Statistical methods for network surveillance

Article *in* Applied Stochastic Models in Business and Industry · April 2018

DOI: 10.1002/asmb.2326

Daniel R. Jeske
University of California, Riverside
142 PUBLICATIONS **1,709** CITATIONS

Nathaniel Stevens
University of San Francisco
23 PUBLICATIONS **125** CITATIONS

James D. Wilson
University of San Francisco
32 PUBLICATIONS **231** CITATIONS

Received: 26 September 2017 | Revised: 23 February 2018 | Accepted: 24 February 2018

DOI: 10.1002/asmb.2326

DISCUSSION PAPER

WILEY

Statistical methods for network surveillance

Daniel R. Jeske[1] | Nathaniel T. Stevens[2] | Alexander G. Tartakovsky[3,4] | James D. Wilson[2]

[1] Department of Statistics, University of California, Riverside, CA, USA

[2] Department of Mathematics and Statistics, University of San Francisco, San Francisco, CA, USA

[3] Moscow Institute of Physics and Technology, Moscow, Russia

[4] AGT StatConsult, Los Angeles, CA, USA

Correspondence
Daniel R. Jeske, Department of Statistics, University of California, Riverside, CA 92521, USA.
Email: daniel.jeske@ucr.edu

Funding information
US Army Research Office, Grant/Award Number: W911NF-14-1-0246; US Department of Energy Phase I SBIR; Ministry of Education and Science of the Russian Federation

The term network surveillance is defined in general terms and illustrated with many examples. Statistical methodologies that can be used as tools for network surveillance are discussed. Details for 3 illustrative examples that address network security, surveillance for data network failures, and surveillance of email traffic flows are presented. Some open areas of research are identified.

KEYWORDS

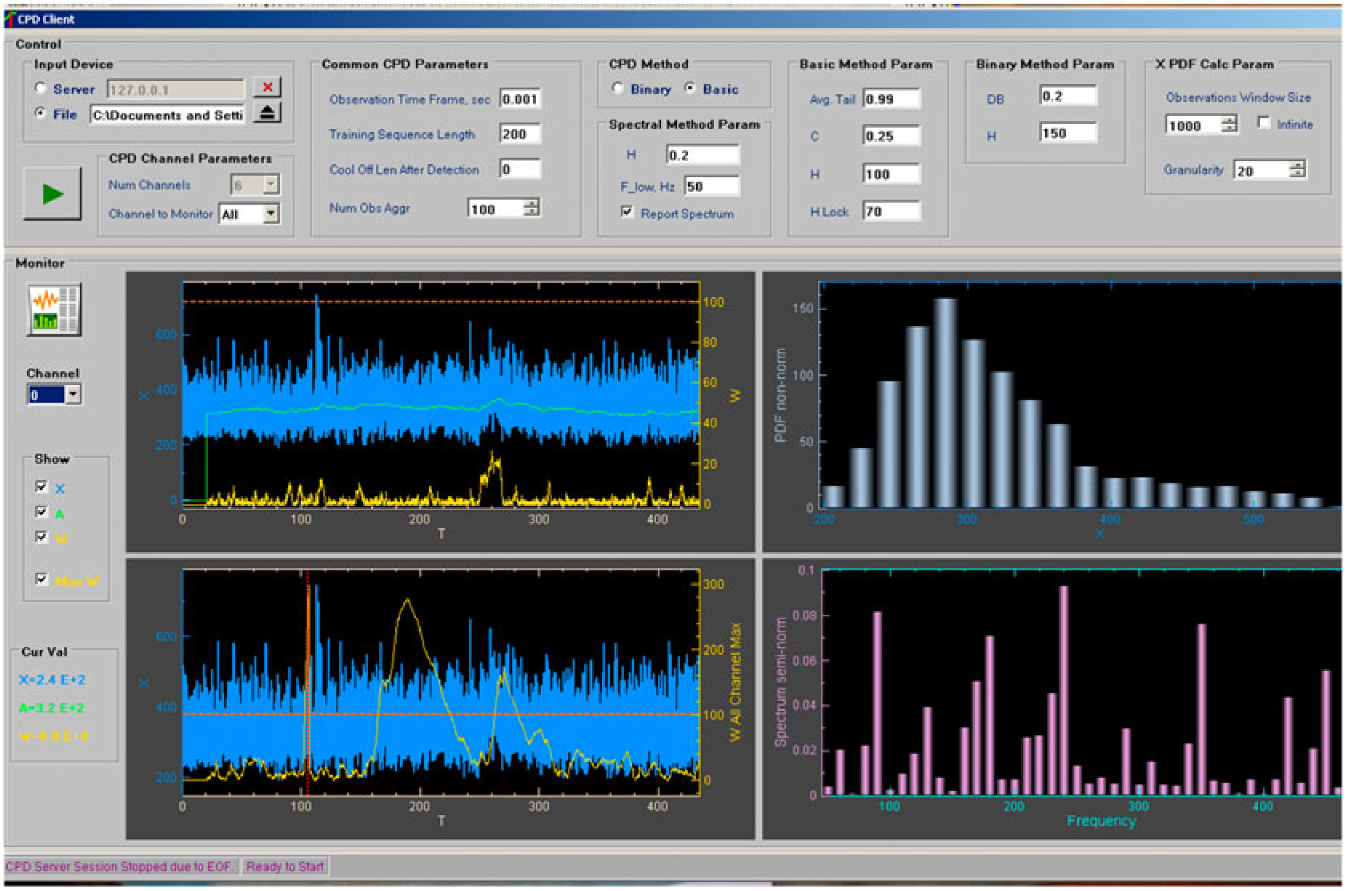
change point, control charts, dynamic networks, reliability, security

