorrelation is found. The detection of anomalous events validates the effectiveness of this study.

The remainder of this paper is organized as follows. Section 2 introduces the separable temporal exponential random graph models. Control charts used for the study of autocorrelation effects are briefly described in Section 3. In Section 4, a STERGM-based simulation approach is proposed for evaluating performances of network control charts. The design and analysis of numerical experiments are shown in Section 5. The utility of the proposed method is validated through a real example in Section 6. Section 7 is the conclusion and some topics for future researches. For readers not familiar with network control charts, some fundamentals are provided in Appendix A. Sample R codes for obtaining control limits are provided in Appendix B.

2. Separable temporal exponential random graph models

In this section, we first introduce the exponential random graph model (ERGM) for time-independent networks. Then we describe the STERGM for autocorrelated networks, which is the basis of the proposed performance evaluation method.

2.1. The exponential random graph model for time-independent networks

It is very natural to posit that networks are “built up” by its local structures such as the edges, stars and triangles (Robins, Pattison,

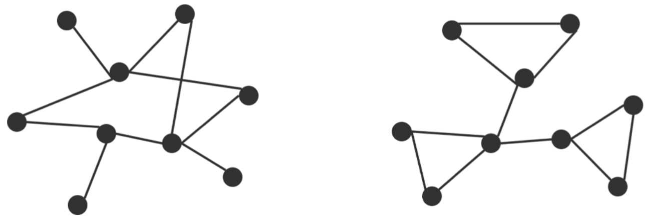


Fig. 1. Networks of 9 nodes with their overall structures dominated by stars (left) and triangles (right).

Kalish, & Lusher, 2007). The overall structure of a network is determined by its dominant local structures and the dominating degrees of different local structures. Two toy examples of networks with 9 nodes are shown in Fig. 1. The left one is dominated by stars while the right one is dominated by triangle structures.

The summary statistics of the local structures provide information about global properties of networks. For instance, the number of edges indicates the overall density of a network; the number of 2-stars quantifies the degree variability (i.e. node heterogeneity); the number of triangles characterizes the transitivity property (often interpreted as “friends of my friends are my friends” in the social network analysis area) (Frank & Strauss, 1986; Morris, Handcock, & Hunter, 2008; Snijders, Pattison, Robins, & Handcock, 2006).

Denote a random network with n nodes by Y and its observation by y. Write the edge variable between nodes i and j as Y_{ij} $(i, j \in \{1, 2, \ldots, n\})$. In the case of binary networks, $Y_{ij} = 1$ indicates there is an edge between i and j and $Y_{ij} = 0$ represents an non-edge. Denote the numbers of edges, 2-stars and triangles in an undirected network Y by $S_1(Y)$, $S_2(Y)$ and $T(Y)$. These count statistics are calculated as

$$\begin{aligned} S_1(Y) &= \sum_{1 \leqslant i<j \leqslant n} Y_{ij} && \text{number of edges,} \\ S_2(Y) &= \sum_{1 \leqslant i \leqslant n} \binom{Y_{i+}}{2} && \text{number of } 2-\text{stars,} \\ T(Y) &= \sum_{1 \leqslant i<j<h \leqslant n} Y_{ij} Y_{ih} Y_{jh} && \text{number of triangles,} \end{aligned} \tag{1}$$

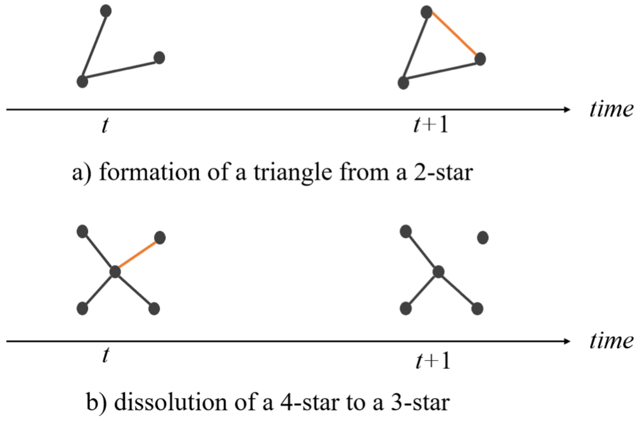
where $i, j, h \in \{1, 2, ..n\}$ are the indexes of nodes; the $+$ sign denotes summation over the index, and Y_{i+} is the degree of node i (Frank & Strauss, 1986).

Write the summary statistics of local structures in a network Y into a vector as $h(Y)$. Assume Y dependent on $h(Y)$ follows a distribution of the exponential family. Then the exponential random graph model for a random network Y is

$$P(Y = y|\theta) = \frac{\exp(\theta' h(y))}{k(\theta, \mathcal{Y})}, \tag{2}$$

where θ is the parameter vector corresponding to the statistic vector $h(y)$;$\mathcal{Y}$ denotes the space containing all possible observations of the random network Y; and $k(\theta, \mathcal{Y}) = \sum_{y^* \in \mathcal{Y}} \exp(\theta' h(y^*))$ is the normalizing constant (Hunter, 2007; Snijders et al., 2006).

By generating random observations from an ERGM by model (2), a time-independent network sequence can be obtained. For two independent networks at successive time points, the probabilities of the structural statistics are the same, and the locations of the local structures are completely random. The transition from Y^t to Y^{t+1} is through formations and dissolutions of local structures. Examples of structural transition include the formation and break of an edge, the formation of $(k + 1)$-star from a k-star and the dissolution of k-star to a $(k - 1)$-star, and the formation and dissolution of a triangle from and to a 2-star. Fig. 2 illustrates the formation of a triangle from a 2-star and the dissolution of a 4-star to a 3-star. Since networks are independent, the structural transitions are random, and the probability of Y^t is not conditional on Y^{t-1}.



a) formation of a triangle from a 2-star

b) dissolution of a 4-star to a 3-star

Fig. 2. Formation of a triangle from a 2-star (upper) and dissolution of a 4-star to a 3-star (lower) over time.

2.2. The STERGM for autocorrelated networks

Denote a network time series by $\{Y^1, Y^2, \ldots, Y^t, Y^{t+1}, \ldots\}$. An autocorrelated network process can be considered as the combination of two intermediate processes, i.e. the formation and the dissolution of a proportion of local structures with the rest part remaining the same (Krivitsky & Goodreau, 2019).

An example of edge formation and dissolution processes in a network is shown in Fig. 3. A friendship network is formed by Lisa, Mary, Peter, John, and David. At time t, five edges of Y^t correspond to the connections between John and every other person plus the connection between Peter and David. Over time, the connection pattern changes to Y^{t+1}, which includes the connections between John and every other person plus the connection between Peter and Lisa. Such an evolution can be seen as the combination of two latent processes of formation and dissolution. The formation process is to add the extra connection between Peter and Lisa based on Y^t, forming an intermediate network Y^+ with 6 edges; the dissolution process is to break the connection between Peter and David from Y^t, dissolving into an intermediate network Y^- with 4 edges. In Fig. 3, the $\cup$ operator denotes combining networks Y and Y' as a new network $Y \cup Y'$ in a way that all edges (i, j) either in Y or in Y' are put into the network $Y \cup Y'$. The $\cap$ operator denotes subsetting edges (i, j) in both Y and Y' into the new network $Y \cap Y'$. The tendency of an edge remaining undissolved reflects the autocorrelation level. Consider the connection between Peter and Mary. If their connection tend to stay unchanged for years, then their connection is highly autocorrelated. Such persistence, or say the duration of such connection, positively reflects the level of autocorrelation. Similarly, the autocorrelation dependency of other network properties can be characterized by the formation and dissolution of corresponding local structures.

By virtue of Y^+ and Y^-, the formation and dissolution processes can be directly modeled as two ERGMs

$$\begin{aligned} P(Y^+ = y^+|Y^t = y^t;\theta^+) &= \frac{\exp(\theta^+ h^+(y^+))}{k(\theta^+, \mathcal{Y}^+(y^t))}, \\ P(Y^- = y^-|Y^t = y^t;\theta^+) &= \frac{\exp(\theta^- h^-(y^-))}{k(\theta^-, \mathcal{Y}^-(y^t))}, \end{aligned} \tag{3}$$

where $Y^+ = Y^t \cup Y^{t+1}$ and $Y^- = Y^t \cap Y^{t+1}$ are the random intermediate networks and y^+ and y^- are their realizations; θ^+ and θ^- are the formation and the dissolution parameters respectively; $\mathcal{Y}^+(y^t) = \{y': y' \supseteq y^t\}$ denotes the space of all possible networks containing y^t as a subset, and $\mathcal{Y}^-(y^t) = \{y': y' \subseteq y^t\}$ denotes the space of all possible networks which are subsets of y^t;$k(\theta^+, \mathcal{Y}^+(y^t))$ and $k(\theta^-, \mathcal{Y}^-(y^t))$ are the normalizing constants (Krivitsky & Handcock, 2014). Here, to simplify the notations, the time index t in the formation and the dissolution networks $Y^{t,+}$ and $Y^{t,-}$ is omitted.

