Adaptive Sampling and Quick Anomaly Detection in Large Networks

Xiaochen Xian, *Member, IEEE*, Alexander Semenov, Yaodan Hu, *Member, IEEE*, Andi Wang, and Yier Jin, *Senior Member, IEEE*

***Abstract*— The monitoring of data streams with a network structure have drawn increasing attention due to its wide applications in modern process control. In these applications, high-dimensional sensor nodes are interconnected with an underlying network topology. In such a case, abnormalities occurring to any node may propagate dynamically across the network and cause changes of other nodes over time. Furthermore, high dimensionality of such data significantly increased the cost of resources for data transmission and computation, such that only partial observations can be transmitted or processed in practice. Overall, how to quickly detect abnormalities in such large networks with resource constraints remains a challenge, especially due to the sampling uncertainty under the dynamic anomaly occurrences and network-based patterns. In this paper, we incorporate network structure information into the monitoring and adaptive sampling methodologies for quick anomaly detection in large networks where only partial observations are available. We develop a general monitoring and adaptive sampling method and further extend it to the case with memory constraints, both of which exploit network distance and centrality information for better process monitoring and identification of abnormalities. Theoretical investigations of the proposed methods demonstrate their sampling efficiency on balancing between exploration and exploitation, as well as the detection performance guarantee. Numerical simulations and a case study on power network have demonstrated the superiority of the proposed methods in detecting various types of shifts.**

***Note to Practitioners*—Continuous monitoring of networks for anomalous events is critical for a large number of applications involving power networks, computer networks, epidemiological surveillance, social networks, etc. This paper aims at addressing the challenges in monitoring large networks in cases where monitoring resources are limited such that only a subset of nodes in the network is observable. Specifically, we integrate network structure information of nodes for constructing sequential detection methods via effective data augmentation, and for designing adaptive sampling algorithms to observe suspicious nodes that are likely to be abnormal. Then, the method is further generalized to the case that the memory of the computation is also constrained due to the network size. The developed method is greatly beneficial and effective for various anomaly patterns, especially when the initial anomaly randomly occurs to nodes in the network. The proposed methods are demonstrated to be capable of quickly detecting changes in the network and dynamically changes the sampling priority based on online observations in various cases, as shown in the theoretical investigation, simulations and case studies.**

***Index Terms*— Adaptive sampling, anomaly detection, network, statistical process control.**

I. Introduction

SEQUENTIAL anomaly detection under network structure has raised increasing attention to ensure system stability and safety in many modern applications. With the rapid development of sensor and communication techniques, the sensing and monitoring of big data streams in large network settings became available for quality and reliability insurance. For instance, massive access points need to be simultaneously monitored in large-scale power networks [1], [2], [3] for tasks like line outage detection and intrusion detection. Other examples of such networked anomaly detection include detecting rumor spread in social networks [4], epidemic detection [5], and detection of malicious code spreading in computer networks [6]. All of these applications involve monitoring massive series of real-time and sequentially ordered observations collected from high-dimensional connecting nodes, where the networks typically involve certain underlying topology instead of distance-based [7], [8], [9], [10]. When abnormality occurs to any node, it propagates dynamically across the network, affects more and more nodes, and changes their associated data streams over time. The goal is to detect this change as quickly as possible subject to false alarm constraints.

Although the monitoring of networks has been studied in latest literature [11], [12], these methods are rooted in the assumption that data from all the nodes are fully observable and can be processed in real time. However, when applied to a very large network, such an assumption requires significant data acquisition, transmission, and processing resources that

Manuscript received 12 May 2022; revised 5 August 2022; accepted 2 October 2022. This article was recommended for publication by Associate Editor Z. Kong and Editor J. Li upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation under Grant 2032734 and Grant 1818500 and in part by the Florida Center for Cybersecurity under Grant 3910-1011-03-A. *(Corresponding author: Xiaochen Xian.)*

Xiaochen Xian is with the Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: xxian@ufl.edu).

Alexander Semenov is with the Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: asemenov@ufl.edu).

Yaodan Hu is with the Department of Electrical and Computer Engineering, Idaho State University, Pocatello, ID 83201 USA (e-mail: cindyhu@isu.edu).

Andi Wang is with the School of Manufacturing Systems and Networks, Arizona State University, Mesa, AZ 85212 USA (e-mail: andi.wang@asu.edu).

Yier Jin is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: yier.jin@ece.ufl.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2022.3214193.

Digital Object Identifier 10.1109/TASE.2022.3214193

are hardly available in practice [13], [14], [15]. For example, when detecting rumor spread in social networks, it is not feasible to continuously monitor the activities of all the users in the social network as the processing capability only allows a certain quantity of information to be evaluated in real time. Similar concerns arise in prohibitive costs of sensors and power consumption during the data acquisition and transmission processes. With the resource constraint, we are forced to monitor a subset of nodes in the network, and adaptive sampling approaches are required.

The objective of this paper is to integrate adaptive sampling in online monitoring of dynamic abnormalities in large and arbitrary networks in resource constrained scenarios. We introduce network measures into the monitoring and sampling framework for quick detection and balance between exploration and exploitation, considering two scenarios: (i) for local or regional networks when the number of nodes can be handled in computation memory and (ii) for global network in which linear computation complexity of node number poses a challenge. The contributions of the paper are detailed as follows.

- A change detection and spatial sampling algorithm is proposed on large and arbitrary networks with limited resources. The proposed method incorporated network centrality information [16] to address the challenge of handling the network topology to ensure the sampling performance.
- The work is the first to incorporate memory constraints in the problem of interest. The memory constraint makes the sampling and detection especially challenging due to possible information loss. The proposed adaptive method not only addresses the challenge of the spatial sampling, but also the adaptive memory set updating.
- The proposed methods are investigated theoretically and also evaluated thoroughly under various conditions.

The remainder of the paper is organized as follows. Section II reviews the related literature on network monitoring and adaptive sampling. Section III introduces the general monitoring and adaptive sampling algorithm for networks where resource constraints exist. Section IV further extends the method to the case where the memory capacity also poses a challenge. Section V details the theoretical properties of the algorithms. Sections VI and VII evaluate and validate the algorithms under various scenarios and the application of power network monitoring. Section VIII finally concludes the article.

II. Literature Review

There are two types of adaptive sampling approaches in statistical process control. One is the adaptive sampling in the time domain, including the variable sample size and variable sample interval approach [17]. Another approach is the spatial adaptive sampling, which adaptively takes observations at each sampling time from a large number of measurements that are distributed spatially [14], [18], [19], [20]. The spatial adaptive sampling approach tackles the online monitoring problem with resource constraints. Given online observations and inferred likelihood of abnormalities, this category of work intelligently employs the monitoring resources to informative locations [18], [19]. While the concept has been proved successful in monitoring big data [14], [20], few existing work has been applied on monitoring data streams with embedded network structures. Though Wang et al. [19] briefly touched this topic, the study is specially designed for 2D grids and cannot be applied to general network with arbitrary topology. This is not a trivial extension as network topology has a strong impact on the monitoring and sampling performance.

Woodall et al. [21] gives an overview of network monitoring and anomaly detection methods. Although both the network monitoring problems and our problem of interest involve a graph, the existing network monitoring papers, such as Azarnoush et al. [22], focus on monitoring edge weights, while we focus on monitoring values on the nodes. Unlike monitoring unstructured data, monitoring values generated from nodes of a network requires special considerations, as abnormalities may propagate dynamically according to the network topology, where the dependence among nodes may inform the design monitoring and adaptive sampling strategy. In the literature, abnormality propagation in networks is typically modeled in cascading fashion, with examples of spread of epidemic infections over the contact networks or spread of information in the social networks. It has been observed that the underlying network characteristics [23] and the initial abnormal nodes (which are typically referred to as seeds) [24] play a crucial role in characteristics of the abnormality propagation and the corresponding detection. Besides, existing monitoring and adaptive sampling works assume sufficient processing capabilities linear to the number of data streams, which may be violated in practice for large-scale networks.

To address the issue of resource constraints in network monitoring problems, there has been literature on sensor placement [1], [4] discussing the optimal placement of monitoring resources to minimize detection delay. The key idea of this category of research is to place the sensors on "important" nodes in the network. With similar considerations, in the social networks, there is a large body of research on influence maximization algorithms that aim to maximize the spread of the information [25], [26], [27], [28], [29], resulting from diffusion-driven information transfer among individuals in the social network. However, the design of these methods primarily focuses on the network structure and prior knowledge of abnormalities. When such prior knowledge is unknown, their performance may exhibit large variability in online detection as abnormalities may occur to any nodes with stochastic patterns unknown beforehand. Some literature such as the preprint Zhang and Mei [30] uses multi-armed bandit formulation to describe the data streams monitoring problem with partial observations. However, there are a few key differences between the bandit setting and the setting considered in our paper. First, it requires a specification of prior distribution that describes the believes of the out-of-control distribution and changepoint. Wrong specification of these believes will compromise the performance. However, such information is hard to find in practice. Second, the objective of multi-armed bandit is to minimize the regret, and thus focus on the performance of the estimation in long term.

However, we focus on quickly detecting the out-of-control scenario. As an additional challenge, when the network is extremely large, a large quantity of data concerning all nodes need to be kept in memory, which will pose huge demand on hardware memory. In cases where the memory constraint presents, existing methods cannot be applied directly. This challenge has been noticed in existing studies on online data analysis [31], [32], [33], but has not been tackled for large-scale adaptive sampling and anomaly detection.

III. Monitoring and Adaptive Sampling for Networks

A detailed mathematical formulation of the network to be monitored is as follows. We denote the network of interest as a graph $G = (V, E)$ with a set of n nodes $V = \{1, \cdots, n\}$ ($|V| = n$ and $|\cdot|$ denotes the cardinality of a set) and a set of edges E. Neighborhood $\mathcal{N}(v)$ of a node v is a set of nodes v' adjacent to v, i.e. $v' \in \mathcal{N}(v)$ for all $(v, v') \in E$. Then, the $degree$ (or degree centrality) of v, $deg(V) = |\mathcal{N}(v)|$. A path between nodes i and j is a set of nodes $\{v_0, \ldots, v_d\} \subseteq V$ such that $v_0 = i$, $v_d = j$, and $(v_k, v_{k+1}) \in E$, $\forall k = 0, \ldots, d-1$. The shortest path between nodes i and j is the path containing the least number of edges among all paths between i and j. The length of the shortest path between i and j in G is referred to as the distance between i and j in G and is denoted by $d(i, j)$.

Let $\mathbf{X}(t) = (X_1(t), \cdots, X_n(t))$ denote the measurement values associated with the n nodes at each time t. Due to resource constraints, only q ($q < n$) out of n nodes are observable at each time t. Denote the sets of observable variables at time t as $O(t)$. Consequently, for any time t $|O(t)| = q$. At a random and unknown change point τ, an anomaly appears in the network, and affects nodes in the network. Without loss of generality, in this paper we adopt the Independent Cascade (IC) model as the abnormality propagation model. Under the model assumption, a node has two states, activated and not activated. An active node u is out of control and may activate (spread the abnormality) to its neighbors $\mathcal{N}(v)$ with propagation probability p_{uv}. If a node i is not affected by the anomaly, the observed $X_i(t)$ are independent and identically distributed (i.i.d.) samples from a pre-change distribution f_0 over time; if a node i is affected by the anomaly, then it receives i.i.d. samples from a post-change distribution f_1 [12], [34]. The formulation of the problem is illustrated in Fig. 1. The key question is how to develop an effective adaptive sampling strategy that can intelligently and sequentially determine the sampling density subject to the resource constraints, to quickly detect the assignable causes while maintaining a system-wide in-control ARL requirement. During the monitoring, we assume that the algorithm can sample from any location at consecutive time points without sampling costs.