1 | INTRODUCTION

Generally speaking, network surveillance is a term that means monitoring a network to detect abnormal behavior. Applications where network surveillance is employed vary considerably, since the term *network* itself has a broad meaning: in telecommunications, a *data network* is infrastructure that enables point-to-point information transfer; a *biological network* describes how different regions of the human brain communicate with each other; an *artificial neural network* is a type of statistical prediction algorithm; *social networks* study the way individuals or communities of individuals interact with one another; *gene regulatory networks* describe the interplay between different genes in the human developmental process.

Interest in monitoring a network is driven by a desire to identify transitions away from normal, or baseline, operating conditions. Such a transition may be associated with an interesting cause. In the case of a data network, the transition may signal a failure of a key piece of equipment or possibly a malicious attack. For biological neural networks or gene regulatory networks, the transition may provide insight for understanding how a disease originates and/or evolves. In social networks, the transition may coincide with changes in other types of social or antisocial behavior.

In this review, we will reveal some of the underlying challenges of network surveillance. Specific examples where statistical methods for monitoring networks have been developed will be presented, and some open areas of research on this topic will be discussed.



CPD Client

Control

Input Device
○ Server 127.0.0.1
● File C:\Documents and Setti

CPD Channel Parameters
Num Channels 6
Channel to Monitor All

Common CPD Parameters
Observation Time Frame, sec 0.001
Training Sequence Length 200
Cool Off Len After Detection 0
Num Obs Aggr 100

CPD Method
○ Binary ● Basic

Spectral Method Param
H 0.2
F_low, Hz 50
☑ Report Spectrum

Basic Method Param
Avg. Tail 0.99
C 0.25
H 100
H Lock 70

Binary Method Param
DB 0.2
H 150

X PDF Calc Param
Observations Window Size 1000 □ Infinite
Granularity 20

Monitor

Channel 0

Show
☑ X
☑ A
☑ W
☑ Max W

Cur Val
X=2.4 E+2
A=3.2 E+2
W=0.0 E+0

CPD Server Session Stopped due to EOF. Ready to Start

FIGURE 1 Graphical user interface with real attack and its detection by the hybrid anomaly-signature intrusion detection system [Colour figure can be viewed at wileyonlinelibrary.com]

2 | ILLUSTRATIVE CONTEXTS

2.1 | Network security

Malicious intrusion attempts such as spam campaigns, phishing, personal data theft, worms, distributed denial-of-service (DDoS) attacks, man-in-the-middle attacks, and fast flux occur practically every day and have become commonplace in contemporary computer networks. These threats can incur significant financial damage and are a severe harm to the integrity of personal information. It is therefore essential to devise automated techniques to detect such events as quickly as possible so that an appropriate response can be provided and the negative consequences for the users are eliminated. Moreover, even routine behavior of users could generate anomalous events requiring the attention of network operators and managers. A good example would be flash crowds. Efficient operation and management of computer networks depend heavily on a relatively precise analysis of anomalies produced by both malicious and legitimate normal behavior.