The formation and the dissolution parameters indicate the

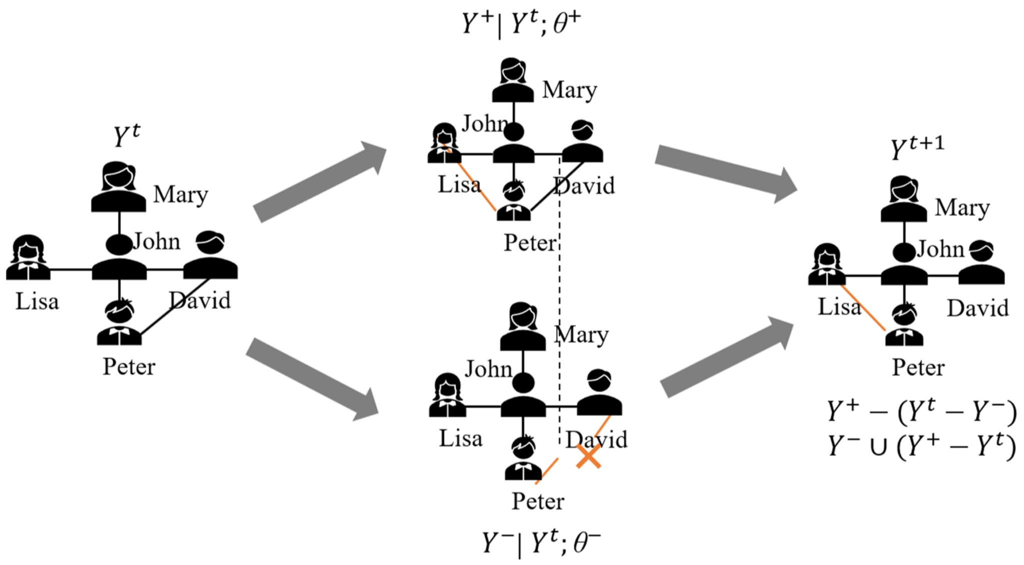


Fig. 3. Illustration of the autocorrelated network process as a combination of edge formation and dissolution processes.

tendencies of corresponding structures to newly form and dissolve over time. Therefore, increasing the formation parameters with the dissolution parameters unchanged induces increased local structure counts, and decreasing the formation parameters leads to reduced counts. Through adjusting the parameter values, shifts are simulated with respect to the levels of the density, reciprocity, degree variability and transitivity properties of a network. For statistics describing various structural characteristics, readers can refer to Snijders et al. (2006), Hunter (2007), Robins et al. (2007), Morris et al. (2008).

2.3. Parameter estimation

Since the normalizing constants $k(\theta^+, \mathcal{Y}^+(y^t))$ and $k(\theta^-, \mathcal{Y}^-(y^t))$ are hard to compute and not of our interest, parameters can be estimated by reexpressing model (3) in a conditional logistic regression form with the normalizing constant term cancelled from the models. Conditioning the probability of Y_{ij}^+ on the rest of the network Y_{ij}^{+c}, model (3) is equivalent to

$$logit(Y_{ij}^+ = 1|y_{ij}^{+c}, \theta^+) = \theta^{+'}\delta(y_{ij}^+),$$
$$logit(Y_{ij}^- = 1|y_{ij}^{-c}, \theta^-) = \theta^{-'}\delta(y_{ij}^-), \tag{4}$$

where *logit* represents the logit function defined as $logit(X = 1) = log\frac{P(X = 1)}{P(X = 0)}$, which calculates the log-odds of a binary variable X taking the value 1; θ^+ and θ^- are the formation and the dissolution parameters to be estimated; ^{c} is the complement operator and y_{ij}^{+c} represents the remaining structure in y^+ with the edge status between nodes i and j left out; $\delta(y_{ij})$ is called the *change statistic* for the node pair (i, j) corresponding to the change of statistics $h(y)$ caused by the formation of an edge between nodes i and j given y_{ij}^c (Krivitsky & Handcock, 2014). When the statistic $h(y)$ is the number of edges, the change statistic is equal to 1, meaning the number of edges of the whole network increases by 1 due to the existence of edge y_{ij}. When $h(y)$ is the number of triangles, the value of $\delta(y_{ij})$ depends on the y_{ij}^c, and is equal to the number of newly formed triangles with nodes i and j connecting from non-edge status.

Parameters θ^+ and θ^- can be estimated by the maximum pseudolikelihood method. The only difference in estimating θ^+ and θ^- is to use samples $\{y^{t,+}\}_{t=1,2,...}$ or samples $\{y^{t,-}\}_{t=1,2,...}$. Denote $\pi_{ij}(\theta)$ as the conditional probability of an edge between nodes i and j, i.e.

$$\pi_{ij}(\theta) = P(Y_{ij} = 1|Y_{ij}^c, \theta) = logit^{-1}(\sum_{r=1}^{R} \theta_r\delta_r(Y_{ij})), \tag{5}$$

where r is the index corresponding to the structural statistics in the vector $h(Y)$;$logit^{-1}$ is the inverse logistic function such that $logit^{-1}(X) = 1/(1 + \exp(-X))$. We consider the first-order temporal dependence for the network time series $\{Y^1, Y^2, ..., Y^{T+1}\}$. The pseudo loglikelihood is

$$l(\theta) = \sum_{t=1}^{T} \sum_{ij} ln[(\pi_{ij}^t(\theta))^{Y_{ij}^t}(1 - \pi_{ij}^t(\theta))^{1-Y_{ij}^t}], \tag{6}$$

where the Y^t is replaced by $y^{t,+}$ or $y^{t,-}$, and the maximum pseudo likelihood estimates (MPLEs) of θ^+ and θ^- are obtained as $\hat{\theta}^+ = \underset{\theta^+}{\operatorname{argmax}}\, l(\theta^+)$ and $\hat{\theta}^- = \underset{\theta^-}{\operatorname{argmax}}\, l(\theta^-)$ (Leifeld, Cranmer, & Desmarais, 2018). Formation and dissolution parameters θ^+ and θ^- can be estimated based on given network samples. Also, network time series can be generated from model (3) with preset θ^+ and θ^-. The network at time $t + 1$ is obtained by combining the two intermediate networks as $y^{t+1} = y^+ - (y^t - y^-) = y^- \cup (y^+ - y^t)$.

3. Network control charts for comparison

An essential part of performance evaluation framework is to construct control chart based on simulated data. In this section, we introduce some commonly-used control charts to illustrate the methods of obtaining control limits. Although many advanced methods have been proposed for monitoring networks with more complex structures, we focus on the control charts for the most basic network property, i.e. the density. Through this study of the simple structure, the pattern of the autocorrelation affecting the monitoring performances can be clearly revealed. The Shewhart, EWMA, and CUSUM control charts are very often adopted in previous studies on network monitoring. In this section, we briefly describe these control charts for the edge count measure. A residual control chart is also suggested with the autocorrelation information considered. Performances of these control charts will be compared in Section 5.

3.1. Shewhart control charts for network density

The number of edges h^t for network y^t reflects the network density property and is usually not very small in a network with frequent communications. The edge count h^t can be directly monitored through the Shewhart control chart. Under the assumption that h^t approximately follows the normal distribution $N(\mu, \sigma^2)$, the control limits for the edge count h^t are

$$UCL = \hat{\mu} + L\hat{\sigma},$$
$$LCL = \hat{\mu} - L\hat{\sigma}, \tag{7}$$

where $\hat{\mu}$ and $\hat{\sigma}$ are the estimates of mean μ and standard deviation σ; L is the upper $\alpha/2$ quantile in a standard normal distribution given the type I error α (Montgomery, 2009). Usually the mean μ is estimated as the mean of $h^1, h^2, ..., h^T$, i.e. $\hat{\mu} = \bar{h} = \frac{1}{T}\sum_{t=1}^{T} h^t$. Different estimation methods can be applied to estimate $\hat{\sigma}$. Here we use the most straightforward one, i.e. the sample standard deviation $\hat{\sigma} = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}(h^t - \bar{h})^2}$.

3.2. EWMA control chart for network density

Compared with the Shewhart control chart, the EWMA control chart is more advantageous in detecting small process shifts by incorporating the information of previous data. The EWMA statistic z_t at time t is dependent on the current edge count h^t and its previous value z_{t-1}, i.e.

$$z_t = \lambda h^t + (1-\lambda)z_{t-1},$$

where λ $(0 < \lambda \leqslant 1)$ is the weighting parameter and the starting value of z_t is usually chosen as $z_0 = \mu$ (Roberts, 1959). Usually, the weighting parameter λ is chosen to be one of several commonly used values, such as 0.05, 0.1, and 0.2.

The control limits for z_t are

$$UCL_{ewma} = \mu + L\sigma\sqrt{\frac{\lambda}{2-\lambda}[1-(1-\lambda)^{2t}]},$$
$$LCL_{ewma} = \mu - L\sigma\sqrt{\frac{\lambda}{2-\lambda}[1-(1-\lambda)^{2t}]}, \tag{8}$$

where L is a constant resulting in a pre-specified in-control average run length (ARL_0) for the control chart; μ and σ are estimated as the sample mean and sample standard deviation of h^t $(t = 1, 2, ..., T)$.

3.3. CUSUM control chart for network density

Similar to the EWMA control chart, the CUSUM control chart makes use of historical data and is usually superior in detecting small shifts as well. Denote the CUSUM statistics for detecting positive and negative shifts at time t by C_t^+ and C_t^- respectively. The CUSUM statistics C_t^+ and C_t^- are

$$C_t^+ = max[0, h^t - (\mu + K) + C_{t-1}^+],$$
$$C_t^- = max[0, (\mu + K) - h^t + C_{t-1}^-], \tag{9}$$

where the reference value K is a half of the target shifts to be detected; μ is usually estimated as the mean value of h^t $(t = 1, 2, ..., T)$ (Page, 1954). The control limits for C_t^+ and C_t^- are a constant d, which can be calculated based on simulations to achieve a pre-specified ARL_0.