To quickly detect the abnormality occurred in the network, this paper proposes to use the Cumulative Sum (CUSUM) statistics, an algorithmic solution to minimize the detection delay of abnormality while maintaining the false alarm rate to be under a specified level. Note that such a framework does

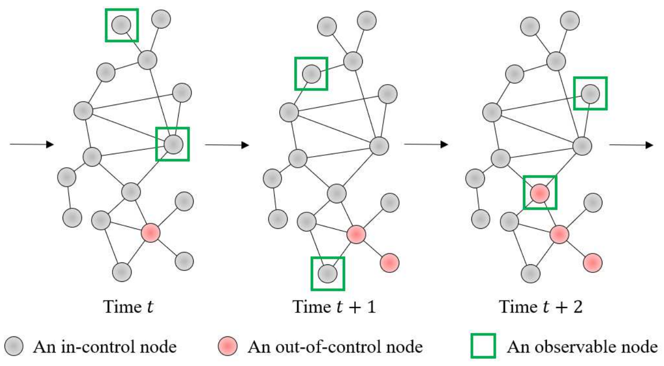


Fig. 1. Illustration of the network anomaly and sampling over time.

not require the observations to follow a specific distribution and thus can be widely applied in practice. Under the full observation scenario, the following local CUSUM statistic shows the likelihood for node i to be OC at time t based on its observations:

$$W_i^F(t) = \max\left(W_i^F(t-1) + \log\frac{f_1(X_i(t))}{f_0(X_i(t))}, 0\right) \tag{1}$$

with $W_i^F(0) = 0$. The superscript F represents the statistics under full observation. However, it is a challenging task extending the mechanism under partial observation scenario. First, when $X_i(t)$ cannot be observed, the CUSUM statistic $W_i^F(t)$ is not well defined. Approaches to systematically monitoring the entire network and infer unobservable nodes leveraging the network structure need investigation. Furthermore, how to design the sampling strategy that ensures an overall effective network monitoring performance is not resolved in the literature. To tackle these challenges, we investigate the monitoring and adaptive sampling strategies leveraging network structure in this section.

A. Local Data Augmentation and Detection Scheme

We first establish a network monitoring framework under partial observations. To estimate the status of all nodes based on partial observations, inspired by existing adaptive sampling algorithms [19], [20], we introduce the data augmentation methods to infer the out-of-control likelihood of the unobservable nodes based on online partial observations and the network structure. Data augmentation refers to a type of methods via adding information or latent variables to the originally "unobservable" or "missing" data, such that the problem becomes tractable [20], [35]. The central idea here is to utilize kernel function to characterize the dependency of the status among neighboring nodes in the network. Specifically, we estimate the out-of-control likelihood of a node i at time t according to the following equation:

$$\sum_{j \in O_t} K_h(d(i, j)) \log\frac{f_1(X_i(t))}{f_0(X_i(t))} \tag{2}$$

where $K_h(\cdot)$ denotes a kernel function that establishes the dependency among nodes close with regards to selected distance function $d(i, j)$, such that an observable node can "share" its acquired information. To highlight our main idea, we choose the popular Epanechnikov form of the kernel function $K_h(d(i,j)) = \max\left(1-\left(\frac{d(i,j)}{h}\right)^2, 0\right)$ where h is the bandwidth parameter here. $d(i,j)$ is the distance between nodes i and j.

Therefore, according to (2), when an out-of-control node is observed, the out-of-control likelihood of its neighboring nodes will also likely increase. This mechanism matches with our assumption that faults may affect the status of the variables along the network structure, and allows to exploit limited partial online observations for quick change detection. Note that if the bandwidth parameter h is chosen as a small number that $h \leq 1$, we will have $K_h(d(i,j)) = \mathbb{I}(i=j)$ given that $d(i,j)$ is an integer. Here $\mathbb{I}(\cdot)$ is the indicator function. In this case, (2) degenerates to likelihood at time t for conventional CUSUM statistics similar to (1), such that observations of a node is only utilized to update its own CUSUM statistics. If h is chosen as a very large number greater than the diameter of the graph, the observations of all nodes are used to update the statistic of each node. Overall, h is recommended to be the diameter of the out-of-control region of the network, which can be determined based on domain knowledge.

Based on the data augmentation, we modify the CUSUM statistic in the partially observed network setting as follows

$$W_i(t) = \max\left(W_i(t-1) + \sum_{j\in\mathcal{O}_t} K_h(d(i,j)) \log\frac{f_1(X_j(t))}{f_0(X_j(t))}, 0\right). \tag{3}$$

where $W_i(t)$ is the CUSUM statistic for node i at time t that demonstrates the cumulative out-of-control likelihood "locally" associated with node i up to time t. At each data acquisition time t, the local statistic $W_i(t)$ is only updated when there is an observable node within distance h in the network. The higher $W_i(t)$ is, the larger out-of-control likelihood node i has. To determine the overall system status, we construct the global monitoring statistic

$$W(t) = \max_{1\leq i\leq n} W_i(t), \tag{4}$$

which is the maximum of all local statistics [36]. Due to the data augmentation mechanism shown in (2), the spatially clustered out-of-control scenario in a network will lead to quick increase of local CUSUM statistic for nodes amid the clustered fault, which ensures the general efficiency of the global monitoring statistic. If the overall monitoring statistic $W(t)$ is greater than a threshold value, i.e., $W(t) > L$, we claim that the overall network is out of control. L is the monitoring threshold based on a prescribed in-control average run length (ARL) requirement. In practice, we determine the value of L by simulation based on large quantities of in-control data, the details of which are shown in Appendix A.

B. Adaptive Sampling Strategy

To ensure quick detection of abnormalities in the network setting, a critical precondition is that out-of-control nodes can be timely observed and kept under surveillance such that the detection framework in Section III-A quickly triggers an alarm. However, this is not an easy task and requires strategic design of adaptive sampling. Intuitively, the desired adaptive sampling strategy will need to balance between exploration and exploitation. When the system is in control (particularly when the observed nodes exhibit in control), we expect to explore other nodes to ensure no abnormalities occur elsewhere; when abnormalities occur, we expect to keep observing out-of-control nodes and their neighbors to reduce detection delay. We aim to design an integrated strategy that can dynamically switch between these two sampling considerations given online observations.

In this subsection, we will extend the adaptive sampling algorithm for online monitoring to the network scenario [19]. Given that q nodes are observable online, we separate the monitoring resources into two sets based on the two desired considerations: (i) A portion of the resources are allocated based on the "Breadth-First Sampling" (BFS) consideration that explores the entire network to locate the clusters of out-of-control nodes in a more stochastic fashion. We denote the number of nodes observed by this mechanism as $q_B(t)$ at time t. (ii) The rest of the monitoring resources are assign based on the "Depth-First Sampling" (DFS) consideration to exploit the most suspicious nodes. We denote the number of nodes observed by this mechanism as $q_D(t)$ at time t. Apparently, $q_B(t)+q_D(t)=q$ is a constant limited by resource constraint. The superiority of the proposed adaptive sampling method lies in the dynamic balancing between exploration and exploitation by adjusting $q_B(t)$ and $q_D(t)$ based on online assessment of system status. If the system is likely to be in control, we will emphasize more on exploration and increase $q_B(t)$; otherwise when the observed nodes are suspicious, we increase $q_D(t)$ to facilitate a large number of observations taken on the suspicious nodes and their neighbors. Below we introduce these methods in detail to determine $q_B(t)$ and $q_D(t)$ as well as the corresponding layouts.

First, we dynamically update $q_B(t)$ and $q_D(t)$ based on the overall system status. To carry out the aforementioned DFS mechanism, we establish the relationship between $q_D(t)$ and the monitoring statistic $W(t)$ that

$$q_D(t) = \min\left(\left\lceil \frac{q\lambda \max(W(t)-D\theta, 0)}{D(1-\theta)} \right\rceil, q\right). \tag{5}$$

In this equation, $q_D(t)$ is determined by the expression in the ceiling function denoted by $\lceil\cdot\rceil$, and capped by the total number of observable nodes q. The expression inside the ceiling function is a linear function of the charting statistic $W(t)$ given that $W(t)$ is greater than $D\theta$. The ceiling function then maps the value of this expression to the smallest following integer to give a valid number of q_D nodes. Inside the ceiling function, $\lambda, \theta \in [0,1]$ and D are the parameters designed to control the sampling performance. Intuitively, $q\lambda$ is the maximum designed value of $q_D(t)$; D represents the threshold of $W(t)$ for which $q_D(t)$ reaches its maximum designed capacity $q\lambda$; and $D\theta$ is the threshold of $W(t)$ for which $q_D(t)$ starts to be non-zero. In practice, D should be close to the alarm threshold L, such that the algorithm utilizes its full exploitation capability when there is sufficient evidence showing the network is out of control. Based on (5), $q_D(t)$ is a non-decreasing function of the monitoring statistic as planned in the depth-first sampling consideration. The number of nodes observed by the BFS mechanism $q_B(t)$ is then determined by $q_B(t) = q - q_D(t)$.

Second, we elaborate on the methods to determine the sampling layout. The DFS mechanism targets at and repetitively evaluates suspicious nodes, and thus we propose to update the sampling layout as the top $q_D(t)$ nodes with the largest local statistics $W_i(t)$. Mathematically, we denote the permutation of $\{W_i(t)\}_{i=1}^n$ in descending order as $\{W_{i_{(j)}(t)}(t)\}_{j=1}^n$, such that $W_{i_{(1)}(t)}(t) \geq W_{i_{(2)}(t)}(t) \geq \cdots \geq W_{i_{(n)}(t)}(t)$. At time $t+1$, we monitor the nodes $\{i_{(j)}(t)\}_{j=1}^{q_D(t)}$. Due to the data augmentation in Section III-A, the sampled nodes under DFS are likely to be out-of-control nodes and their neighbors. We will further show in Section V that such a sampling strategy will likely stick to the out-of-control nodes over time. On the contrary, the BFS mechanism aims at exploring the network in a more stochastic manner to identify new out-of-control nodes. In graph theory, centralities characterize the importance of nodes within a network corresponding to their network position. Besides degree centrality introduced before, many centrality measures have been used to identify important nodes in the networks [16]. For example, closeness centrality $C_C(i)$ of node i is reciprocal of the average shortest distance from i to any other node $j \in V$,

$$C_C(i) = \frac{|V| - 1}{\sum\limits_{j \in V \setminus v} d(i, j)} \tag{6}$$

Betweenness centrality $C_B(i)$ of i is the fraction of shortest paths connecting any two other nodes j and j' that pass through i,

$$C_B(i) = \sum_{j, j' \in V} \frac{\sigma_{j,j'}(i)}{\sigma_{j,j'}} \tag{7}$$

where $\sigma_{j,j'}$ is number of shortest paths between j and j', and $\sigma_{j,j'}(i)$ is the number of those shortest paths that pass through node i. We will also use other centralities (such as LocalRank and PageRank) in later sections to evaluate the monitoring and sampling performance of the proposed algorithms. Nodes with higher centralities are better connected or involved in the network, and thus may be more prone to be affected by cascading faults. Therefore, to increase the detection capability, we sample among all nodes not observed by DFS, with probability $\rho(i)$ for node i proportional to a centrality $C(i)$, i.e., $\rho(i) \propto C(i)$. In Section VI, we will experiment with various choices of centralities to compare their impacts on sampling performances. Besides betweenness, and closeness centralities, we have used LocalRank [37], and PageRank of the nodes.