The ability of change point detection techniques to run at high speeds and with low detection delays presents an interesting opportunity. What if one could combine such techniques with others that offer a very low false alarm rate (FAR) but are too heavy to use at line speeds? Do such systems exist? How can they be integrated? Figure 1 shows a graphical user interface with the results of detecting a real DDoS attack with false alarm filtering using the hybrid anomaly-signature intrusion detection system that will be described in detail in Section 3.3.

2.2 | Data networks

Data networks provide an infrastructure to rapidly move information around the globe. An information technology (IT) group is usually responsible for monitoring the health of a data network and maintaining its ability to provide the services it delivers. Within this context, data network surveillance means defining metrics that describe the health of a data network and using them to detect unusual patterns that might indicate a failure in the network.

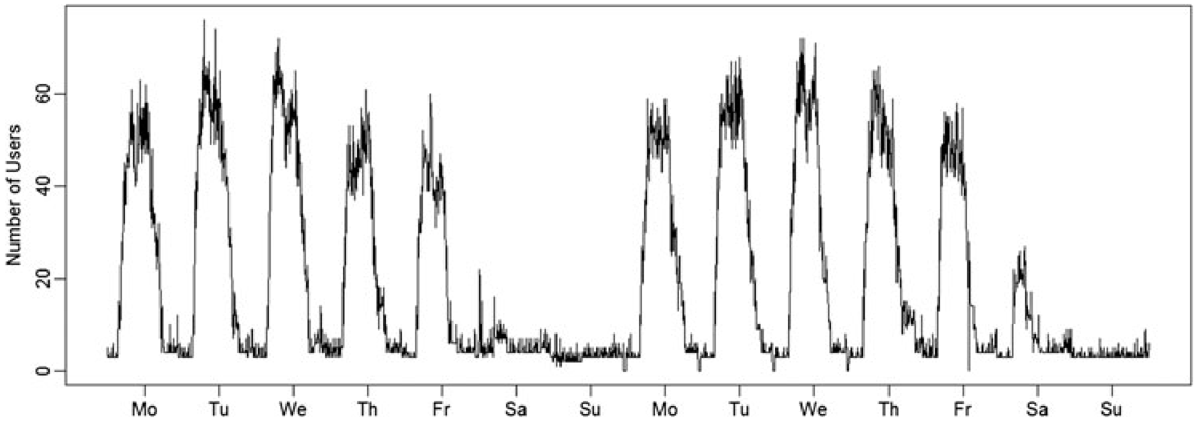


FIGURE 2 Two weeks of 5-minute counts of users logged into a data network server

Often the metrics that will be used to monitor the health of a data network will be various measures of traffic flow over the network. Figure 2 from the work of Fu and Jeske[1] illustrates 5-minute counts of the number of users on a particular data network server. The volume of traffic that crosses over a network link during a specified time of day is another example. We return to this example in Section 4.3.

2.3 | Dynamic networks

Some applications require monitoring a dynamic network or a network whose topology changes through time. In particular, a dynamic network is a temporally ordered sequence of networks whose node set and/or edge set varies through time. Dynamic network models provide insights about the relational structure of evolving complex systems. These models are commonly used in a variety of fields, ranging from political science to analyze the polarization of US Senators[2] to biology for analyzing gene-to-gene interactions over time and their relationship with breast cancer.[3]

Dynamic network models have been particularly useful in analyzing social dynamics among groups of individuals. For a concrete example, consider the Enron email network illustrated in Figure 3**.** Nodes in this network represent 184 employees at Enron, and edges between a pair of nodes quantify the number of emails exchanged between the 2 employees. One can readily see from Figure 3 that the email communication among employees dramatically changed between June 1999 and June 2001. Notably, in 2001, the network was much more densely connected than at other points in time. It turns out that this high level of email activity occurred at the time at which fraud investigations of the company began and activity subsided with the ultimate filing of bankruptcy from the company in December of 2001. By monitoring the global and local connections in the Enron dynamic network, one can readily detect this anomalous behavior among the Enron employees. We return to this example in Section 5.3.