3.4. Residual control chart with autocorrelation considered

An underlying assumption for the above three control charts is that samples are independent over time. To account for the autocorrelation between successive samples, we suggest the strategy of modeling the autocorrelation by a time series model and monitoring the residuals by the Shewhart control charts, which is originally proposed by Alwan and Roberts (1988). Considering the case that the number of edges for the network samples approximately follows an $AR(1)$ model, the edge count at time t is dependent on the edge count at time $t-1$, i.e.

$$h^t - \mu = \phi(h^{t-1} - \mu) + \epsilon_t,$$

where ϕ is the autocorrelation coefficient, and $\epsilon_t \sim N(0, \sigma_e^2)$. The parameter estimate $\hat{\phi}$ can be obtained through the least squares method or the maximum likelihood methods. We monitor the residuals at time t

$$e_t = h^t - \hat{\mu} - \hat{\phi}(h^{t-1} - \hat{\mu}).$$

The control limits for the residuals are

$$UCL_{resid} = L_e\hat{\sigma}_e,$$
$$LCL_{resid} = -L_e\hat{\sigma}_e, \tag{10}$$

where $\hat{\sigma}_e$ is estimated as the standard deviation of the e_t $(t = 1, 2, ..., T)$ from the Phase I samples; L_e can be obtained through simulations to achieve a pre-specified ARL_0.

4. STERGM-based simulation approach for performance evaluation

Network control charts are designed to timely detect network anomalies. To evaluate the performance of a network control chart, network time series with anomalies at certain time points can be simulated to check the ability of the control chart to signal such anomalies. In this section, we first explain the ways of controlling autocorrelation and shift levels of networks. Then, we provide a STERGM-based framework for evaluating the performance of network control charts.

4.1. Manipulating autocorrelation levels and shifts

Generating networks from a STERGM is determined by three parts: the formation and dissolution parameters θ^+ and θ^- as well as the network y^t based on which y^{t+1} is generated. To simulate temporal networks for performance evaluation, we discuss the connections between STERGM parameters and process autocorrelation along with the anomalous types.

Network structures evolve over time based on the combined processes of formation and dissolution. The formation parameter θ^+ corresponds to the tendency of dominant local structures to form in the latent network Y^+. Based on model (4), a positive θ^+ indicates a larger probability of forming a network structure from the status of non-existence than staying unchanged. The larger the θ^+ is, the more probably the structure forms. Similarly, the dissolution parameter θ^- explains the probability of persistence of an existing structure than dissolution. A smaller θ^- indicates higher probability of dissolution and lower tendency of persistence. Positive autoregression can be characterized by the persistence of network structures (Leifeld et al., 2018). The more durable a structure is, the more autocorrelated the network is. As mentioned in Section 2, formation and dissolution parameters also affect the level of the summary statistics for chracterizing network properties. Thus, the autocorrelation level of network time series and the shift level of network properties should be well controlled by adjusting the formation and the dissolution parameters.

When intermediate networks for the formation and the dissolution processes are from the same distribution (i.e. $Y^+ = Y^- = Y^t = Y^{t+1}$), such a dynamic process is in fact an independent process. The parameters have the relationship $\theta^+ = \theta^- = \theta$, where θ is the parameter of the ERGM for network Y^t $(t = 1, 2, ..., T)$. Through increasing the dissolution parameter θ^- from θ, we set the autocorrelation of network processes to different levels.

Basically, the simulation of network time series for performance evaluation involves the setting of autocorrelation levels and shift levels of local structure counts. Since adjusting the autocorrelation level through the dissolution parameter also changes the number of featured local structures, we need to set θ^+ accordingly in the formation process to remedy the changes of structure counts in the dissolution process. To hold the number of local structures at a certain level, we can set the formation parameter θ^+ such that the change statistics $\delta(y_{ij}^+) = \delta(y_{ij}^-)$, meaning the increased number of local structures due to formation is equal to the decreased number due to dissolution.

4.2. Performance evaluation framework based on STERGM

In statistical process monitoring, a well-performed control chart is expected to (1) falsely alarm at a very small probability in an in-control status, and (2) detect process shifts in a timely manner. A commonly-used criterion for evaluating the performance is the average run length (ARL) for a control chart to signal an outlier. The ARLs of a control chart in in-control and out-of-control statuses are usually denoted by ARL_0 and ARL_1. The ARL_0 of a control chart is expected to be large while the ARL_1 value is expected to be small. To evaluate and compare performances of control charts, control limits for different control charts are set to achieve the same ARL_0 for an in-control process so that their performances are compared through the ARL_1 values for out-of-control processes.

For simplicity of illustration, we focus on the change of network density in an undirected network and simulate networks with shifts of the edge count. The autocorrelation level is adjusted through the dissolution parameter θ^- and the edge count is held at a target level through the formation parameter θ^+. Write the total number of node pairs as $C_n^2 = n(n-1)/2$. The edge formation parameter is set as

$$\theta^+ = -log\left(\frac{1+\exp(\theta^-)}{m/(C_n^2 - m)} - 1\right) \tag{11}$$

to achieve an expected edge count of m (Krivitsky & Handcock, 2018). Denote the edge count of network sample y^t by h^t and the charting statistic for monitoring y^t by z^t. The framework of performance evaluation for network control charts is as follows.

Phase I settings (at time $1, 2, 3, \ldots, T$)

(a) *Initial setup:*

Set the number of nodes n and the expected edge count m. Randomly generate an initial network y^1 with n nodes and m edges. Fit model (2) to y^1 and estimate the parameter θ. Set the dissolution parameter $\theta_0^- = \hat{\theta}$ for the independent process or $\theta_0^- > \hat{\theta}$ for a positively autocorrelated process. According to Eq. (11), set the formation parameter $\theta_0^+ = -log\left(\frac{1+exp(\theta_0^-)}{m/(C_n^2 - m)} - 1\right)$ for in-control networks.

(b)*Network generation:*

Draw two random intermediate networks $y^{1,+}$ and $y^{1,-}$ from the probability distributions $P(Y^{1,+} = y^{1,+} | Y^1 = y^1; \theta_0^+)$ and $P(Y^{1,-} = y^{1,-} | Y^1 = y^1; \theta_0^-)$ based on model (3). Obtain a network sample y^2 at time $t = 2$ as $y^2 = y^{1,+} - (y^1 - y^{1,-})$.

Repeat drawing intermediate networks and generating network samples y^{t+1} based on sample y^t for $t = 2, 3, \ldots, T$.

(c) *Estimation of control limits:*

Calculate the numbers of edges for each network sample and obtain the sequence of edge counts $h^1, h^2, \ldots, h^T$. Estimate the control limits UCL and LCL for the charting statistic z^t.

Phase II settings (at time $T+1, T+2, T+3, \ldots$)

(a) *Initial setup:*

Set the expected edge counts as m_*. Generate network y_*^T with n nodes and m_* edges. Set the edge dissolution parameter $\theta_*^- = \theta_0^-$ and calculate the edge formation parameter as $\theta_*^+ = -log\left(\frac{1+exp(\theta_*^-)}{m_*/(C_n^2 - m_*)} - 1\right)$.

(b) *Network generation:*

Draw two random intermediate networks $y_*^{T,+}$ and $y_*^{T,-}$ from the probability distributions $P(Y^{T,+} = y_*^{T,+} | Y^T = y_*^T; \theta_*^+)$ and $P(Y^{T,-} = y_*^{T,-} | Y^T = y_*^T; \theta_*^-)$. Obtain a network sample y^{T+1} as $y^{T+1} = y_*^{T,+} - (y_*^T - y_*^{T,-})$.

Compute the charting statistic z^{T+1}. Record the run length as $RL = 1$ if $z^{T+1} > UCL$ or $z^{T+1} < LCL$. Otherwise, draw intermediate networks $y^{T+1,+}$ and $y^{T+1,-}$ based on model (3) with parameters θ_*^+, θ_*^- and sample y^{T+1}.

Repeat drawing new networks y^t and calculating z^t for $t = T+2, T+3, \ldots$until $z^t > UCL$ or $z^t < LCL$ and a $RL = t - T$ is recorded.

(c) *Evaluation of run lengths:*

Repeat the network generation step for N times and obtain N run lengths. Calculate ARL as $ARL = \frac{1}{N}\sum_{i=1}^{N} RL_i$ to evaluate the performance of the network control chart.

This STERGM-based performance evaluation framework can be used to simulate process shifts and the changes of autocorrelation. A process shift of edge counts is introduced by setting $m_* \neq m$. When $\theta_*^- \neq \theta_0^-$, it indicates the autocorrelation of edge counts changes. In the following study on evaluating effects of neglecting autocorrelation to network monitoring performance, we set $\theta_*^- = \theta_0^-$, assuming the autocorrelation of network structures does not change.

5. Simulation experiments

To explore the effects of autocorrelations on network control charts, we employ the STERGM-based simulation framework to evaluate performances of the Shewhart, EWMA, and CUSUM control charts for edge counts under scenarios of low, medium and high levels of autocorrelation as well as the independent scenario. For the EWMA control charts, the commonly-used constants 0.05, 0.10 and 0.20 for λ are used for investigation. For the CUSUM control charts, we use C_t^+ and C_t^- statistics to monitor the positive and negative shifts respectively. Under the independent scenario, the residual control chart is equivalent to the Shewhart control chart, and therefore is not investigated.

5.1. Experimental settings

We set the number of nodes as $n = 50$ and expected edge counts as $m = 50$. In the Phase I analysis, we randomly generate an undirected network of 50 nodes and 50 edges as the initial network y^1 as shown in Fig. 4(a). We fit model (2) to y^1 by specifying the network statistic as the number of edges, i.e. $h^1 = 50$. The model parameter θ is estimated to be -3.16 by the maximum likelihood estimation method through the R package "ergm" (Handcock et al., 2018; Hunter, Handcock, Butts, Goodreau, & Morris, 2008).

By setting $\theta_0^+ = \theta_0^- = \theta = -3.16$, we obtain an independent network sequence. For autocorrelated scenarios, we set $\theta_0^- = -1$, 0, log(30), corresponding to low, medium and high duration of an edge. The formation parameters are calculated to be $\theta_0^+ = -3.44, -3.83, -6.95$ based on Eq. (11) to assure the expected edge count equal to m. Next, we generate networks for Phase I analysis based on model (3) and estimate

(a) Initial network y^1

(b) Network y^{1000}

Fig. 4. Plots of the initial network y^1 and the 1000^{th} network in Phase I under the independent scenario.