IV. Extension to Large Networks With Memory Constraint

Although the proposed methodology described in Section III has significantly reduced the number of observations to be collected, transmitted, and processed, its computation complexity is linear in the number of nodes n in the network in the sense that local statistics of all nodes need to be stored. Therefore, when the network is extremely large, such a method will pose huge demand on hardware memory. This challenge has been noticed in existing studies on online data analysis [31], [32], [33] but has not been addressed for large-scale anomaly detection. The memory constraint impedes a large number of quantities to be kept in memory and thus the proposed method in Section III cannot be applied directly. To cope with this issue, we further extend the monitoring and adaptive sampling method in this section for extremely large networks with memory constraint.

When the monitoring and sampling strategy is limited by memory capacity, the major impact is that the local statistics $\{W_i(t)\}_{i=1}^n$ cannot be fully saved over time. Therefore, we are forced to keep a partial list of local statistics in memory and part of the historical node behavior will be erased. It should be noted that nodes not in memory at time t will not cause an alarm to be triggered even if it is out-of-control, since no local statistic is associated with it. Similar to the sampling strategy, we expect to keep the local statistics of suspicious nodes in memory such that the identified suspicious nodes can be further investigated for anomaly detection. Meanwhile, we expect the mechanism to explore all nodes such that they may be included in the memory occasionally in case anomalies occur at them. To handle the memory constraint, we denote the set of nodes kept in memory at time t as $\mathscr{P}_t$, and assume that the local statistics of a subset of nodes are available in memory. Without loss of generality, we assume that $|\mathscr{P}_1| = M$, where M is determined by the memory capacity. We then modify the monitoring and sampling procedure in Section III such that the algorithm can be applied to adaptively determine the subset of nodes to be kept and deleted from the memory on top of the monitoring and sampling considerations. Here we present these four key components in the algorithm.

Determining nodes in memory. We start from determining the nodes in $\mathscr{P}_t$, i.e., determining the set of nodes to be removed from memory and nodes to be added to memory. Intuitively, a critical criterion for node informativity is its local statistic that shows the out-of-control likelihood. Therefore, at each time t, we remove nodes with small local statistics from the memory and instead replace with new nodes for quick exploration. Specifically, we propose to remove nodes with local statistics $W_i(t)$ smaller than a threshold value l. Here l is a parameter that acts as the "keep-in-memory" threshold. It should be noted that l should in general satisfy $0 < l < L$. If $l \leq 0$, no nodes will be ever removed from memory and thus the algorithm degenerates to the sampling strategy with a fixed layout over time. Therefore, a very small l may fail to filter an in-control node and thus cause ineffective exploration among nodes. If $l \geq L$, no nodes will be kept in memory and thus the in-memory layout will change every time. Thus, a large l might lead to detection failure for smaller shifts as these nodes will be removed quickly before their local statistics grow. The optimal value of l depends on the steady-state behavior of local statistic $W_i(t)$ and the actual mean shift magnitude, which will be further investigated in Section VI.

If there are nodes removed at time t, to replace the removed nodes, we follow the BFS mechanism in Section III and randomly select new nodes into the memory based on network topology. Specifically, we sample among all nodes in $V \cap \overline{\mathscr{P}_{t-1}}$, with probability $\rho(i)$ for node i proportional to its centrality $C(i)$, i.e., $\rho(i) \propto C(i)$. As a special case, at the initial time $t = 1$, the entire memory set $\mathscr{P}_1$ is randomly selected with probability proportional to centrality $C(i)$, i.e., all M nodes are sampled to keep in the memory.

New node initialization. At each time t, for a node i that is newly added to the memory (i.e., $i \in \mathscr{P}_t \cap \overline{\mathscr{P}_{t-1}}$), we initialize its monitoring statistics such that

$$W_i(t-1) = W_0, \text{for } i \in \mathscr{P}_t \cap \overline{\mathscr{P}_{t-1}}, \tag{8}$$

where W_0 is the initiation parameter. Intuitively, $W_0 > 0$ as a head-start for a node cannot be too small in case the new nodes are immediately deleted. The design is very similar to the fast initial response in the literature [38], [39], [40], such that W_0 gives a head-start to the monitoring statistics of newly added nodes. It is shown in the literature that the fast initial response feature has little effect if the process starts out in control, but permits a more rapid response to an out-of-control situation [38]. This feature is especially useful for our problem of interest as nodes may be frequently moved into the memory as a new node without historical records, and thus the head-start boosts the out-of-control nodes for quick detection. Meanwhile, it has minimum effect on the local statistics of in-control nodes such that their local statistics will drop down to normal. Overall, the new node initialization is critical for quick screening of the entire system under memory constraint. The value of W_0 balances between quick response to out-of-control signals and the drop-off of in-control nodes, which will be investigated further in Section VI. There is a natural constraint for W_0 that it should satisfy $W_0 \geq l$. Otherwise, a newly added node will likely be removed very soon and the exploration of new nodes is impaired.

Sampling strategy. Without loss of generality, we again assume that only partial observations are available due to data acquisition and transmission capabilities. We note that data acquisition and transmission related constraints are in general more restricting than the memory constraints regarding data quantity, as the former is typically more expensive for hardware. Therefore, we assume that the in-memory capacity is no smaller than the number of observable nodes, i.e., $M \geq q$. In this case, the sampling strategy introduced in Section III can be applied similarly under the memory constrained case, with the modification that the observable variables are only selected from $\mathscr{P}_t$ based on the BFS and DFS mechanisms, mathematically, $\mathscr{O}_t \subseteq \mathscr{P}_t$.

Local statistics and monitoring statistics. To update the statistics at time t, (3) can still be applied in the memory constrained scenario for $i \in \mathscr{P}_t$, as $\mathscr{O}_t \subseteq \mathscr{P}_t$. The monitoring statistic $W(t)$ is updated based on only the in-memory nodes that

$$W(t) = \max_{i \in \mathscr{P}_t} W_i(t), \tag{9}$$

To determine the status of the system, we compare $W(t)$ with the monitoring threshold L, and trigger an out-of-control alarm if $W(t) > L$.

The illustration of the mechanism in the memory constrained scenario is shown in Fig. 2. A newly added node will

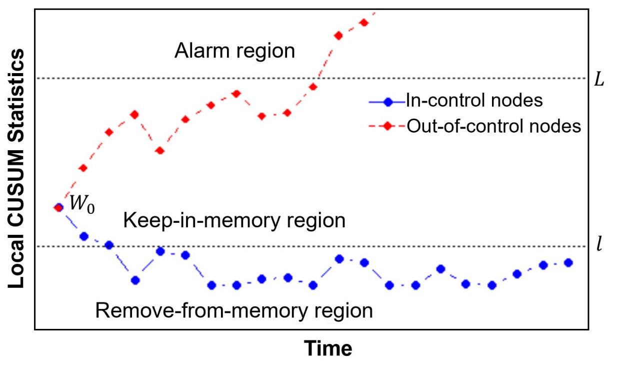


Fig. 2. Illustration of monitoring statistic, W_0, alarm threshold and memory threshold l.

have a head-start with a magnitude of W_0. For an in-control node, its local statistic will quickly drop down below l and thus removed from memory. In contrast, an out-of-control node will instead quickly increase and thus be kept in memory until an out-of-control alarm is triggered.

V. Theoretical Properties

In this section, we investigate the theoretical performance of the proposal algorithms and show the following properties. Without loss of generality, we assume that the variables $\mathbf{X}(t)$ follows a distribution from the exponential family. We denote their in-control and out-of-control probability density functions as $f_0(x) = \exp[\eta(\zeta_0)T(x) - A(\zeta_0) + B(x)]$ and $f_1(x) = \exp[\eta(\zeta_1)T(x) - A(\zeta_1) + B(x)]$. Note that the exponential family contains a wide range of distributions including the normal, exponential, log-normal, gamma, Bernoulli, categorical, and Poisson distributions, which covers a wide range of distributions frequently used for online monitoring. We begin with the theorem that investigates the in-control property of the proposed algorithms.

Theorem 1 (In-Control Sampling Property): Assume that all nodes $i \in V$ satisfy $[\eta(\zeta_1)-\eta(\zeta_0)]\mathbb{E}T(X_i) < A(\zeta_1)-A(\zeta_0)$. Let U denote the set of nodes that there exist a finite time t_0 such that for any $i \in U$, i will never be observed after t_0. Then $P(U = \emptyset) \to 1$ as $L \to \infty$ for $\theta, \lambda \in (0, 1)$.

The assumption $[\eta(\zeta_1) - \eta(\zeta_0)]\mathbb{E}T(X_i) < A(\zeta_1) - A(\zeta_0)$ in this property means that the process exhibits in-control in the sense that f_1 and f_0 are close. In addition, $L \to \infty$ ensures that the monitoring threshold is large enough and thus the process will be run forever without termination. Given these conditions, Theorem 1 shows that no variables will be left unobserved in the long run. This means that the proposed sampling strategy ensures effective exploration of all nodes such that abnormalities occur at any nodes can be identified. Note that the theorem is proved under both scenarios with and without memory constraints in Appendix B.

Theorem 2 (Out-of-Control Sampling Property): Suppose that there is at least one node with a mean shift, and its magnitude δ is large enough such that $[\eta(\zeta_1) - \eta(\zeta_0)]\mathbb{E}T(X_j(t)) - [A(\zeta_1) - A(\zeta_0)] > 0$. In this case, once such a shifted node i is observed at time t, there is a nonzero probability such that the variable will always be observed ever after.

According to Theorem 1, the sampling strategy will continue to sample all nodes until at least one observed variables is out-of-control. Theorem 2 then follows in the out-of-control case, and shows that if the shift is large enough, there is a nonzero probability that the shifted variable will be observed forever. This is a critical property as it ensures continuous assessment of a potential out-of-control node and thus quickly triggering of alarm. The theorem is proved in Appendix C for both cases with or without memory constraints.

Theorems 1 and 2 together shows the balance of and dynamic switch between exploration and exploitation of the proposal algorithms.

Theorem 3 (Upper Bound of Detection Delay): Let γ_i denotes the frequency that node i is observed by the proposed algorithms. The proposed strategies with stopping time described in Eq. (4) satisfy average detection delay

$$\mathbb{E}(T) \leq \min_{i\in V}\left\{\frac{L}{q\gamma_i I(f_{1,i}, f_{0,i})}\right\} + O(1)$$

as the alarm threshold $L \to \infty$, where $f_{0,i}$ and $f_{1,i}$ are the in-control and out-of-control probability density functions of node i respectively, and $I(f_{1,i}, f_{0,i})$ is the Kullback-Leibler information for node i.

Theorem 3 provides a detection performance bound for the proposed algorithms which guarantees the asymptotic detection speed. The proof of the theorem is shown in Appendix D.