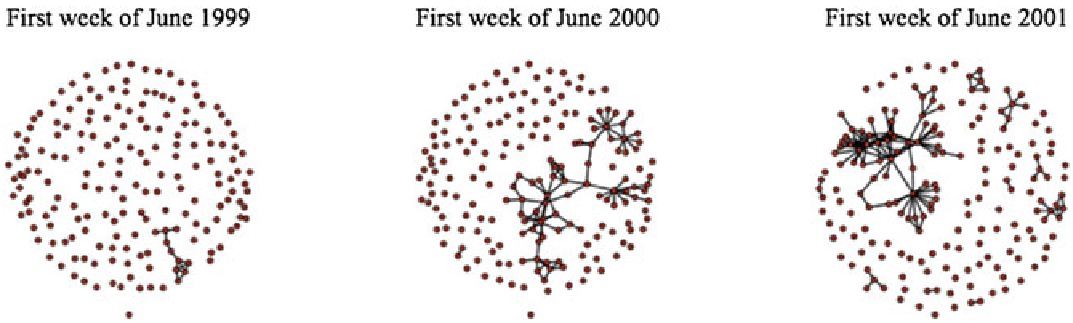


FIGURE 3 Weekly snapshots of the Enron email communication network [Colour figure can be viewed at wileyonlinelibrary.com]

3 | MONITORING NETWORK SECURITY

3.1 | Objectives

Detection of traffic anomalies is performed by employing *intrusion detection systems* (IDSs). Such systems in one way or another capitalize on the fact that maltraffic is noticeably different from legitimate traffic. Depending on the principle of operation, there are 2 categories of IDSs: either signature or anomaly based. For an overview, see the works of Debar et al[4] and Kent.[5] A signature-based IDS inspects passing traffic with the intent to find matches against already known malicious patterns. In contrast, an anomaly-based IDS is first trained to recognize the normal network behavior and then watches for any deviation from the normal profile, classifying deviations as potential attacks.[6-10]

As an example, consider DDoS attacks. These DDoS attacks typically involve many traffic streams resulting in a large number of packets aimed at congesting the target's server or network. As a result, these attacks usually lead to abrupt changes in network traffic and can be detected by noticing a change in the average number of packets sent through the victim's link per unit time. Therefore, it is appealing to formulate the problem of detecting DDoS as a quickest change point detection problem: to detect changes in statistical models as rapidly as possible (i.e., with minimal expected delays) while maintaining the FAR at a given level.

Currently, both anomaly- and signature-based IDSs are plagued by a high rate of false positives and are susceptible to carefully crafted attacks that “blend” themselves into normal traffic. Clearly, these 2 systems are complementary, and neither alone is sufficient to detect and isolate the myriad of malicious or legitimate network anomalies. For this reason, many different types of IDSs have been developed, each better suited for a particular attack type. As network speeds increase and applications get more complex, rapid intrusion detection with a low FAR becomes increasingly more difficult.

Solutions must focus on 2 main objectives: (i) development of an efficient adaptive anomaly-based IDS based on change detection techniques and (ii) integration of 2 detection techniques—anomaly IDS and signature-spectral detection techniques. The resulting hybrid anomaly-signature IDS is synergistic and performs better than any of the individual systems alone. The hybrid anomaly-signature IDS, described in Section 3.3, was tested on real attacks, and the results demonstrate the benefits of integrating anomaly and signature IDSs.

3.2 | Literature review

Typical computer network attacks include IP fragments, malformed packets, SYN floods, ICMP redirect messages, perpetual echo, restricted IP options and restricted IPs. IDSs also have to be capable of detecting other unwanted activities such as scans and traffic regulation anomalies for TCP and UDP. As we discussed in Section 2.1, the 2 main classes of detection methodologies are signature-based and anomaly-based. Both classes of systems have certain advantages and disadvantages.[4-6,11]

Specifically, signature-based detection technologies have little understanding of many network or application protocols and cannot track and understand the state of complex communications. They also lack the ability to remember previous requests when processing the current request. This limitation prevents these methods from detecting attacks that comprise multiple events if none of the events contains a clear indication of an attack. Signature methods are also unable to detect attacks within encrypted network traffic, including VPN, HTTPS, and SSH sessions. Performance evaluations showed that such IDSs are sensitive to both packet and ruleset content. For example, analysis of SNORT (www.snort.org) shows that as much as 31% of total processing is due to string matching, and in the case of Web-intensive traffic, this cost is increased to 80% of the total processing time. Clearly, this is a very serious drawback for ultra-high-speed networks.