Table 1
Settings of edge formation and dissolution parameters for different autocorrelation levels as well as the first-order partial autocorrelation coefficient of edge counts.

Scenario	Formation	Dissolution	PACF coef
None	−3.16	−3.16	0
Low	−3.44	−1	0.22
Medium	−3.83	0	0.47
High	−6.59	3.4	0.97

the mean and variance of the edge number in scenarios of different autocorrelation levels. To reduce the effect of variation due to parameter estimation to the performances, the sample size in Phase I is set as 1000. Network time series for four autocorrelation levels are simulated through the R package "tergm" (Krivitsky & Handcock, 2018). Fig. 4(b) shows the simulated 1000^{th} network in the independent case.

Independence of the edge counts for network time series with dissolution parameter −3.16 is validated by the Ljung-Box test. The first-order partial autocorrelation coefficients of edge counts for the four levels of network autocorrelation are 0, 0.22, 0.47, 0.97 as listed in Table 1. To guarantee the comparability of these control charts, we obtain the control limits through simulating in-control networks to achieve $ARL_0 = 200$. The exact ARL_0 values for these control charts can be found in Table 3. Table 2 lists the charting parameter estimates obatined for the four types of control charts in scenarios with no, low, medium, and high autocorrelations. To illustrate the method of obtaining the charting parameters, we present the sample R codes in Appendix B. This sample shows the algorithm of calculating L for the Shewhart control chart under the medium correlation scenario. With this example as a reference, charting parameters for other control charts can be obtained in a similar manner.

Next step is to evaluate the performances of the four control charts in Phase II monitoring. *ARL*s for different process shifts are calculated. We set the shifts of edge count levels as $\delta = \pm 10\%m, \pm 25\%m, \pm 40\%m$. Considering both positive and negative shifts, we randomly generate networks with expected edge counts of 55, 62, 70, 45, 38, 30 as the initial networks for Phase II monitoring. The edge formation parameters are calculated based on the shifted expected edge counts with the dissolution parameters remaining the same as in Phase I. For each combination of the autocorrelation levels and the edge count levels, we simulate network time series based on model (3) and record the *RL*s. We run the simulations for 1000 times and obtain ARL_1 values in each out-of-control scenario.

To illustrate how the control charts detect the shifts, we simulate four sets of data corresponding to the four autocorrelation scenarios based on the STERGMs. In each scenario, we randomly simulate 30 networks with no shifts for an illustration of the in-control status. Then we simulate 30 networks with the largest positive shift as set in the Phase II monitoring experiments. Charting parameters are from Table 2. Figs. 5–8 are the Shewhart, EWMA ($\lambda = 0.05$), CUSUM, and residual control charts for the edge count in different autocorrelation scenarios. In the Shewhart control charts, a jump can be seen after the 30^{th} network in the four plots, especially in the high-autocorrelation scenario. The anomalous shifts are almost immediately detected in all scenarios since the shifts are very large. In the EWMA and CUSUM control charts, we observe a gradual delay in shift detection with the autocorrelation increasing. In the residual control charts, we can see that the detection of shifts are more timely with the autocorrelation increasing.

Table 2
Estimates of charting parameters for different control charts under scenarios with no, low, medium and high autocorrelations.

		None	Low	Medium	High
Shewhart	*UCL*	69.00	69.00	69.00	64.00
	LCL	31.00	31.00	31.00	35.99
EWMA-$L\sigma$	$\lambda = 0.05$	15.83	19.08	23.94	47.76
	$\lambda = 0.10$	17.10	20.60	25.30	42.14
	$\lambda = 0.20$	18.24	21.57	25.74	34.00
CUSUM-h	Positive shifts	4.57	4.48	8.50	15.80
	negative shifts	4.36	4.26	8.02	16.80
Residual	*UCL*	–	18.99	17.12	5.20
	LCL	–	−18.99	−17.12	−5.20

5.2. Analysis of experimental results

The *ARL*s of the Shewhart, EWMA, CUSUM, and residual control charts are listed in Table 3 under each combination of process shifts and autocorrelation levels. Comparing ARL_1 values by autocorrelation levels in Table 3, we can see there is an overall trend that the ARL_1 increases with the autocorrelation for the three types of control charts, which do not consider the autocorrelation information. This indicates their detection power decreases with autocorrelation increasing. Especially, the ARL_1 values of the Shewhart and EWMA control charts for detecting small and medium shifts (i.e. $m_* = 55, 62, 45, 38$) increase dramatically when the autocorrelation level grows from medium to high. In general, the CUSUM control charts perform the best in detecting the small and medium shifts in the low and medium autocorrelation scenarios. Compared with the Shewhart and EWMA control charts, adverse effects of the high autocorrelation on the CUSUM control charts performance are smaller. When the shift is large (i.e. $m_* = 70, 30$), the effect of autocorrelation is not so strong since the shifts are large enough and all the control charts can detect large shifts of network density quite well. The performances of the residual control charts are not favorable in detecting the small and medium shifts in the low and medium autocorrelation scenarios. With the autocorrelation increasing to a higher level, the residual control chart performs exceptionally well in detecting both medium and large shifts.

In addition, it is found that there is an interaction effect between autocorrelation level and the weighting parameter on the performances of EWMA control charts. As can be seen in Table 3, in scenarios of no, low and medium autocorrelation levels, the ARL_1 values increase with the weighting parameter λ increasing from 0.05, 0.1 to 0.2. It implies that choosing a smaller λ leads to a better performance. In contrast, in the scenario of high autocorrelation, the ARL_1 values decrease with λ going up. In this case, a larger weighting parameter is preferred for process monitoring.

In summary, autocorrelation has adverse effects on performances of Shewhart, EWMA, and CUSUM control charts since they usually assume independence between samples over time. The EWMA control charts are most sensitive to the increase of autocorrelation in detecting process shifts. A smaller weighting parameter contributes to a better performance when the autocorrelation is not high and a larger value is preferred otherwise. The CUSUM control charts perform generally better in detecting the small and medium shifts in the low and medium autocorrelation scenarios. In the presence of relatively high autocorrelations, we suggest the adoption of the residual control charts since it performs the best in detecting medium and large shifts compared to other control charts.

6. Application to the Enron email networks

In this section, we analyze the Enron email communication networks as an example. The version of data is from Priebe et al. (2005). This dataset contains the email records of 184 unique email addresses from 150 users. An edge exists if there is at less one email between two addresses in a week. In the year 2001, crucial events happened to the Enron company including the revelation of its financial scandals as well as its bankruptcy. A total of 53 networks in 2001 are obtained for analysis.

We take the first 20 networks as the Phase I data to estimate the

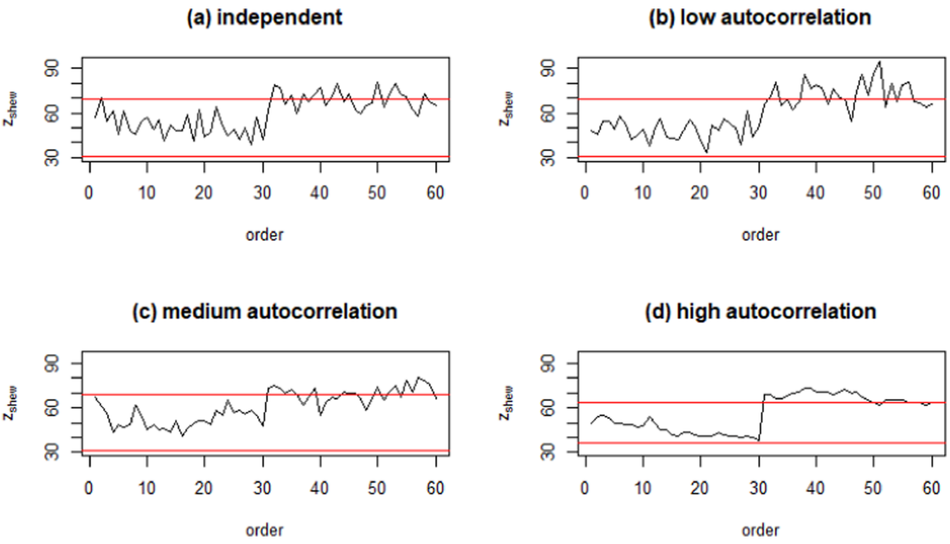


Fig. 5. Shewhart control chart for detecting a positive shift in different autocorrelation scenarios.

charting parameters. Firstly, we test whether there is autocorrelation among the edge counts of the 20 networks. Fig. 9 is the autocorrelation and partial autocorrelation plots, in which a strong autocorrelation can be observed. The partial autocorrelations of both the lag 1 and lag 2 are shown to be significant from the pacf plot in Fig. 9. The value of the lag-1 partial autocorrelation is above 0.6, which is relatively high. Therefore, we consider the use of the residual control chart for monitoring the Enron networks. By the AIC criterion, the first order autoregressive model is preferred. The parameter ϕ of the AR(1) model are estimated as 0.62, which is significant with a p-value smaller than 0.01. The standard deviation of the residuals is estimated as 23.05. Diagnostic plots for the AR(1) model are shown in Fig. 10. No special patterns are observed in the plot of the standardized residuals. No autocorrelations are significant based on the acf plot for the residuals. Further, the residuals are independent since the p-values for the Ljung-Box statistic are over 0.05 for up to 10 lags as shown in the bottom plot. Therefore, the AR(1) model fits the data well.