Remark 1 (Computation Complexity): The computational complexity for the small and large networks are $O(|V|q + |V| + q\log q)$ and $O(|V| + M\log M + Mq + q\log q)$, respectively.

Each run of the algorithm requires selection of M variables into memory. In the case of large networks, this procedure involves partial sort with complexity $O(|V| + M\log M)$. For small networks, the memory set is equal to the number of nodes in the graph $|V|$ and thus no sorting needed. For calculating W_i for each node i in memory $\mathscr{P}$, two nested loops calculate with regard to q observable nodes for all M nodes in memory, resulting in complexity $O(Mq)$ (or, $O(|V|q)$ for small networks). Lastly, at most q deep variables are selected from the memory with complexity $M+q\log q$. We precompute distances $d(i, j)$ between each pair of nodes resulting in $O(1)$ complexity. For large graphs, number of nodes $|V|$ is much greater than Mq, thus, the complexity would be dominated by $O(|V|+M\log M)$ for large networks, and $O(|V|q)$ for small networks. The flow of the algorithm is shown in Algorithm 1. Note that when $\mathscr{P}_t = V$ Algorithm 1 is for small networks and else for large networks.

Remark 2 (Selection of Parameters): Theoretical investigation of the effects of the parameters of the algorithm on the algorithm's behavior is quite difficult. In essence, the system being monitored is essentially a Markov chain whose state space describes all local variables and the distribution of observable variables. The state space has a very high dimension, and individual parameters (λ, D, θ, h, l, and L) affect the transitional probability of this Markov chain, and how these parameters affect the stopping times and asymptotic behavior of the chain is very complicated. Intuitively, the interpretations of individual parameters are as follows:

Algorithm 1 The Monitoring and Sampling Process

```
Input: Graph $G(V, E)$, Centralities $C$, distance matrix
       $\{d(i, j) : 1 \leq i, j \leq n\}$, sampling constraint $q$,
       memory constraint $M$;
$q_D(1) \leftarrow 0$, $t \leftarrow 1$, randomize the memory set $\mathscr{P}_1$;
while True do
    Determine the $q_B(t) = q - q_D(t)$ BFS nodes by
    sampling among all nodes not observed by DFS with
    probability proportional to a centrality $C(i)$
    (Section III-B);
    Update monitoring statistics $W_i(t)$ for $i \in \mathscr{P}_t$ and
    $W(t)$ according to (3) and (4);
    if $W(t) > L$ then
        The process triggers an alarm;
        Break;
    else
        Update memory set $\mathscr{P}_{t+1}$ by removing nodes with
        $W_i(t) < l$, adding new nodes to the memory by
        sampling among all nodes in $V \cap \overline{\mathscr{P}_t}$ with
        probability proportional to centrality $C(i)$, and
        initialize $W_i(t)$ for newly added nodes based on (8)
        (Section IV);
        Determine $q_D(t)$ based on (5) and the DFS nodes
        by selecting the nodes with the largest local
        statistics $W_i(t)$ (Section III-B);
    $t \leftarrow t + 1$;
```

- λ, D, and θ jointly affect the deployment of the deep sensors. The discussion of these parameters can be found after Eq. (5).
- h influence how much neighbors are affected by each observation. Big h would hinder the exploration and make the monitoring scheme too sensitive to occasional extreme values of observations. If h is too small, the out-of-control node will be easily missed even right after an observation.
- l determines the rate of memory and thus the exploitation on suspicious nodes. Larger l will lead to lower probability on keeping a node under memory and observation.
- As the alarm threshold, L is typically selected based on prescribed average run length.

More discussions of the parameters can be found in Section VI.

VI. Simulation

In this section, we evaluate the proposed algorithms and gain better understanding of parameters via numerical simulation on large networks. The performance of the proposed method will also be compared with several benchmark methods to demonstrate the advantages and special characteristics. In the following subsections, we first introduce the simulation setups and then show the simulation results.

A. Simulation Setup

The overall objective of the simulation is to check how quickly proposed algorithms locate out-of-control nodes in the network and trigger an alarm for the entire system. We experiment with different parameter settings, network structures, different number of seeds, and different centralities to determine the sampling weights for the BFS mechanism. Below we provide the details of the simulation setup.

We run the simulation procedures discussed below on two datasets: synthetic dataset generated by Barabasi-Albert model with $|V| = 1000$, and $|E| = 1000$; and real-world dataset NetHEPT comprised of citation network between authors of "High Energy Physics-Theory" section of arXiv.org from 1991 to 2003. The NetHEPT dataset had $|V| = 15233$, and $|E| = 31398$, we take the largest connected component with $|V| = 6794$, $|E| = 19071$. The NetHEPT dataset is frequently used in influence maximization research [41]. To cope with potential randomness of the simulation from (i) choice of seeds, (ii) the spreading of the cascade (depending on the propagation probability), and (iii) the BFS procedure (sampling randomness, partly because of the selected centrality), we repeat the simulation steps $NumRuns = 5000$ times for each case detailed below. The overview of the simulation steps is outlined in Algorithm 2.

Throughout the section, we assume that variables associated with nodes in a network follow normal distributions with standard deviation 1. Specifically, we assume that the in-control nodes follow normal distributions with mean 0. Out-of-control nodes have mean shifts with magnitudes of $\delta = 1, 2,$ and 3. Specifically, we utilize the independent cascade model such that the shift starts from $NumSeeds$ seeds (i.e., initial infected nodes) with an activation probability p_{uv} to infect their adjacent nodes to be constant across all edges (u, v); further we denote it as p_a. The number of seeds $NumSeeds$ ranges from 0.5% to 2% of the total number of nodes in the network, and propagation probability ranges from 0.1 to 0.8. Meanwhile, we consider two mechanisms of the selection of seeds. The first one is to randomly select nodes as seeds, which is referred to as the *random* seed case. This is a practical case where the initial faults may randomly occur, i.e., mechanical failures in a power grid. The second one is to select nodes that have the largest values of degree centrality, which we later refer to as the *degree* seed case. It is a less common case that initial faults only occur to nodes with highest centrality, but it holds for certain cases in practice, e.g., sabotage in a connected IoT system by an intruder who intends to maximize the influence. Although it is a relatively extreme case, we test under this case for thorough performance assessment.

For each case we consider, we report the average run length (ARL) and standard deviation of run lengths for the out-of-control cases, which shows the statistical property of detection delay. In-control average run length (denoted as ARL_0) for all cases are set as 370, and all results regarding average run length are reported based on 5000 simulation runs. We consider three benchmark methods to be compared with the proposed methods: (i) the CUSUM procedure with full observations, which is referred to as the "FULL" method later; (ii) the CUSUM procedure that applies to a fixed set of nodes that has the highest centrality, which is referred to as the "Fixed" method; and (iii) the "TRAS" method proposed in

Algorithm 2 Simulation Steps

```
Input: Graph $G(V, E)$
Output: $Run\_length\_statistics$
$Centralities \leftarrow Compute\_all\_centralities(G)$
$Run\_length\_statistics \leftarrow [\ ]$
for $InfectionProbability \leftarrow 0.1$ to $0.8$ do
    for $NumSeeds \leftarrow 0.5\%$ to $2\%$ do
        $S \leftarrow Generate\_seeds(G, NumSeeds)$
        for $j \leftarrow 0$ to $|Centralities|$ do
            $temporaryResult \leftarrow [\ ]$
            for $step \leftarrow 0$ to $NumRuns$ do
                $temp \leftarrow simulate(G, S, Centralities[j])$
                #network monitoring and sampling process
                $temporaryResult \leftarrow append(temp)$
            $Run\_length\_statistics[NumSeeds, j] \leftarrow$
            $append(temporaryResult)$
return $Run\_length\_statistics$
```

Liu et al. [14], which is a baseline adaptive sampling method without considering the network structure. For the proposed method, we let $\theta = 0.3$, $\lambda = 0.4$, and $h = 2$. Here we choose h as a small number to demonstrate the robustness of the bandwidth parameter to the size of the network. For TRAS, we set $r = 1$ and $\Delta = 0.1$.

B. Simulation Results

We first evaluate the impact of the unique parameters of the proposed method and provide insights on parameter selection. We first test the unique parameters W_0 and l in the proposed method in Section IV. Specifically, we conduct the study on simulated network with $NumSeeds = 10$, random seed, $p_a = 0.5$, and the betweenness centrality is used. The results are shown in Table I that reports the ARL_1 values (i.e., detection delay) under various magnitude of mean shifts. In this table, W_0 ranges from 1 to 2, l ranges from 0.5 to 1.5, and $W_0 > l$. We can observe that the performance of the propose method is fairly robust with regard to the parameter combinations. For large shift ($\delta = 3$), smaller W_0 and larger l in general lead to quicker detection. This is because they result in smaller head-start values and quicker removal of in-control nodes in the memory, such that noticeable shifted nodes can be quickly observed and thus identified. For small shift ($\delta = 1$), in general a larger W_0 value results in a better performance. It will give newly observed nodes a quicker initial response such that nodes with less noticeable shifts can be observed longer. Meanwhile, the value of l cannot be too large, as it may lead to early filtering of nodes with small shifts. Overall, the combination $W_0 = 1.5$ and $l = 1$ gives a satisfactory performance for all magnitudes of shifts considered. Therefore, these values are adopted in the rest of the section.

Recall that the proposed methods utilized centralities of nodes for effective sampling and quick identification of out-of-control nodes. We here compare various choices of centralities to understand the impact of the used centrality on the monitoring and sampling of the proposed method in Section IV.

TABLE I

ARL_1 AND THE CORRESPONDING STANDARD ERRORS (IN PARENTHESES) FOR THE PROPOSED METHOD UNDER COMBINATIONS OF W_0 AND l

W_0	l	$\delta = 1$	$\delta = 2$	$\delta = 3$
1	0.5	11.42 (0.23)	6.31 (0.09)	5.05 (0.06)
1.5	0.5	11.56 (0.30)	6.37 (0.10)	5.03 (0.04)
1.5	1	10.99 (0.20)	6.31 (0.09)	5.02 (0.04)
2	0.5	11.34 (0.24)	6.42 (0.10)	5.14 (0.06)
2	1	11.25 (0.24)	6.32 (0.09)	5.11 (0.05)
2	1.5	11.30 (0.23)	6.32 (0.08)	5.10 (0.05)

TABLE II

ARL_1 AND THE CORRESPONDING STANDARD ERRORS (IN PARENTHESES) FOR THE PROPOSED METHOD UNDER VARIOUS CENTRALITIES

Centrality	Seed type	$\delta = 1$	$\delta = 2$	$\delta = 3$
Closeness	random	137.55 (1.94)	59.08 (0.84)	40.45 (0.58)
	degree	28.52 (0.30)	11.42 (0.10)	7.76 (0.06)
Degree	random	188.14 (3.09)	98.70 (1.67)	67.93 (1.19)
	degree	12.51 (0.09)	5.43 (0.03)	3.95 (0.02)
LocalRank	random	116.13 (1.75)	37.36 (0.51)	23.13 (0.31)
	degree	12.11 (0.09)	5.26 (0.02)	3.84 (0.02)
Pagerank	random	183.82 (2.98)	90.46 (1.54)	61.29 (1.04)
	degree	13.45 (0.10)	5.78 (0.03)	4.16 (0.02)
Betweenness	random	282.49 (4.35)	233.04 (3.79)	215.71 (3.60)
	degree	7.51 (0.04)	3.62 (0.01)	2.80 (0.01)

Table II shows the ARL_1 and corresponding standard errors for the proposed method applied on the simulated network with $NumSeeds = 5$ and $p_a = 0.1$. Five centralities are considered under both random and degree seed scenarios. For all cases, the detection delay decreases as the magnitude of mean shifts increase, as expected. For the two mechanisms of seed selection, we can see that the detection results are very different for each centrality and are not consistent over the choices of centralities. Overall, the detection for degree seed is much faster than random seed, as the degree seed will naturally affect more nodes in the network given that the seeds are highly connected with other nodes. For random seed, the local rank centrality shows the best performance. This is because local rank is a "local" measure and is based on the counts of first and second hop neighbors [37]. Betweenness centrality leads to the worst performance for random nodes as it is the highest for the nodes that lie on the intersection of many shortest paths; these nodes may not necessarily be important for the networks without pronounced community structure. On the contrary, for degree seed, betweenness centrality performs the best, because nodes with high betweenness could be close to the "superseeds" with high degree. Closeness centrality leads to the slowest detection, as closeness centrality may be highly affected by nodes with less neighbors and thus does not match with degree seeds. Overall, the local rank centrality shows a robust performance for both cases and all magnitudes of mean shifts. Therefore, we use the local rank centrality for the results later in this section.