Anomaly-based detection is the process of comparing definitions of what activity is considered normal against observed events to identify significant deviations. The major benefit of anomaly-based detection methods is that they can be very effective at detecting previously unknown threats. However, anomaly IDSs often produce many false positives, especially in more diverse or dynamic environments. Another noteworthy problem is that it is often difficult to determine why a particular alert was generated and to validate that an alert is not false.

For DDoS detection, Cheng et al[12] examined individual flows and applied spectral analysis to characterize periodicities and to separate normal TCP traffic during a DDoS attack. Barford et al[13] used wavelets to investigate anomaly detection techniques that make use of IP flow-level and SNMP information to identify frequency characteristics of DDoS attacks and other anomalous network traffic. Hussain et al[14] used signal processing techniques to differentiate between single-source and multi-source DDoS attacks but examined attack traffic in isolation of background traffic. Lakhina et al[15] discovered anomalies in network traffic by studying entropy in packet IP addresses and ports. Partridge et al.[16] used signal processing techniques to analyze wireless traffic. Wavelets have also been used to study self-similarity in network traffic and to detect some network problems.[17] Li and Lee[18] utilized energy distribution-based wavelet analysis to detect DDoS attack traffic.

The results of the work of He et al[19] suggest that a careful choice of training approaches yields good detection performance at moderate "signal-to-noise ratio" (SNR), where traffic of interest is only 5% to 10% of the total traffic, and excellent detection performance (98%) at SNR where target traffic is over 10% of total. Additional research has further studied these issues.[11,20-24]

For cyber-security applications, several nonparametric and adaptive parametric change detection methods based on cumulative sum (CUSUM)–type and Shiryaev-Roberts (SR)–type statistics were developed and tested based on real data by Tartakovsky,[7,25] Tartakovsky and Polunchenko,[26] Tartakovsky and Veeravalli,[8,27] Tartakovsky et al,[9,10,28-30] and Polunchenko et al.[31] A sample of results from these works is presented in the next section.

Research has shown that neither anomaly-based nor signature-based security solutions provide adequate protection by themselves.[32,33] Hence, a new approach is required. Such an approach based on combining statistical change detection methods with spectral-signature signal processing techniques is discussed in Section 3.3.2.

3.3 | Spotlight method

In this section, we outline approaches for designing 2 efficient IDSs (the anomaly IDS and hybrid anomaly-signature IDS) proposed by Tartakovsky,[7,25] Tartakovsky and Polunchenko,[26] Tartakovsky and Veeravalli,[8,27] Tartakovsky et al,[9,10,28-30] and Polunchenko et al.[31] Results from testing the IDSs on data sets that captured real attacks are also presented.

3.3.1 | Change point detection-based anomaly IDS

We begin by describing the adaptive semiparametric detection method proposed in the works of Tartakovsky et al,[9,10,28-30] which is applicable to the detection of any changes of statistical patterns in the monitored data flows, both abrupt and gradual. It is capable of detecting a wide variety of internal and external intrusions, including stealthy attacks, with small delays to detection for a fixed prescribed FAR.

Consider a multichannel (or multistream) scenario where data $\boldsymbol{X}_n = (X_n^{(1)}, \dots, X_n^{(N)})$, $n \geq 1$, are used for identifying the presence of anomalies. Here, $X_n^{(i)}$ is a sample obtained at time n in the ith channel. The importance of using multiple channels for detecting DDoS attacks has been shown.[8-10] For example, in the case of UDP flooding attacks, the channels correspond to packet sizes, whereas, for TCP SYN attacks, they correspond to IP addresses. When the prechange and the postchange models are completely specified, efficient detection procedures with certain optimal properties can be constructed based on the log-likelihood ratio (LLR)–based CUSUM and SR rules (see, eg, the works of Tartakovsky et al[29]). However, in intrusion detection applications, these models are usually unknown. For this reason, we undertake a semiparametric approach. More specifically, the unknown LLRs in channels are replaced by appropriate score functions $S_i(n)$, $i = 1, \dots, n$ that have negative mean values $E_\infty[S_i(n)]$ before the change occurs and positive mean values $E_k[S_i(n)]$ after the change occurs. Here, E_k stands for expectation when the point of change is k and E_∞ corresponds to the no-change scenario.