To calculate the control limits for the residuals, we can simulate the network time series by the STERGM to achieve an ARL_0 of around 500 for a small false alarm probability. We fit the STERGM to the sample networks from the first 20 weeks as the in-control model. The formation and dissolution parameters are estimated to be −5.84 and −0.29 respectively. We simulate 1000 network time series of size 20 based on the fitted model and calculate the lag-1 partial autocorrelations. The histogram for the lag-1 partial autocorrelation coefficients from the simulated network time series are presented in Fig. 11. The red vertical line is the reference line for the partial autocorrelation coefficient estimated from the Enron network observations for the first 20 weeks. It can be seen that most of the pacf values from the simulated samples are around the reference value obtained from the real observations, showing that the fit of the STERGM to the Enron data is satisfactory.

We use the STERGM as the network generator and obtain the control limit for the residual control chart with ARL_0 of approximately 500 by following the performance evaluation framework provided in Section 4. The control limit is calculated to be 48.41 and −48.41. Fig. 12 shows the residual control charts. Dates for the outliers are the weeks around 5/21, 5/28, 6/04, 06/11, 07/23, 09/10, 09/17, 09/24, 10/08, 10/22, 11/05, 11/12, 11/19, 12/03, and 12/17. These outliers are around May to June, September to November, corresponding to the “Secret” meeting among Schwarzenegger, Lay, and Milken in LA on May 17, the fierce arguments in dealing with the California energy crisis in June, the meeting of Skilling with analysts and investors on

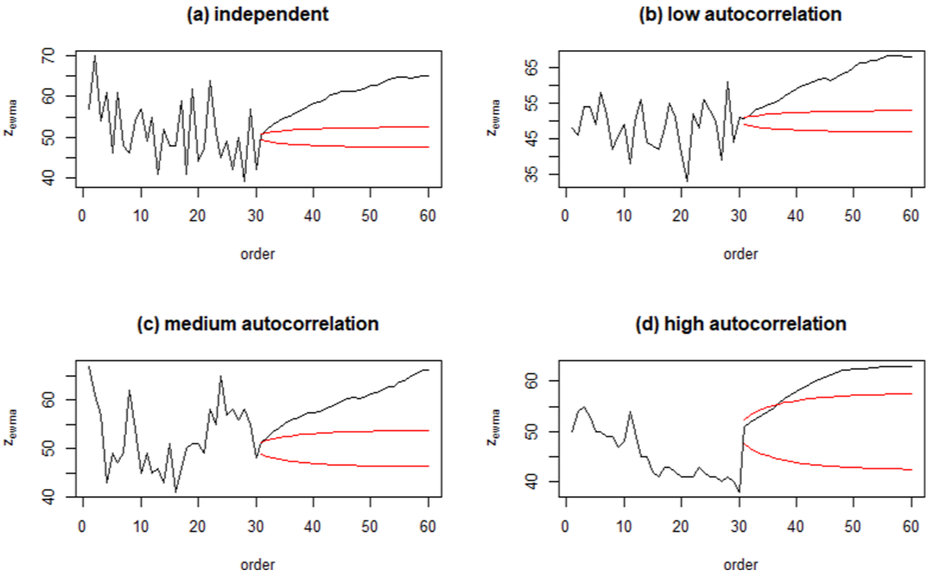


Fig. 6. EWMA control chart for detecting a positive shift in different autocorrelation scenarios.

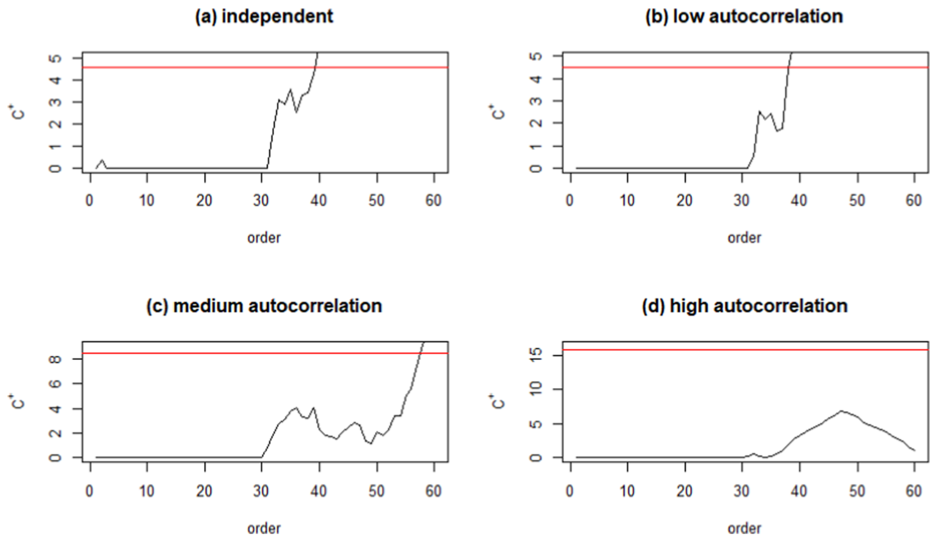


Fig. 7. CUSUM control chart for detecting a positive shift in different autocorrelation scenarios.

July 24–25, the employ meeting with deceptive financial announcement by Lay on Sep. 26, the report of a $618 million third-quarter loss of Enron on Oct. 16, the informal probe by SEC on Oct. 17, the rival Dynegy agreeing to buy Enron for about $9 billion in stock and cash on Nov. 9, and Enron filing for bankruptcy protection on Dec. 2. The crucial events are correctively detected through the residual control chart with the limits obtained based on the STERGM-based performance evaluation approach.

7. Conclusions and future researches

As an important category of "big data", networks are ubiquitous in manufacturing and service industries. Modern measuring devices enable collecting data at a high frequency and inevitably induce autocorrelations in dynamic networks. Performance evaluation of control charts contributes to assessing and selecting an effective monitoring method for autocorrelated networks, which thereby facilitates the early detection of potential failures favorable to the engineering management in the era of Industry 3.5. This paper proposed a performance evaluation approach for network control charts based on the separable temporal random graph models. Interpreting network autocorrelation as the persistence of local structures, we provided the method of controlling the autocorrelation level by adjusting the dissolution parameter. Through adjusting the formation parameter, we showed the way of holding the number of local structure at a constant level.

The proposed STERGM-based simulation framework were applied to the study of the autocorrelation effects on performances of the Shewhart, EWMA, CUSUM, and residual control charts for the edge count. Results of the numerical experiments have shown that autocorrelations have adverse effects on performances of the Shewhart, EWMA, and CUSUM control charts. For EWMA control charts, a smaller value should be chosen in the presence of low and medium autocorrelations while a larger value is preferred otherwise. Generally, the CUSUM control charts perform better in detecting small and medium shifts in the low and medium autocorrelation scenarios. In the presence of relatively high autocorrelations, we suggest the adoption of the residual control charts since it performs the best in detecting medium and large shifts.

The utility of the proposed STERGM-based framework and the effectiveness of the guide obtained from the autocorrelation effect study were illustrated through the Enron email communication network data. In this real case, the autocorrelation of the edge count statistic was shown to be at a medium-to-high level. Therefore, a residual control chart was adopted. Control limits were obtained through the proposed STERGM-based framework to achieve a pre-specified ARL_0 value. The fact that crucial events are correctively detected shows the

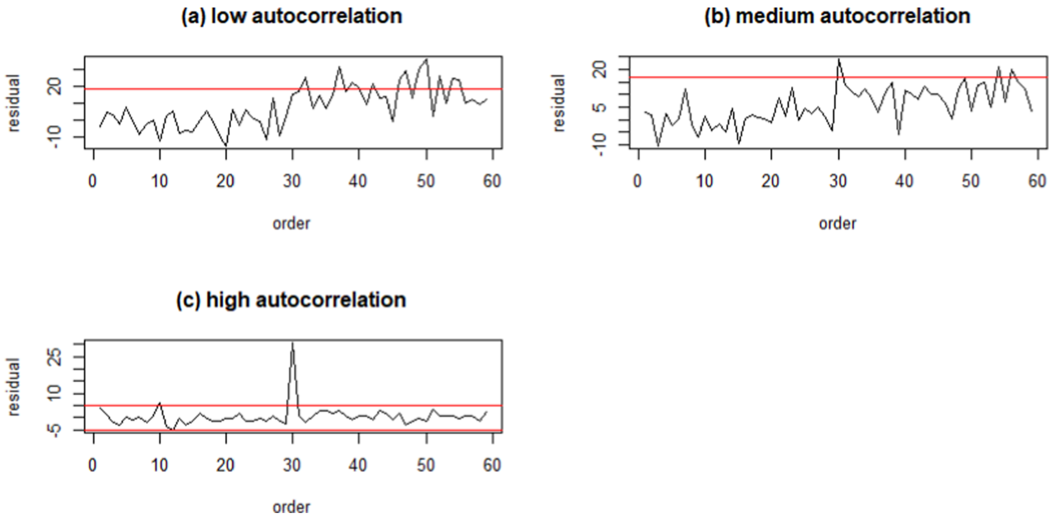


Fig. 8. Residual control chart for detecting a positive shift in different autocorrelation scenarios.