Next, we compare the proposed method with the three benchmark methods for comprehensive evaluation and validation. We assume that $q = 10$ in the comparisons, which means the monitoring resources are extremely limited for online monitoring. We start with the comparison for the proposed method in Section III without memory constraint in the degree seed scenario, conducted on the simulated network. The results are shown in Table III. In this table, we consider various number of seeds $NumSeeds$, activation probability p_a, and magnitudes of mean shifts. First of all, it is obvious that the Fixed method performs well for small to moderate shifts. This observation is consistent with our expectation as the Fixed method acts like an oracle method that monitors only the seed nodes. The proposed methods show comparable performances in most cases, and even outperforms the Fixed method for large shifts or large $NumSeeds$ due to the augmentation mechanism. It allows the proposed method to leverage the network structure and exploit observations of neighboring modes, which leads to quicker detection when larger shifts or more out-of-control nodes are observed. The proposed method also consistently outperforms the TRAS method. When the monitoring resources are highly limited, the TRAS method heavily relies on random sampling. Therefore, it takes longer for TRAS to trigger an alarm than the proposed method which leverages network structure and centrality. Although the Full method has perfect information of the entire network, it performs comparable or even worse than the proposed method at times. It makes sense in the degree seed scenario where more nodes will be affected, given that the probability for adaptive sampling methods to locate an out-of-control node is higher. Furthermore, since the Full method constructs monitoring statistic based on all nodes, its compensation for high-dimensional simultaneous tests results in the loss in detection power. In general, the proposed method shows overall satisfactory performance in this scenario compared to the benchmark methods.

The next comparison is for the four competing methods in the random seed scenario without memory constraints. The results are shown in Table IV, the setting of which is the same with Table III except that the seeds are randomly

TABLE III

COMPARISON OF ARL_1 AND THE CORRESPONDING STANDARD ERRORS (IN PARENTHESES) FOR DEGREE SEED WITHOUT MEMORY CONSTRAINT

NumSeeds	p_a	Methods	$\delta = 1$	$\delta = 2$	$\delta = 3$
5	0.1	Proposed	12.11 (0.09)	5.26 (0.02)	3.84 (0.02)
		TRAS	41.69 (1.09)	24.97 (1.06)	21.86 (1.06)
		Fixed	8.51 (0.04)	4.50 (0.01)	3.41 (0.01)
		Full	11.66 (0.04)	6.49 (0.02)	4.85 (0.01)
5	0.5	Proposed	7.71 (0.04)	4.16 (0.01)	3.34 (0.01)
		TRAS	13.31 (0.07)	7.36 (0.03)	5.80 (0.02)
		Fixed	7.99 (0.03)	4.43 (0.01)	3.40 (0.01)
		Full	9.45 (0.03)	5.91 (0.01)	4.66 (0.01)
20	0.1	Proposed	8.68 (0.05)	4.19 (0.02)	3.17 (0.01)
		TRAS	21.36 (0.19)	10.72 (0.09)	8.45 (0.08)
		Fixed	7.17 (0.03)	4.05 (0.01)	3.13 (0.01)
		Full	10.02 (0.03)	5.73 (0.01)	4.36 (0.01)
20	0.5	Proposed	6.83 (0.03)	3.77 (0.01)	3.02 (0.01)
		TRAS	11.12 (0.05)	6.30 (0.02)	4.94 (0.01)
		Fixed	7.08 (0.03)	4.04 (0.01)	3.13 (0.01)
		Full	8.55 (0.02)	5.39 (0.01)	4.24 (0.01)

TABLE IV

COMPARISON OF ARL_1 AND THE CORRESPONDING STANDARD ERRORS (IN PARENTHESES) FOR RANDOM SEED WITHOUT MEMORY CONSTRAINT

NumSeeds	p_a	Methods	$\delta = 1$	$\delta = 2$	$\delta = 3$
5	0.1	Proposed	116.13 (1.75)	37.36 (0.51)	23.13 (0.31)
		TRAS	130.61 (3.06)	108.18 (3.12)	104.61 (3.13)
		Fixed	326.92 (4.92)	325.99 (4.93)	325.79 (4.93)
		Full	13.92 (0.05)	6.85 (0.02)	4.92 (0.01)
5	0.5	Proposed	32.57 (0.79)	12.88 (0.24)	9.02 (0.14)
		TRAS	26.84 (0.73)	16.08 (0.68)	13.98 (0.68)
		Fixed	107.49 (3.41)	103.84 (3.43)	102.84 (3.43)
		Full	11.38 (0.04)	6.59 (0.02)	4.88 (0.01)
20	0.1	Proposed	40.05 (0.59)	13.37 (0.15)	8.60 (0.08)
		TRAS	31.04 (0.47)	16.29 (0.40)	13.48 (0.40)
		Fixed	211.80 (4.41)	208.59 (4.44)	207.86 (4.45)
		Full	10.87 (0.03)	5.89 (0.01)	4.41 (0.01)
20	0.5	Proposed	9.32 (0.05)	5.49 (0.02)	4.51 (0.01)
		TRAS	13.39 (0.07)	7.68 (0.03)	6.07 (0.02)
		Fixed	10.79 (0.40)	7.02 (0.40)	5.89 (0.40)
		Full	9.67 (0.03)	5.79 (0.01)	4.38 (0.01)

TABLE V

COMPARISON OF ARL_1 AND THE CORRESPONDING STANDARD ERRORS (IN PARENTHESES) FOR DEGREE SEED WITH MEMORY CONSTRAINT

NumSeeds	p_a	Methods	$\delta = 1$	$\delta = 2$	$\delta = 3$
33	0.05	Proposed	16.48 (0.14)	7.55 (0.05)	5.46 (0.03)
		TRAS	36.75 (0.70)	20.34 (0.62)	17.23 (0.61)
		Fixed	7.40 (0.03)	4.15 (0.01)	3.20 (0.01)
		Full	11.10 (0.03)	6.39 (0.01)	4.79 (0.01)
33	0.1	Proposed	11.75 (0.07)	6.20 (0.03)	4.74 (0.02)
		TRAS	17.60 (0.11)	9.89 (0.05)	7.95 (0.05)
		Fixed	7.36 (0.03)	4.14 (0.01)	3.19 (0.01)
		Full	10.39 (0.03)	6.28 (0.01)	4.76 (0.01)
67	0.05	Proposed	13.18 (0.10)	6.15 (0.03)	4.45 (0.02)
		TRAS	26.56 (0.27)	13.85 (0.17)	11.19 (0.16)
		Fixed	6.98 (0.03)	3.99 (0.01)	3.09 (0.01)
		Full	10.41 (0.03)	6.05 (0.01)	4.60 (0.01)
67	0.1	Proposed	10.65 (0.07)	5.50 (0.03)	4.14 (0.02)
		TRAS	16.38 (0.10)	8.86 (0.05)	7.01 (0.04)
		Fixed	6.95 (0.03)	3.99 (0.01)	3.09 (0.01)
		Full	9.82 (0.03)	5.95 (0.01)	4.56 (0.01)

TABLE VI

COMPARISON OF ARL_1 AND THE CORRESPONDING STANDARD ERRORS (IN PARENTHESES) FOR RANDOM SEED WITH MEMORY CONSTRAINT

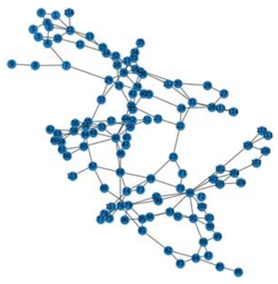
NumSeeds	p_a	Methods	$\delta = 1$	$\delta = 2$	$\delta = 3$
33	0.05	Proposed	141.12 (2.35)	67.82 (1.21)	45.96 (0.85)
		TRAS	128.14 (3.02)	106.99 (3.08)	103.41 (3.09)
		Fixed	307.44 (4.94)	305.95 (4.95)	305.61 (4.96)
		Full	12.25 (0.04)	6.58 (0.01)	4.83 (0.01)
33	0.1	Proposed	19.03 (0.30)	11.82 (0.13)	9.73 (0.11)
		TRAS	22.63 (0.29)	14.11 (0.27)	12.00 (0.26)
		Fixed	20.30 (0.95)	15.86 (0.96)	14.42 (0.96)
		Full	11.71 (0.03)	6.54 (0.01)	4.83 (0.01)
67	0.05	Proposed	89.27 (1.60)	37.08 (0.64)	24.49 (0.41)
		TRAS	54.58 (1.43)	34.78 (1.40)	31.41 (1.40)
		Fixed	198.90 (4.40)	195.14 (4.43)	194.29 (4.44)
		Full	11.13 (0.03)	6.16 (0.01)	4.63 (0.01)
67	0.1	Proposed	15.47 (0.08)	9.55 (0.04)	7.85 (0.03)
		TRAS	19.27 (0.11)	11.33 (0.06)	9.24 (0.05)
		Fixed	12.05 (0.05)	7.69 (0.03)	6.27 (0.02)
		Full	10.80 (0.03)	6.13 (0.01)	4.62 (0.01)

selected. Compared to the degree seed scenario, the detection delay is in general longer as random seed leads to fewer out-of-control nodes overall. As expected, the performance of the Fixed method degrades the most, which confirms that the Fixed method is rigid in mechanism and relies heavily on the out-of-control scenario. The proposed method, on the contrary, is adaptive to online observations and consistently outperforms the Fixed method. The detection performance of the Full method is among the best in the random seed scenario as it utilizes all observations. However, it is not practical under the resource constraints. The proposed method utilizes only around 1% of the information and thus is less effective when $NumSeeds$, p_a, and δ are extremely small. But it can achieve an comparable performance otherwise. In the random scenario, the TRAS algorithm is generally outperformed by the proposed method since it utilizes a simple sampling framework. From Tables III and IV, it can be observed that the proposed method shows overall best performance under various scenarios with partial observations and well balances between exploration and exploitation.

We further test the proposed method in Section IV with memory constraint on the large network. We assume $M = 200$ in this section. The result for degree seed is shown in Table V and that for random seed is in Table VI. The results in both scenarios are very similar to those in Tables III and IV. Therefore, the superiority of the proposed method still holds in the memory-constrained case, which shows the effectiveness and robustness of the proposed methods.