While no assumptions are made in terms of probability distributions, some assumptions on the change should be made. In intrusion detection applications, the detection problem can be usually reduced to detecting changes in mean values or in variance or in both mean and variance. In the works of Tartakovsky et al,[9,10] a linear memoryless score was introduced for detecting changes in the mean, and in other works of Tartakovsky[25] and Tartakovsky et al,[28,29] this score was generalized to linear quadratic in order to be able to handle changes in both mean and variance.

Specifically, let $\mu_{i,\infty} = E_\infty[X_n^{(i)}]$, $\sigma_{i,\infty}^2 = \mathrm{Var}_\infty[X_n^{(i)}]$ and $\theta_i = E_0[X_n^{(i)}]$, $\sigma_i^2 = \mathrm{Var}_0[X_n^{(i)}]$ denote the prechange and postchange mean values and variances in the ith channel. Write $Y_n^{(i)} = (X_n^{(i)} - \mu_{i,\infty})/\sigma_{i,\infty}$ for $i = 1, \dots, N$, introduce the following linear-quadratic score functions

$$S_n^{(i)} = a_i Y_n^{(i)} + b_i \left(Y_n^{(i)}\right)^2 - c_i,$$

where a_i, b_i, and c_i are the design parameters. Introduce recursively the score-based CUSUM and SR statistics

$$W_n^{(i)} = \max\left\{0\,, W_{n-1}^{(i)} + S_n^{(i)}\right\} \quad , \quad R_n^{(i)} = \left(1 + R_{n-1}^{(i)}\right)\, \exp\!\left(S_n^{(i)}\right).$$

Typically, these statistics remain close to zero in normal conditions; when the change occurs in the *i*th channel, the *i*th statistics starts rapidly drifting upward eventually crossing a threshold, at which time the change is declared.

The MAX algorithm[9,10] is based on the maximal statistic, $W_{\max}(n) = \max_{1 \le i \le N} W_n^{(i)}(n)$, which is compared to a threshold that controls the FAR, i.e., the algorithm stops and declares the attack at

$$T_{\max}(h) = \min \{ n : W_{\max}(n) \ge h \} \quad , \quad h > 0.$$

This method shows very high performance and is the best one can do when attacks are visible in either one or very few channels. The most general and realistic case is where the number of affected channels is a priori unknown and may vary from small to large. This challenging problem was considered in the work of Tartakovsky et al[29] where several asymptotically optimal likelihood ratio-based detection procedures were suggested for known prechange and postchange models. When models are unknown, similar procedures can be used with the LLRs replaced with the scores. In particular, the reasonable detection statistic is $\sum_{i=1}^{N} W_n^{(i)}(n)$.

A similar approach can be used to form the SR-type multichannel detection procedure given by the stopping time

$$T_{\mathrm{SR}} = \min \left\{ n : \sum_{i=1}^{N} \log R_n^{(i)} \ge h \right\}.$$

Yet another approach is to exploit a nonparametric algorithm with binary quantization and optimization of the quantization threshold. In this case, it is possible to implement optimal binary quantized CUSUM and SR algorithms that are based on true likelihood ratios for Bernoulli sequences at the output of quantizers.[25]

Note that the parameters $\mu_{i,\infty}$ and $\sigma_{i,\infty}$ are unknown and should be estimated from the data and the post parameters θ_i and σ_i^2 are usually unknown. In order to make the IDS fully adaptive, the following 2 procedures might be used for evaluation of unknown parameters, which is performed online. Since the estimation is performed identically for all channels, we omit the subscripts i,∞ when describing these procedures.

Fixed window zero-reflection estimation procedure: Form a pilot estimate based on initial data if available. Periodically reestimate, using a retrospective sliding window back from the point when the statistic W_n hits a zero reflecting barrier.