Table 3
ARLs of various control charts for detecting shifts of network density under scenarios with no, low, medium and high autocorrelations.

chart	level	$m_* = 50$	$m_* = 55$	$m_* = 62$	$m_* = 70$	$m_* = 45$	$m_* = 38$	$m_* = 30$
Shewhart	None	200.60	38.19	6.10	1.84	97.30	9.31	1.72
	Low	200.38	40.94	6.97	2.04	94.81	10.89	2.06
	Medium	200.45	47.01	8.23	2.36	105.94	13.19	2.24
	High	200.59	116.39	21.66	1.01	144.87	27.90	1.00
EWMA $\lambda = 0.05$	None	199.99	10.70	3.56	2.30	10.43	3.40	2.20
	Low	200.09	15.54	4.49	2.65	15.48	4.48	2.53
	Medium	200.10	22.89	6.27	3.26	23.20	6.15	3.02
	high	199.95	113.70	30.26	8.62	116.20	27.67	7.73
EWMA $\lambda = 0.1$	none	199.96	11.89	3.88	2.41	11.86	3.71	2.32
	Low	200.19	17.37	4.83	2.78	17.96	4.86	2.65
	Medium	200.01	25.22	6.78	3.46	27.45	6.70	3.28
	High	200.12	114.12	29.29	7.18	120.60	27.40	6.45
EWMA $\lambda = 0.2$	None	200.17	14.12	4.10	2.49	14.45	3.94	2.39
	Low	200.20	20.48	5.19	2.90	22.25	5.22	2.76
	Medium	200.28	28.71	7.22	3.52	34.80	7.31	3.35
	High	199.76	111.70	26.59	5.31	125.83	25.96	4.74
CUSUM	None	203.48	12.14	4.05	2.33	4.69	2.68	1.96
	Low	202.98	12.50	4.16	2.44	11.77	3.94	2.28
	Medium	199.76	22.77	7.42	4.17	21.95	6.97	3.76
	High	199.48	65.81	16.97	7.21	69.68	16.15	7.32
Residual	None	–	–	–	–	–	–	–
	Low	202.38	54.66	10.33	2.55	142.68	20.75	2.74
	Medium	200.76	72.83	14.21	3.15	189.73	51.81	4.82
	High	203.97	85.71	1.00	1.00	162.09	1.00	1.00

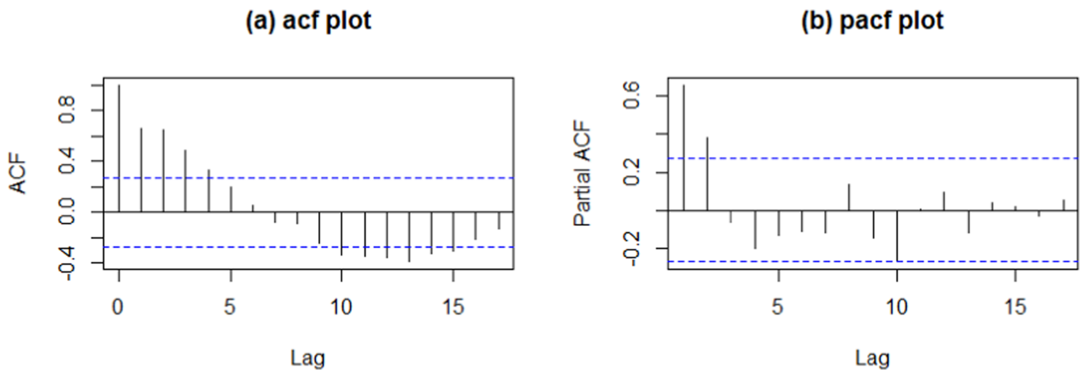


Fig. 9. The ACF and PACF plots for the density of Enron data from the first 20 weeks in the year 2001.

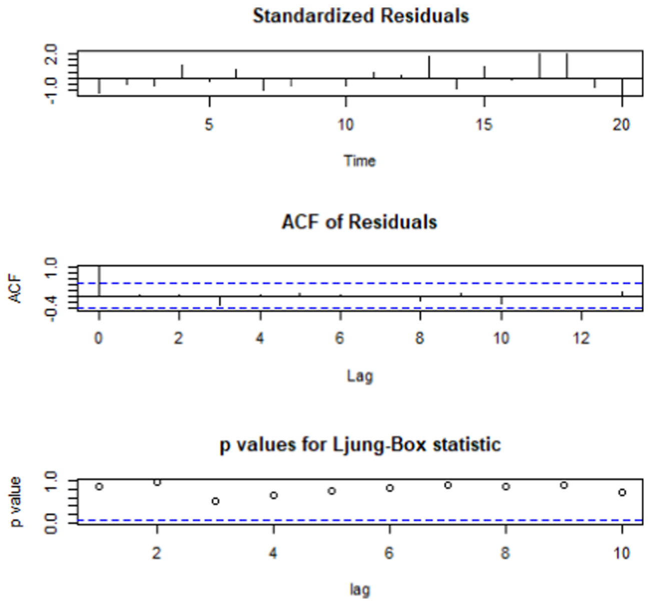


Fig. 10. Diagnostic plots of the AR(1) model.

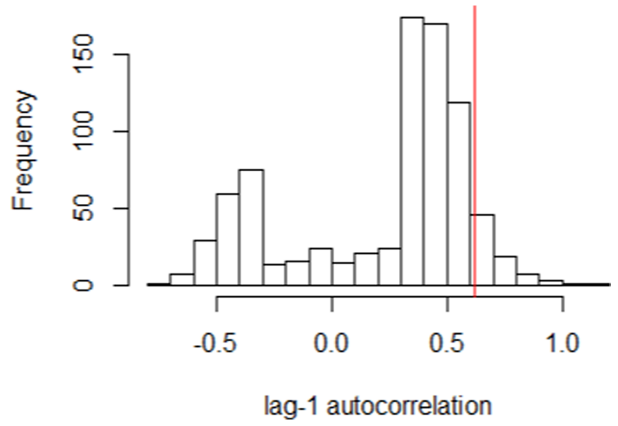


Fig. 11. The histogram of the lag-1 autocorrelation values from the simulated samples.

practicability of our method.

There are also some limitations of this paper. First, the focus of current study is mainly on the performance evaluation of network control charts in detecting anomalous changes of network density. In practice, however, network formation is usually dependent on multiple dominant local structures such as the stars and triangles as mentioned

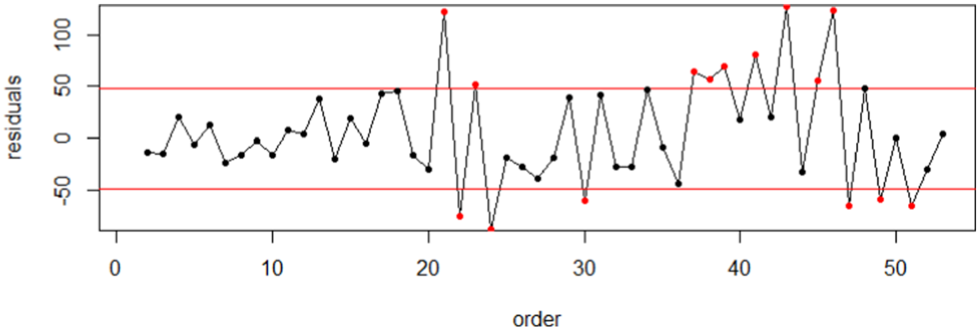


Fig. 12. The residual control chart for the Enron email networks.

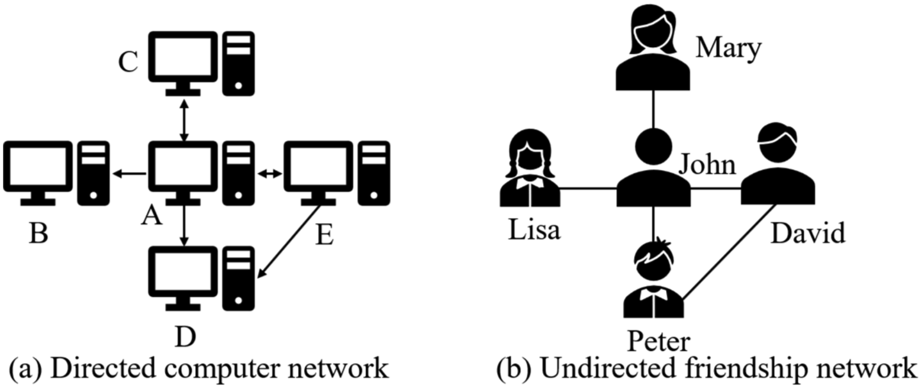

C
B
A
E
D
(a) Directed computer network
Mary
John
Lisa
David
Peter
(b) Undirected friendship network

Fig. 13. Examples of a directed computer network and an undirected friendship network.

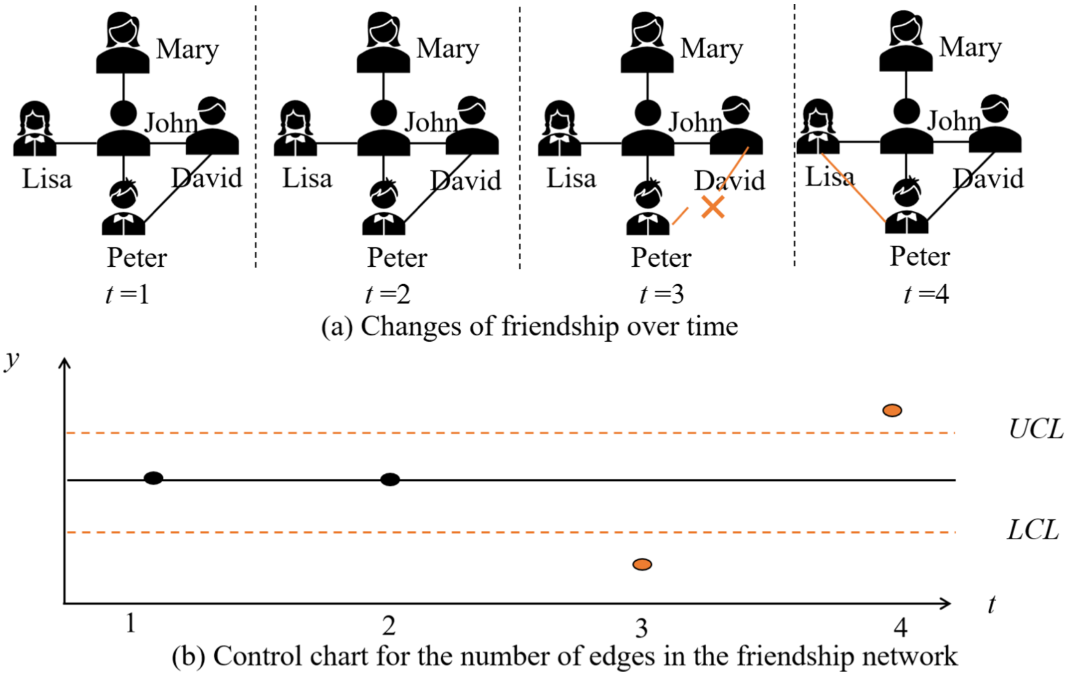

Mary
John
Lisa
David
Peter
t =1
t =2
t =3
t =4
(a) Changes of friendship over time
y
UCL
LCL
t
1
2
3
4
(b) Control chart for the number of edges in the friendship network