VII. CASE STUDY

In this section, we use smart grid as a case study. Thanks to the development of the information and communication technology, sensors such as smart meters and phasor measurement units can report the measurements timely and remotely. Besides, as the price of the sensors decreases, the wide deployment of sensors becomes affordable. Thus, with real-time measurements monitoring most grid nodes, the grid operators are enhanced with situational awareness and can maintain the grid more efficiently. On the other hand, since the scale of a grid can be extremely large, monitoring the states of each node exposes heavy burdens to the communication network. Moreover, analyzing such large amount of data is time-consuming and can violate the real-time requirement of the grid. To address these limitations and monitor the grid more robustly, we apply the proposed adaptive monitoring method to power grids. Specifically, to monitor the stability of a grid, the voltage magnitude at each node is selected as the interested metric because it presents a direct perception on the state of the grid. Since electrical devices can function improperly or be damaged with high or low voltages, it is critical to keep monitoring the voltages and guarantee that the voltages are within a pre-defined range. The voltage measurements of neighboring nodes are correlated by the


(a) IEEE 118-bus network

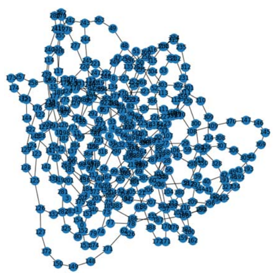

(b) LVNTS network

Fig. 3. Graphs of two power networks in the case study.

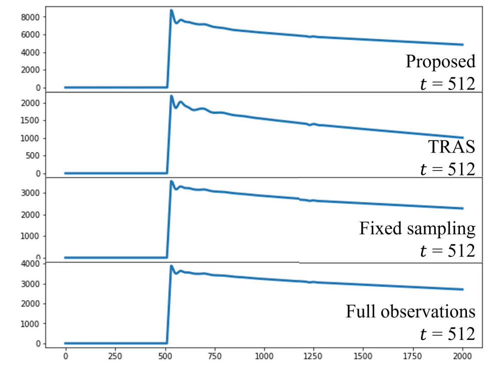


Fig. 4. Monitoring statistics and detection time for four competing methods implemented on the IEEE 118-bus network.

Kirchoff's circuit law and the Ohm's law. Thus, the proposed method has an improved anomaly detection capability with the neighboring measurements included.

We simulate line outages as the out-of-control abnormalities. During a line outage, the power between two nodes is cut off. Thus, the power flows in the remaining network will be re-distributed to meet the network demands, which leads to voltage variations. Depending on different types of network structures, the voltage variation patterns can be different. Specifically, for meshed networks, in which most nodes have more than one neighbors, the voltage variations will be small. Based on the direction of the power flow, we differentiate two nodes connected by a line as the starting node and the ending node, and we call the starting node as the parent node of the ending node. The voltage of the starting node will slightly increase because itself or its parent node need to push more powers to the remaining neighbors. Meanwhile, the voltage of the ending node will decrease but not to zeros because powers can still be supplied by other neighbors. On the other hand, for radial networks, the power flowing into a sub-tree is supplied by a single node. Once the connection to the node is cut off, the voltages of all the nodes in the subtree will become zero. The voltage of the parent node will increase because less power is injected into the network (meshed networks do not have decreased power injections because the graph is still connected). Furthermore, when the line outage is cleared, in the radial network, the voltage of the parent node will have a sudden decrease because of the suddenly increasing load, while in the meshed network, such a voltage variation is negligible.

We will investigate the effectiveness of the proposed method on grids with different sizes and types. Simulations are conducted on three grid models: the IEEE 118-bus network [42], the 342-Node Low Voltage Network Test System (LVNTS) [43], and the 8500-Node Test Feeder [44], each of which is run for 20s with a time step of 0.01s. Line outages occur at lines which are randomly selected. The IEEE 118-bus network is a transmission network with a meshed topology. It has 118 nodes, including 19 generators, which inject power to the network. Line outages are simulated at time 5s-5.2s on line 35-65 and 12s-12.2s on line 77-82. Since generators appear in the network, we formulate detailed physical models and control models of the generators and simulate the grid with the root-mean square (RMS) model to capture the voltage oscillations caused by the line outages. Besides, LVNTS is a distribution network with a mesh topology. It has 390 nodes, including 150 nodes constituting 8 radial primary distribution feeders, 192 constituting the grid network, and 48 auxiliary nodes constituting 8 spot networks. Three line outages are simulated on the three parts, including line outages at 5s-5.2s on line P141-P142, 10s-10.2s on line S53-S54, and 15s-15.2s on line S196-S194. Since no generator appears, the voltage oscillations are negligible, and the phasor simulation is used for speeding up. Furthermore, the 8500-Node Test Feeder is a distribution network as well but with a radial topology. There is only one feeder in the network. Therefore, the tree structure of the grid makes the anomaly at a node observable at its off-springs. Line outages are simulated at 5s-5.2s on line L3235275-M1026347, 10s-10.2s on line L2673313-M1027011, and 15s-15.2s on line R18243-N1139552. Similar to LVNTS, phasor simulation is used. For simplicity, we only monitor the voltages of phase A.

The monitoring statistics and detection time of the four competing methods on the IEEE 118-bus network are shown in Fig. 4. The parameters are set as $q = 10$ and $M = 50$. As the network size is small and the shift is large, the detection results are similar for all four methods. Since the RMS model is used, the voltage oscillation increases gradually. Thus, a small delay is induced, and the abnormality is detected at $t = 512$ for all four methods.

The monitoring statistics and detection time of the four competing methods on the LVNTS network are shown in Fig. 5. The parameters are set as $q = 10$ and $M = 200$ in this case. For this scenario, the proposed method triggers the out-of-control alarm slightly later than the Fixed and Full methods. The TRAS method, however, takes much longer to detect the shift. From the monitoring statistic of the TRAS method, it seems that it failed to sample the major out-of-control nodes of the first line outage and was not fully aware of the large shift.

The monitoring statistics and detection time of the four competing methods on the 8500-Node Test Feeder network are shown in Fig. 6. The parameters are set as $q = 10$ and $M = 200$ in this case. For this scenario, the proposed method triggers the out-of-control alarm right after the shift, which is the same as the Full methods. Although the TRAS method also triggers an alarm quickly, it did not kept any key

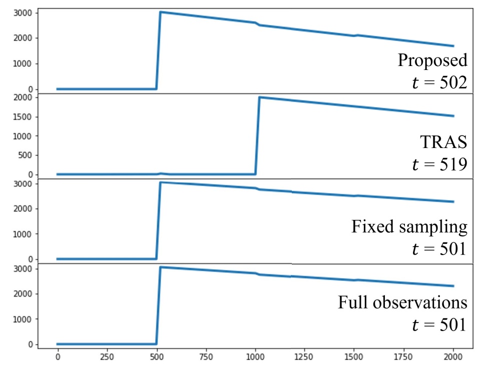


Fig. 5. Monitoring statistics and detection time for four competing methods implemented on the LVNTS network.

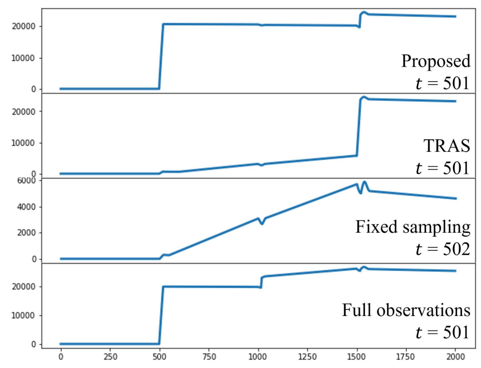


Fig. 6. Monitoring statistics and detection time for four competing methods implemented on the 8500-Node Test Feeder network.

out-of-control nodes under surveillance as its monitoring statistic did not arise quickly after time $t = 500$. The same observation comes from the Fixed method. The proposed method, on the contrary, had an quick increase in the monitoring statistic and kept monitoring those node, evident by the monitoring statistic value after $t = 500$. This observation is consistent with the out-of-control property in Section V.

Combining the results from all three networks considered in the case study, we can see that the proposed method can effectively detect changes with only limited monitoring and memory resources, and its performance is the closest to the Full method even though it uses less resources. The results show the advantage and practicality of the proposed method in real applications. Note that the consecutive line outages are less significant because the monitoring statistic is dominated by the first line outage. To make all the line outages remarkable, we can re-initialize the monitoring statistics after the line outages are cleared.

VIII. Conclusion

Effective online monitoring of big data streams in networks has been a challenging topic in anomaly detection. This paper investigated a network-based sampling strategy which can adaptively identify the most informative data streams to observe for quick change detection with resource constraints. In particular, the proposed algorithms effectively handled the monitoring variables in network where only a subset of observations are available at each acquisition time. The proposed method is also generalized to the case where memory constraint poses a challenge. In addition, we investigated and proved two important properties regarding the sampling layout of the proposed algorithm with and without memory constraints. When the process is in control, the algorithms keep searching for abnormality among all variables. This property ensures that no matter when and where the mean shift occurs, our sampling algorithm is able to quickly realize the changes. When the process is out of control, the algorithms stick to the nodes that are most likely out of control. This property ensures that our sampling algorithm is able to effectively leverage the limited resources to maximize the detection capability. The performance of the proposed algorithms was demonstrated and evaluated through simulations and a case study on power grid monitoring. Both studies revealed the effectiveness of the proposed algorithms under various network settings.

Appendix A
Calculating the Average Run Length

Here is the algorithm for calculating the run length (RL) of a control chart for a single process, given a threshold L:

1) Run the monitoring procedure for a time horizon T (we call it a single run).
2) For $t = 1 : T$, if $W(t) > L$, the run length $RL(L) = t$.
3) Return t.

Algorithm for calculating the average run length (ARL) by repeating the above procedure for n repetitions:

1) For run $i = 1 : n$, calculate its run length $RL_i(L)$.
2) The average run length is calculated as $ARL(L) = \frac{1}{n}\Sigma_i^n RL_i(L)$.

As a side note, most of the time we also calculate the standard deviation of ARL for comparison purpose. n is typically chosen as 1000, 5000, 10000, depending on the computation time.

To control the false alarm rate, the threshold L should be selected such that the average run length is a target value ARL_0 (typically 200 or 370). Here is a basic algorithm for searching for the threshold value L given a ARL_0 value:

```
1) For $j = 1, 2, \cdots$, calculate $ARL(L = exp(j))$ until $ARL(L = exp(j)) > ARL_0$ (find out the upper and lower bound of ARL, which later denoted as $L_1$ and $L_2$);
2) Denote $L_1 = exp(j-1)$ and $L_2 = exp(j)$;
3) Until $|ARL(L) - ARL_0| < \epsilon$ or $|L_1 - L_2| < \epsilon$:
       calculate $ARL(L = \frac{1}{2}(L_1 + L_2))$;
       if $ARL(L = \frac{1}{2}(L_1 + L_2)) < ARL_0$:
           let $L_1 = L$;
       else:
           let $L_2 = L$.
```

Appendix B
Proof of Theorem 1

Proof for networks without memory constraint: We first show that there are infinite many time stamps such that

$q_D(t) < q$. In the in-control case, the local monitoring statistics

$$\begin{aligned}W(t) &= \max_i W_i(t)\\ &= \max_i \max\Big(W_i(t-1) + \sum_{j\in\mathscr{O}_t} K_h(d|i,j|)\big([\eta(\zeta_1)\\ &\quad -\eta(\zeta_0)]T(X_j(t)) - [A(\zeta_1)-A(\zeta_0)]\big), 0\Big)\\ &\le \max(S(t),0),\end{aligned}$$

where $S(t) = \sum_{\tau\le t}\sum_{j\in\mathscr{O}_\tau}\max\Big([\eta(\zeta_1)-\eta(\zeta_0)]T(X_j(\tau)) - [A(\zeta_1)-A(\zeta_0)], 0\Big)$. $S(t)$ on the right hand side is a random walk with a negative mean, which means that $W(t)$ is bounded by a random process that returns to 0 infinite number of time [45]. As an special example, if $X_i(t)$ follows standard normal distribution and select $f_1(x)$ to be two-sided $N(\mu_1,1)$ and $N(-\mu_1,1)$ for $\mu_1>0$,

$$\begin{aligned}W(t) &= \max_i W_i(t) = \max_i \max\Big(W_i(t-1)\\ &\quad + \sum_{j\in\mathscr{O}_t} K_h(d|i,j|)\big(\mu_1 X_j(t) - \frac{\mu_1^2}{2}\big),\\ &\quad W_i(t-1) + \sum_{j\in\mathscr{O}_t} K_h(d|i,j|)\Big(-\mu_1 X_j(t) - \frac{\mu_1^2}{2}\Big), 0\Big)\\ &\le \max(S(t),0),\end{aligned}$$

where $S(t) = \sum_{\tau\le t}\sum_{j\in\mathscr{O}_\tau}\max\big(\mu_1 X_j(t) - \frac{\mu_1^2}{2}, -\mu_1 X_j(t) - \frac{\mu_1^2}{2}, 0\big)$

Therefore, there are infinite number of time points such that $W(t) < D\Big(\frac{1-\theta}{\lambda}\frac{q-1}{q} + \theta\Big)$, which further leads to

$$\frac{q\lambda(W(t)-D\theta)}{D(1-\theta)} < q-1,$$

such that $q_D(t) = \min\Big(\Big\lceil \frac{q\lambda\max(W(t)-D\theta,0)}{D(1-\theta)}\Big\rceil, q\Big) \le q-1$. This is equivalent to the fact that there are infinite number of times $t_1, t_2, \cdots$ such that $q_B(t) \ge 1$. Note that at any of these time there is non-zero probability p for any variable to be sampled by the BFS mechanism given that employed centralities are positive for all nodes. Therefore, the probability for a node i to be in set U is

$$P(i\in U) = \lim_{j\to\infty}(1-p)^j \to 0.$$

Proof for networks with memory constraint: Following the proof for small networks, we only need to show that there are infinite number of times that certain variables will be moved out of the memory, as any nodes has a positive probability bounded from below. This is equivalent to show $W_i(t) < l$ for infinite number of times. Actually,

$$\begin{aligned}W_i(t) &= \max\Big(W_i(t-1) + \sum_{j\in\mathscr{O}_t} K_h(d|i,j|)\big([\eta(\zeta_1)-\eta(\zeta_0)]\\ &\quad \cdot T(X_j(t)) - [A(\zeta_1)-A(\zeta_0)]\big), 0\Big)\\ &< \sum_{\tau\le t}\sum_{j\in\mathscr{O}_\tau}\max\Big([\eta(\zeta_1)-\eta(\zeta_0)]T(X_j(\tau))\\ &\quad -[A(\zeta_1)-A(\zeta_0)], 0\Big).\end{aligned}$$

Similarly, the last term is a random walk with a negative mean, which means that $W_i(t)$ is bounded by a random process that returns to 0 infinite number of times. This finished the proof.

Appendix C
Proof of Theorem 2

Proof for networks without memory constraint: We show the proof for large enough δ such that $[\eta(\zeta_1)-\eta(\zeta_0)]\mathbb{E}T(X_j(t)) - [A(\zeta_1)-A(\zeta_0)] > 0$, $\lambda > \frac{q-1}{q}$ and $h=1$. Without loss of generality, we assume the shift is positive. At time t, if $[\eta(\zeta_1)-\eta(\zeta_0)]T(X_j(t)) > [A(\zeta_1)-A(\zeta_0)] + D$,

$$\begin{aligned}W(t) &\ge W_j(t) \ge \max\Big(W_j(t-1) + \big([\eta(\zeta_1)-\eta(\zeta_0)]\\ &\quad \cdot T(X_j(t)) - [A(\zeta_1)-A(\zeta_0)]\big), 0\Big)\\ &\ge \big([\eta(\zeta_1)-\eta(\zeta_0)]T(X_j(t)) - [A(\zeta_1)-A(\zeta_0)]\big) > D.\end{aligned}$$

As a special example that X_j follows normal distribution, $[\eta(\zeta_1)-\eta(\zeta_0)]T(X_j(t)) - [A(\zeta_1)-A(\zeta_0)] = \mu_1 X_j - \frac{\mu_1^2}{2}$ is positive if $\delta > \frac{\mu_1}{2}$. Then $\mu_1 X_j - \frac{\mu_1^2}{2} > D$ with probability $1-\Phi\Big(\frac{D}{\mu_1} + \frac{\mu_1}{2} - \delta\Big)$ and

$$\begin{aligned}W(t) &\ge W_j(t) \ge \max\Big(W_j(t-1) + \mu_1 X_j(t) - \frac{\mu_1^2}{2},\\ &\quad W_j(t-1) - \mu_1 X_j(t) - \frac{\mu_1^2}{2}, 0\Big) \ge \mu_1 X_j(t) - \frac{\mu_1^2}{2} > D.\end{aligned}$$

Therefore,

$$\begin{aligned}\frac{q\lambda\max(W(t)-D\theta,0)}{D(1-\theta)} &= \frac{q\lambda(W(t)-D\theta)}{D(1-\theta)}\\ &> \frac{q\lambda(D-D\theta)}{D(1-\theta)} > q-1,\end{aligned}$$

and thus

$$q_D(t) = \min\Big(\Big\lceil \frac{q\lambda\max(W(t)-D\theta,0)}{D(1-\theta)}\Big\rceil, q\Big) = q.$$

Considering any other node $i \notin \mathscr{O}_t$, let

$$\begin{aligned}Y_i(t+n) &= W_j(t-1) - W_i(t-1)\\ &\quad + \sum_{\tau=t}^{t+n}\big([\eta(\zeta_1)-\eta(\zeta_0)]T(X_j(\tau))\\ &\quad -[A(\zeta_1)-A(\zeta_0)]\big).\end{aligned}$$

$Y_i(t+n)$ is a general random walk with mean $[\eta(\zeta_1)-\eta(\zeta_0)]\mathbb{E}T(X_j(\tau)) - [A(\zeta_1)-A(\zeta_0)] > 0$. Therefore, according to Ross et al. [46], there is a non-zero probability $P(Y_i(t+n) > 0$ for all $n \ge 0)$. This shows that there is non-zero probability that the monitoring statistic of variable j is always large or equal to any other variable not in $\mathscr{O}_t$, which finishes the proof.

As a special case where $X_j(t)$ follows normal distribution, $Y_i(t+n)$ is a Gaussian random walk with mean $\mathbb{E}\Big(\mu_1 X_j(\tau) - \frac{\mu_1^2}{2}\Big) = \mu_1\delta - \frac{\mu_1^2}{2} > 0$. Therefore, according to Janssen et al. [47], $P(Y_i(t+n) > 0 \text{ for all } n \geq 0)$ is a non-zero probability that

$$\begin{aligned}&P(Y_i(t+n) > 0 \text{ for all } n \geq 0)\\ &\quad\geq P\left(\frac{W_j(t-1)-W_i(t-1)}{\mu_1} + X_j(t) - \frac{\mu_1}{2} > 0\right)\cdot\\ &P\left(\sum_{\tau=t+1}^{\infty}\left(\mu_1 X_j(\tau) - \frac{\mu_1^2}{2}\right) \geq 0\right)\\ &\quad= \left(1-\Phi\left(\delta - \frac{W_j(t-1)-W_i(t-1)}{\mu_1} + \frac{\mu_1}{2}\right)\right)\cdot\\ &\sqrt{2}(\delta - \frac{\mu_1}{2})\exp\left\{\frac{\delta-\frac{\mu_1}{2}}{\sqrt{2\pi}}\sum_{r=0}^{\infty}\frac{\zeta(\frac{1}{2}-r)}{r!(2r+1)}\left(-\frac{(\delta-\frac{\mu_1}{2})^2}{2}\right)^r\right\},\end{aligned} \tag{10}$$

for $0 \leq \delta - \frac{\mu_1}{2} \leq 2\sqrt{\pi}$, where $\zeta(\cdot)$ is the Riemann zeta function.

Proof for networks with memory constraint: We show the proof for $D > l$. It is obvious from the proof for networks without memory constraint that once variable i is kept observed, $W_j(t) > D > l$ and thus will always be kept in memory.

Appendix D
Proof of Theorem 3

Proof: Without loss of generality, consider a scheme where the monitoring statistic is $W_i(t)$ instead of $W(t) = \max_{i\in\mathscr{P}_t} W_i(t)$. The stopping time in this scheme is $T' = \inf\{t \geq 1 : W_i(t) \geq L\}$. Let S_k denotes the total time that node k is observed before the stopping time T'. We have $T' = \frac{\sum_{k\in V} S_k}{q}$. Then $\mathbb{E}(T') = \frac{\mathbb{E}(\sum_{k\in V} S_k)}{q} = \frac{\mathbb{E}(\sum_{k\in V} S_k)}{q\mathbb{E}(S_i)}\mathbb{E}(S_i)$. As W_1 is only updated when node i is observed, and the stopping time depends solely on W_i, we have $\mathbb{E}(S_i) \leq \frac{L}{I(f_{1,i}, f_{0,i})} + O(1)$ as $L \to \infty$, followed by Chapter 2 of [45]. In addition, $\lim_{L\to\infty}\frac{\mathbb{E}(\sum_{k\in V} S_k)}{\mathbb{E}(S_i)} = \frac{1}{q_i}$. Therefore, we have $\mathbb{E}(T') \leq \frac{L}{q\gamma_i I(f_{1,i}, f_{0,i})} + O(1)$ as $L \to \infty$.

At the time that W_i is updated, by the definition of W_i, we have $W(t) = \sum_{k=1}^{n} W_k(t) \geq W_i(t)$. Therefore, $\mathbb{E}(T) \leq \mathbb{E}(T') \leq \frac{L}{q\gamma_i I(f_{1,i}, f_{0,i})} + O(1)$ as $L \to \infty$. Since the inequality holds for all locations $i \in V$, we have

$$\mathbb{E}(T) \leq \min_{i\in V}\left\{\frac{L}{q\gamma_i I(f_{1,i}, f_{0,i})}\right\} + O(1).$$

Acknowledgment

The authors appreciate the kind support from National Science Foundation and the Florida Center for Cybersecurity. The authors would also like to thank the associate editor and the reviewers' helpful comments.

References

[1] A. N. Samudrala, M. H. Amini, S. Kar, and R. S. Blum, "Sensor placement for outage identifiability in power distribution networks," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 1996–2013, May 2020.