Fig. 14. An illustrative example of using control chart for monitoring the changes of a friendship network.

in Section 2.1. Therefore, the joint influence of autocorrelations of multiple network properties is worth further studying. Second, we only discussed the first-order autocorrelation of network structures. Networks with higher order autocorrelations can be investigated as a future research direction. Third, though the residual control chart is suggested for high autocorrelation scenarios to reduce adverse effects of neglecting autocorrelation, its performance is not so well as other methods in the low and medium autocorrelation scenarios. More studies should be conducted on methods for monitoring networks with low and medium autocorrelations.

CRediT authorship contribution statement

Panpan Zhou: Conceptualization, Software, Writing - original draft. **Dennis K.J. Lin:** Conceptualization, Supervision, Writing - review & editing. **Xiaoyue Niu:** Methodology, Software. **Zhen He:** Supervision, Writing - review & editing, Funding acquisition.

Acknowledgments

This work was supported by the National Key R&D Program of China [Grant No. 2016QY02D0301]; the National Natural Science Foundation of China [Grant No. 71661147003, 71672122, 71871204, and 71802110].

Appendix A. Fundamentals of network control charts

Networks are usually used to describe the relationships between the different entities in complex systems. A network consists of a set of nodes and the edges connecting node pairs. For example, in a local computer network, each computer is a node and different computers form edges if they transmit data from one to another; in a social network, each person is a node and those who communicate with each other form edges; in a manufacturing network, the production equipments are nodes and two equipments form a directed edge if one sends parts to the other. Fig. 13 shows an example of a directed computer network and an undirected friendship network. In the computer network, five computers indexed from A to E are the nodes. Data are sent from A to D, A to B, E to D, A to C, C to A, A to E, and E to A. Therefore, in this network, there are 7 directed edges. In the friendship network, Lisa, Mary, David, and Peter are friends of John; David and Peter are also friends. Here the friendship are always mutual with no direction. The five people are the nodes and they form an undirected network with 5 edges.

Control charts are commonly-used to monitor whether a process with random noises stays statistically in-control over time. Charting statistics such as the sample mean, the EWMA, and CUSUM statistics are constructed. Control limits are obtained based on the distribution of the charting statistics for anomaly detection. Points going beyond the limits are identified as outliers, indicating potential assignable causes for process changes. A network control chart is the application of control charting methods to network data, aiming to detect anomalous changes of networks. For example, Fig. 14(a) illustrates the evolution of a friendship network formed by Lisa, Mary, Peter, John, and David. From times $t = 1$ to 2, the network stays unchanged. At time $t = 3$, connection between David and Peter breaks. At time $t = 4$, the connection between Lisa and Peter is newly formed. If we concern about the overall connectivity in the network, a possible means is to directly monitor the total number of edges, which is shown in Fig. 14(b). Consider the extreme case that only 5 is acceptable and the control limits are set as 4.5 and 5.5. Then the networks at times 3 and 4 are outliers since they have 4 and 6 edges, respectively. Plot the edge count over time, and we identify the network at times 3 and 4 as outliers.

A good control chart is expected to detect the process changes quickly in the out-of-control case while not much falsely alarming when the process is in-control. These two goals, however, cannot be achieved simultaneously. In other words, increasing the in-control performance will lead to a decrease of the out-of-control performance. A typical way for method comparison is to set the in-control performances equal, and compare the out-of-control performances in detecting shifts of different scales. The performance of a control chart is usually evaluated based on the run length (*RL*) distribution, which may be obtained through analytical derivation and/or through simulation studies. For networks with complex data structures, the derivation of a theoretical form of the *RL* distribution is usually very difficult. Therefore, a simulation based performance evaluation approach is essential to obtain its empirical distribution of *RLs*, based on which metrics such as the average run lengths (*ARLs*) and the standard deviation of *RLs* are calculated.

Appendix B. Sample R codes for obtaining control limits

```
library(tergm)
library(statnet)
library(network)

#Initial Network: size of 50 & edge count of 50
n<-50 # network size
target.stats<-edges<-50 # edge count
g0<-network.initialize(n,dir = FALSE)
g1<-san(g0 edges,target.stats = target.stats,verbose = TRUE) # a random initial network

# Calculate the density
dyads<-network.dyadcount(g1)
edges<-network.edgecount(g1)
D0<-edges/dyads

# Estimate ergm coefficient
ergm.fit<-ergm(g1 edges)
theta0<-ergm.fit$coef #-3.157

# Settings of independent case, small/medium/large autocorrelation
#[0] independent case
coef.diss.0 <-coef.form.0<-theta0 # −3.157 in my case
#[1] small autocorrelation
coef.diss.1<–1 # increase to coef.diss.1, leading to a partial autocorrelation coefficient of 0.22
#[2] medium autocorrelation
coef.diss.2<-log(1) # increase to coef.diss.2, leading to a partial autocorrelation coefficient of 0.47
#[3] large autocorrelation
coef.diss.3<-log(30) # increase to coef.diss.3, leading to a partial autocorrelation coefficient of 0.97

# make a list of coef.diss
coef.diss.list<-lapply(0:3,function(i) get(paste0(”coef.diss.”,i)))
names(coef.diss.list)<-paste0(”coef.diss.”,0:3)

coef.form.f<-function(coef.diss,density) -log(((1 + exp(coef.diss))/(density/(1-density)))-1) # the function for calculating formation coefficient given the dissolution parameter
coef.form.ls<-lapply(coef.diss.list,function(x) coef.form.f(x,density = D0))

# define the stergm simulation function
```

```
stergm.sim.fun<-function(coef.diss = NULL, coef.form = NULL, density = NULL, start.net,timeslice = 1000, seed = NULL, nsim = 1, output = c(“networkDynamic”),
    return.type = c(“stat”)){
  if(is.null(coef.diss)&is.null(coef.form)){return(warning(“Please at least provide one of the coef.form and coef.diss parameters.”))}
  if(is.null(coef.diss)&{!is.null(coef.form)}){coef.diss<-coef.diss.f(coef.form,density)}
  if(is.null(coef.form)&{!is.null(coef.diss)}){coef.form<-coef.form.f(coef.diss,density)}
  net.sim<-simulate(start.net,
    formation = edges,
    dissolution = edges,
    monitor = ”all”,
    coef.form = coef.form,
    coef.diss  = coef.diss,
    time.slices  = timeslice,
    nsim = nsim,
    seed = seed,
    output = output
  )
  stat.sim<-as.vector(attributes(net.sim)$stats[,”edges”])
  if(return.type= =”stat”){return(stat = stat.sim)}
  if(return.type= =”net”){return(net = net.sim)}
  if(return.type= =”both”){return(list(net = net.sim,stat = stat.sim))}
}

# define the function for calculating the ARL for the Shewhart control chart
shew.cc.rl.fun<-function(mu, sig, L, coef.diss = NULL, coef.form = NULL, density = NULL, start.net, timeslice = 1000, seed = NULL, nsim = 1, output  = c(”networkDynamic”),
    return.type  = c(”stat”)){
  stat<-mu
  ucl<-mu + L∗sig
  lcl<- mu-L∗sig
  j<-0
  stat<-stergm.sim.fun(coef.diss = coef.diss, coef.form = coef.form, density = density, start.net = start.net, timeslice = timeslice, seed = seed, nsim = nsim, output
     = c(”networkDynamic”), return.type  = c(”stat”)) # simulate networks through stergm.sim.fun()
  rl<-which(stat>ucl | stat<lcl)[1];
rl < -ifelse(!is.na(rl),rl,timeslice)
  return(rl)
}
shew.cc.arl.fun<-function(n.rl = 1000,mu, sig, L, coef.diss = NULL, coef.form = NULL, density = NULL, start.net, timeslice = 1000, seed = NULL, nsim = 1, output
     = c(”networkDynamic”), return.type  = c(”stat”)){
  rl.vec<-unlist(lapply(1:n.rl,function(i) shew.cc.rl.fun(mu = mu, sig = sig, L = L, coef.diss = coef.diss, coef.form = coef.form, density = density, start.net = start.net,
    timeslice = timeslice, seed = seed, nsim = nsim, output  = c(”networkDynamic”), return.type  = c(”stat”))))
  arl<-mean(rl.vec)
  return(c(arl = arl))
}

# Take the medium correlation scenario as an example
# simulate the Phase I samples to estimate the standard deviation
stat.ic<-stergm.sim.fun(coef.diss = coef.diss.2, coef.form = NULL, density = D0, start.net = g1, timeslice = 1000, seed = NULL, nsim = 1)
sig<-sd(stat.ic)
# calculate L for a pre-specified value for $ARL_0$
# take the low autocorrelation scenario as an example
arl.set<-200 # target $ARL_0$ value
precision<-5
arl<-0 # initial value
mu<-edges

# Initial setting of a large and small trial values for L
L.large<-3.2
arl.large<-shew.cc.arl.fun(n.rl = 1000,mu = mu, sig = sig, L = L.large, coef.diss = coef.diss.2, coef.form = NULL, density = D0, start.net = g1, timeslice = 1000, seed = NULL,
    nsim = 1)
L.small<-2.6
arl.small<-shew.cc.arl.fun(n.rl = 1000,mu = mu, sig = sig, L = L.small, coef.diss = coef.diss.2, coef.form = NULL, density = D0, start.net = g1, timeslice = 1000, seed = NULL,
    nsim = 1)

# Obtain L at a desired level of precision
while(abs(arl-arl.set)>precision){
  L<-L.small+(arl.set-arl.small)/(arl.large-arl.small)∗(L.large-L.small) # use a linear interpolation for a faster computation
  ucl.next<-mu + L∗sig
  lcl.next<-mu-L∗sig
  arl<-shew.cc.arl.fun(n.rl = 1000, mu = mu, sig = sig, L = L, coef.diss = coef.diss.2, coef.form = NULL, density = D0, start.net = g1, timeslice = 1000, seed = NULL, nsim = 1)
  if(arl>arl.set){L.large <- L;arl.large < -arl;}else{ L.small<-L; arl.small<-arl;}
}
```

References

Alwan, L. C. (1992). Effects of autocorrelation on control chart performance. *Communications in Statistics, 21*, 1025–1049.

Alwan, L. C., & Roberts, H. V. (1988). Time-series modeling for statistical process control. *Journal of Business & Economic Statistics, 6*, 87–95.

Chien, C. F., Chou, C. W., & Yu, H. C. (2016). A novel route selection and resource allocation approach to improve the efficiency of manual material handling system in 200-mm wafer fabs for industry 3.5. *IEEE Transactions on Automation Science and Engineering, 13*, 1567–1580.

Chien, C. F., Hong, T. Y., & Guo, H. Z. (2017). An empirical study for smart production for tft-lcd to empower industry 3.5. *Journal of the Chinese Institute of Engineers, 40*, 552–561.

Chien, C. F., Lin, Y. S., & Lin, S. K. (2020). Deep reinforcement learning for selecting demand forecast models to empower industry 3.5 and an empirical study for a semiconductor component distributor. *International Journal of Production Research*. https://doi.org/10.1080/00207543.2020.1733125.

Costa, A., & Fichera, S. (2017). Economic statistical design of arma control chart through a modified fitness-based self-adaptive differential evolution. *Computers & Industrial*

Engineering, 105, 174–189.

Dong, H., Chen, N., & Wang, K. (2020). Modeling and change detection for count-weighted multilayer networks. *Technometrics, 62*(2), 184–195.

Farahani, E. M., & Kazemzadeh, R. B. (2019). Phase I monitoring of social network with baseline periods using Poisson regression. *Communications in Statistics - Theory and Methods, 48*, 311–331.

Fotuhi, H., Amiri, A., & Maleki, M. R. (2018). Phase I monitoring of social networks based on Poisson regression profiles. *Quality & Reliability Engineering International, 34*, 572–588.

Frank, O., & Strauss, D. (1986). Markov graphs. *Journal of the American Statistical Association, 81*, 832–842.

Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M., Krivitsky, P. N., & Morris, M. (2018). ergm: Fit, simulate and diagnose exponential-family models for networks. *The Statnet Project. R package version, 3*(9), 4.

He, Q. P., Wang, J. (2017). Statistical process monitoring in the era of smart manufacturing. In: *2017 American Control Conference (ACC)* (pp. 4797–4802).

He, Z., Wang, Z., Tsung, F., & Shang, Y. (2016). A control scheme for autocorrelated bivariate binomial data. *Computers & Industrial Engineering, 98*, 350–359.

Hosseini, S. S., & 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 & Reliability Engineering International, 34*, 477–500.

Hsu, C. Y., Chen, W. J., & Chien, J. C. (2020). Similarity matching of wafer bin maps for manufacturing intelligence to empower industry 3.5 for semiconductor manufacturing. *Computers & Industrial Engineering*. https://doi.org/10.1016/j.cie.2020.106358.

Hunter, D. R. (2007). Curved exponential family models for social networks. *Social Networks, 29*, 216–230.

Hunter, D. R., Handcock, M. S., Butts, C. T., Goodreau, S. M., & Morris, M. (2008). ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software, 24*, 1–29.

Jamrus, T., Wang, H. K., & Chien, C. F. (2020). Dynamic coordinated scheduling for supply chain under uncertain production time to empower smart production for industry 3.5. *Computers & Industrial Engineering*. https://doi.org/10.1016/j.cie.2020.106375.

Khedmati, M., & Niaki, S. T. A. (2016). Monitoring simple linear profiles in multistage processes by a MaxEWMA control chart. *Computers & Industrial Engineering, 98*, 125–143.

Komolafe, T., Quevedo, A. V., Sengupta, S., & Woodall, W. H. (2019). Statistical evaluation of spectral methods for anomaly detection in networks. *Network Science, 7*, 319–352.

Krivitsky, P. N., & Goodreau, S. M. (2019). Stergm – separable temporal ergms for modeling discrete relational dynamics with statnet. http://www.vps.fmvz.usp.br/CRAN/web/packages/tergm/vignettes/STERGM.pdf (June 11, 2019).

Krivitsky, P. N., & Handcock, M. S. (2014). A separable model for dynamic networks. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 76*, 29–46.

Krivitsky, P. N., & Handcock, M. S. (2018). tergm: Fit, simulate and diagnose models for network evolution based on exponential-family random graph models. *The Statnet Project. R package version, 3*(5), 2.

Ku, C. C., Chien, C. F., & Ma, K. T. (2020). Digital transformation to empower smart production for industry 3.5 and an empirical study for textile dyeing. *Computers & Industrial Engineering*. https://doi.org/10.1016/j.cie.2020.106297.

Leifeld, P., Cranmer, S., & Desmarais, B. (2018). Temporal exponential random graph models with btergm: Estimation and bootstrap confidence intervals. *Journal of Statistical Software, 83*, 1–36.

Maleki, M. R., Amiri, A., & Castagliola, P. (2018). An overview on recent profile monitoring papers (2008–2018) based on conceptual classification scheme. *Computers & Industrial Engineering, 126*, 705–728.

Maragah, H. D., & Woodall, W. H. (1992). The effect of autocorrelation on the retrospective X-chart. *Journal of Statistical Computation & Simulation, 40*, 29–42.

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

Megahed, F. M., & Jones-Farmer, L. A. (2015). Statistical perspectives on big data. In S. Knoth, & W. Schmid (Eds.). *Frontiers in statistical quality control 11* (pp. 29–47). Cham: Springer International Publishing.

Montgomery, D. C. (2009). *Introduction to Statistical Quality Control* (6th ed.). Hoboken, NJ: John Wiley & Sons.

Morris, M., Handcock, M., & Hunter, D. (2008). Specification of exponential-family random graph models: Terms and computational aspects. *Journal of Statistical Software, 24*, 1–24.

Noorossana, R., & Vaghefi, S. J. M. (2006). Effect of autocorrelation on performance of the MCUSUM control chart. *Quality & Reliability Engineering International, 22*, 191–197.

Page, E. S. (1954). Continuous inspection schemes. *Biometrika, 41*, 100–115.

Park, Y., Priebe, C. E., & Youssef, A. (2013). Anomaly detection in time series of graphs using fusion of graph invariants. *IEEE Journal of Selected Topics in Signal Processing, 7*, 67–75.

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., & Park, Y. (2005). Scan statistics on Enron graphs. *Computational & Mathematical Organization Theory, 11*, 229–247.

Roberts, S. W. (1959). Control chart tests based on geometric moving averages. *Technometrics, 1*, 239–250.

Robins, G., Pattison, P., Kalish, Y., & Lusher, D. (2007). An introduction to exponential random graph (p*) models for social networks. *Social Networks, 29*, 173–191.

Sengupta, S., & Woodall, W. H. (2018). Discussion of ”statistical methods for network surveillance”. *Applied Stochastic Models in Business and Industry, 34*, 446–448.

Snijders, T. A. B., Pattison, P. E., Robins, G. L., & Handcock, M. S. (2006). New specifications for exponential random graph models. *Sociological Methodology, 36*, 99–153.

Soleimani, P., Noorossana, R., & Amiri, A. (2009). Simple linear profiles monitoring in the presence of within profile autocorrelation. *Computers & Industrial Engineering, 57*, 1015–1021.

Vanhatalo, E., & Kulahci, M. (2015). The effect of autocorrelation on the Hotelling T^2 control chart. *Quality & Reliability Engineering International, 31*, 1779–1796.

Wang, H., Tang, M., Park, Y., & Priebe, C. E. (2014). Locality statistics for anomaly detection in time series of graphs. *IEEE Transactions on Signal Processing, 62*, 703–717.

Wang, Y. H. T., & Huang, W. H. (2017). Phase II monitoring and diagnosis of autocorrelated simple linear profiles. *Computers & Industrial Engineering, 112*, 57–70.

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

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

Yin, S., & Kaynak, O. (2015). Big data for modern industry: Challenges and trends [point of view]. *Proceedings of the IEEE* (pp. 143–146). .

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

Zhao, M. J., Driscoll, A. R., Sengupta, S., Fricker, R. D., Jr., Spitzner, D. J., & Woodall, W. H. (2018). Performance evaluation of social network anomaly detection using a moving window-based scan method. *Quality & Reliability Engineering International, 34*, 1699–1716.

Zou, N., & Li, J. (2017). Modeling and change detection of dynamic network data by a network state space model. *IISE Transactions, 49*, 45–57.