[2] F. Passerini and A. M. Tonello, "Smart grid monitoring using power line modems: Anomaly detection and localization," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6178–6186, Nov. 2019.

[3] J. H. Hong, C.-C. Liu, and M. Govindarasu, "Integrated anomaly detection for cyber security of the substations," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1643–1653, Jul. 2014.

[4] H. Zhang, M. A. Alim, M. T. Thai, and H. T. Nguyen, "Monitor placement to timely detect misinformation in online social networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 1152–1157.

[5] H. Zhang, L. Zhang, D. Cheng, Y. Wu, and C. Zhao, "EpCom: A parallel community detection approach for epidemic diffusion over social networks," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Nov. 2017, pp. 1607–1614.

[6] A. G. Tartakovsky, A. S. Polunchenko, and G. Sokolov, "Efficient computer network anomaly detection by changepoint detection methods," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 1, pp. 4–11, Feb. 2013.

[7] M. Nabhan, Y. Mei, and J. Shi, "Correlation-based dynamic sampling for online high dimensional process monitoring," *J. Quality Technol.*, vol. 53, no. 3, pp. 289–308, 2021.

[8] H. H. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, and A. Liotta, "Spatial anomaly detection in sensor networks using neighborhood information," *Inf. Fusion*, vol. 33, pp. 41–56, Jan. 2017.

[9] V. Saligrama and Z. Chen, "Video anomaly detection based on local statistical aggregates," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2112–2119.

[10] H. Yan, K. Paynabar, and J. Shi, "Real-time monitoring of high-dimensional functional data streams via spatio-temporal smooth sparse decomposition," *Technometrics*, vol. 60, no. 2, pp. 181–197, 2018.

[11] G. Rovatsos, S. Zou, and V. V. Veeravalli, "Sequential algorithms for moving anomaly detection in networks," *Sequential Anal.*, vol. 39, no. 1, pp. 6–31, Jan. 2020.

[12] S. Zou, V. V. Veeravalli, J. Li, and D. Towsley, "Quickest detection of dynamic events in networks," *IEEE Trans. Inf. Theory*, vol. 66, no. 4, pp. 2280–2295, Apr. 2020.

[13] A. Jain and E. Y. Chang, "Adaptive sampling for sensor networks," in *Proc. 1st Int. Workshop Data Manag. Sensor Netw. Conjunct (VLDB-DMSN)*, 2004, pp. 10–16.

[14] K. Liu, Y. Mei, and J. Shi, "An adaptive sampling strategy for online high-dimensional process monitoring," *Technometrics*, vol. 57, no. 3, pp. 305–319, Jul. 2015.

[15] C. Alippi, G. Anastasi, C. Galperti, F. Mancini, and M. Roveri, "Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications," in *Proc. IEEE Internatonal Conf. Mobile Adhoc Sensor Syst.*, Oct. 2007, pp. 1–6.

[16] D. Koschützki, K. A. Lehmann, L. Peeters, S. Richter, D. Tenfelde-Podehl, and O. Zlotowski, "Centrality indices," in *Network Analysis*. Berlin, Germany: Springer, 2005, pp. 16–61.

[17] D. C. Montgomery, *Introduction to Statistical Quality Control*. Hoboken, NJ, USA: Wiley, 2020.

[18] X. Xian, A. Wang, and K. Liu, "A nonparametric adaptive sampling strategy for online monitoring of big data streams," *Technometrics*, vol. 60, no. 1, pp. 14–25, Jan. 2018, doi: 10.1080/00401706.2017.1317291.

[19] A. Wang, X. Xian, F. Tsung, and K. Liu, "A spatial-adaptive sampling procedure for online monitoring of big data streams," *J. Quality Technol.*, vol. 50, no. 4, pp. 329–343, Oct. 2018.

[20] X. Xian, C. Zhang, S. Bonk, and K. Liu, "Online monitoring of big data streams: A rank-based sampling algorithm by data augmentation," *J. Quality Technol.*, vol. 53, no. 2, pp. 135–153, Mar. 2021.

[21] W. H. Woodall, M. J. Zhao, K. Paynabar, R. Sparks, and J. D. Wilson, "An overview and perspective on social network monitoring," *IISE Trans.*, vol. 49, no. 3, pp. 354–365, Mar. 2017.

[22] B. Azarnoush, K. Paynabar, J. Bekki, and G. Runger, "Monitoring temporal homogeneity in attributed network streams," *J. Quality Technol.*, vol. 48, no. 1, pp. 28–43, Jan. 2016.

[23] K. Lerman and R. Ghosh, "Information contagion: An empirical study of the spread of news on Digg and Twitter social networks," 2010, *arXiv:1003.2664*.

[24] M. Samadi, R. Nagi, A. Semenov, and A. Nikolaev, "Seed activation scheduling for influence maximization in social networks," *Omega*, vol. 77, pp. 96–114, Jun. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305048316303577

[25] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York, NY, USA, 2003, pp. 137–146, doi: 10.1145/956750.956769.

[26] S. Erkol, C. Castellano, and F. Radicchi, "Systematic comparison between methods for the detection of influential spreaders in complex networks," *Sci. Rep.*, vol. 9, no. 1, p. 15095, Oct. 2019, doi: 10.1038/s41598-019-51209-6.

[27] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2009, pp. 199–208. [Online]. Available: https://www.microsoft.com/en-us/research/publication/efficient-influence-maximization-social-networks/

[28] A. Braunstein, L. Dall'Asta, G. Semerjian, and L. Zdeborová, "Network dismantling," *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 44, pp. 12368–12373, Nov. 2016. [Online]. Available: https://www.pnas.org/content/113/44/12368

[29] A. Semenov, A. Veremyev, A. Nikolaev, E. L. Pasiliao, and V. Boginski, "Network-based indices of individual and collective advising impacts in mathematics," *Comput. Social Netw.*, vol. 7, no. 1, pp. 1–18, Jan. 2020, doi: 10.1186/s40649-019-0075-0.

[30] W. Zhang and Y. Mei, "Bandit change-point detection for real-time monitoring high-dimensional data under sampling control," 2020, *arXiv:2009.11891*.

[31] X. Xian, R. Archibald, B. Mayer, K. Liu, and J. Li, "An effective online data monitoring and saving strategy for large-scale climate simulations," *Qual. Technol. Quant. Manag.*, vol. 16, no. 3, pp. 330–346, May 2019.

[32] S. Babu, U. Srivastava, and J. Widom, "Exploiting K-constraints to reduce memory overhead in continuous queries over data streams," *ACM Trans. Database Syst.*, vol. 29, no. 3, pp. 545–580, Sep. 2004.

[33] A. Eswaran, A. Rowe, and R. Rajkumar, "Nano-RK: An energy-aware resource-centric RTOS for sensor networks," in *Proc. 26th IEEE Int. Real-Time Syst. Symp. (RTSS)*, Dec. 2005, p. 10.

[34] V. Raghavan and V. V. Veeravalli, "Quickest change detection of a Markov process across a sensor array," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1961–1981, Apr. 2010.

[35] D. A. van Dyk and X.-L. Meng, "The art of data augmentation," *J. Comput. Graph. Statist.*, vol. 10, no. 1, pp. 1–50, 2001.

[36] A. Tartakovsky and H. Kim, "Performance of certain decentralized distributed change detection procedures," in *Proc. 9th Int. Conf. Inf. Fusion*, Jul. 2006, pp. 1–8.

[37] D. Chen, L. Lü, M.-S. Shang, Y.-C. Zhang, and T. Zhou, "Identifying influential nodes in complex networks," *Phys. A, Statist. Mech. Appl.*, vol. 391, pp. 1777–1787, Feb. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378437111007333

[38] J. M. Lucas and R. B. Crosier, "Fast initial response for CUSUM quality-control schemes: Give your CUSUM a head start," *Technometrics*, vol. 42, no. 1, pp. 102–107, Feb. 2000.

[39] A. Luceño and J. Puig-Pey, "The random intrinsic fast initial response of one-sided CUSUM charts," *J. Appl. Statist.*, vol. 33, no. 2, pp. 189–201, Mar. 2006.

[40] T. R. Rhoads, D. C. Montgomery, and C. M. Mastrangelo, "A fast initial response scheme for the exponentially weighted moving average control chart," *Qual. Eng.*, vol. 9, no. 2, pp. 317–327, Jan. 1996.

[41] W. Chen, T. Lin, Z. Tan, M. Zhao, and X. Zhou, "Robust influence maximization," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016, pp. 795–804, doi: 10.1145/2939672.2939745.

[42] *IEEE 118-Bus Network*. Accessed: May 1, 2021. [Online]. Available: http://labs.ece.uw.edu/pstca/pf118/pg_tca118bus.htm

[43] *IEEE 118-Bus Network*. Accessed: May 1, 2021. [Online]. Available: http://site.ieee.org/pes-testfeeders/files/2017/08/European_LV_Test_Feeder_v2.zip

[44] *IEEE 118-Bus Network*. Accessed: May 1, 2021. [Online]. Available: http://site.ieee.org/pes-testfeeders/files/2017/08/8500node.zip

[45] D. Siegmund, *Sequential Analysis: Tests and Confidence Intervals*. Berlin, Germany: Springer, 2013.

[46] S. M. Ross et al., *Stochastic Processes*, vol. 2. Hoboken, NJ, USA: Wiley, 1996.

[47] A. J. E. M. Janssen and J. S. H. van Leeuwaarden, "On Lerch's transcendent and the Gaussian random walk," *Ann. Appl. Probab.*, vol. 17, no. 2, pp. 421–439, Apr. 2007.

Xiaochen Xian (Member, IEEE) received the B.S. degree in mathematics from Zhejiang University, Hangzhou, China, in 2014, and the M.S. degree in statistics and the Ph.D. degree in industrial engineering from the University of Wisconsin–Madison in 2017 and 2019. She is currently an Assistant Professor at the Department of Industrial and Systems Engineering, University of Florida. Her research interests are focused on big data analytics and system informatics. She is a member of INFORMS and IISE.

Alexander Semenov received the Ph.D. degree in computer science from the University of Jyväskylä, Finland. His research interests include network science, efficient algorithms, analysis of large datasets, optimization, and machine learning. He has coauthored over 40 peer-reviewed publications and has been a recipient of multiple research grants.

Yaodan Hu (Member, IEEE) received the Bachelor of Science degree in applied physics from the University of Science and Technology of China (USTC), China, in 2015, and the Ph.D. degree in electrical and computer engineering from the University of Florida. She is currently an Assistant Professor at the Department of Electrical and Computer Engineering, Idaho State University. Her research interests include attack and defense mechanism design in cyber-physical systems.

Andi Wang received the Ph.D. degree in industrial engineering and the M.S. degree in computer science and engineering from the Georgia Institute of Technology. He is currently an Assistant Professor with the Polytechnic School (TPS) and the Newly Founded School of Manufacturing Systems and Networks (MSN), Arizona State University (ASU). His research interests include the intersection of data science and manufacturing systems.

Yier Jin (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Zhejiang University, China, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from Yale University in 2012. He is currently an Associate Professor and the IoT Term Professor with the Department of Electrical and Computer Engineering (ECE), University of Florida (UF). His research interests include the areas of hardware security, embedded systems design and security, trusted hardware intellectual property (IP) cores, and hardware-software co-design for modern computing systems.