Exponentially weighted moving average (EWMA) estimation procedure: Define $\mu_n = \mu_{n-1}(1-\rho) + X_n \rho$ and $\sigma_n^2 = \sigma_{n-1}^2(1-\rho) + (X_n - \mu_n)^2 \rho$, where $0 < \rho < 1$ is a smoothing factor, which is usually taken in the range from 0.005 to 0.1. This procedure shows extremely high performance and, based on a preliminary analysis, is recommended for the implementation in the anomaly IDS.

Figure 4 illustrates the EWMA estimation procedure along with the behavior of the fully adaptive CUSUM statistic based on the linear-quadratic score with parameters estimated by the EWMA procedure for a real data set that captures an ICMP DDoS attack. The solid black region in the plot corresponds to the detection of the attack when the CUSUM exceeds its threshold and further resets to zero and continually flags the attack. Figure 4 shows that the proposed EWMA estimation algorithm allows us to track the change in the mean very accurately: both prechange and postchange mean values are estimated very accurately and the detection statistic increases very rapidly after the attack starts while immediately decreases after the attack stops. However, we can also see quite a few false alarms prior to the attack and in the post-attack segment. These false alarms are being filtered by the spectral algorithm at the second stage in the hybrid IDS, as will be discussed in Section 3.3.2.

Figure 5 compares the multicyclic adaptive CUSUM and SR detection procedures (renewed after each alarm) for yet another real data set. This data set contains real background traffic and a UDP packet storm DDoS attack. The SR procedure detects this attack slightly earlier than CUSUM, but the difference is almost negligible.

It is impossible to develop a single detection algorithm that is optimal in all possible conditions. The reason for this is that the network environment is highly changing and both legitimate traffic models and attack models are not completely specified. Therefore, a real IDS has to exploit a bank of detection filters that includes the algorithms described above. This bank should also be supplemented with a nonparametric multichannel algorithm based on the optimal binary quantization. In the interest of brevity, we omit further details regarding the latter algorithm.

3.3.2 | An adaptive hybrid IDS

This section describes a hybrid approach to network intrusion detection, proposed by Tartakovsky,[25] that can effectively deal with stealthy (slow and low-contrast) attacks. The system has a 2-stage cascade architecture, utilizing the change point detection methodology for preliminary detection of attacks and a discrete Fourier transform or a wavelet transform to reveal periodic patterns in network traffic, which are then used to confirm the presence of attacks and reject false positives prior to attack occurrence. In other words, the methodology is based on using the change point detection method

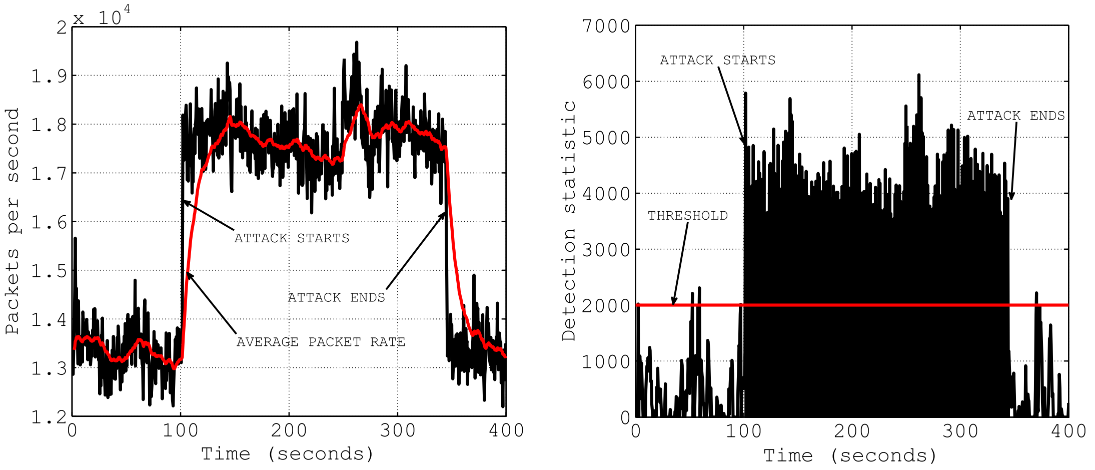


FIGURE 4 Exponentially weighted moving average estimate of the mean value (left) and the adaptive cumulative sum statistic (right) for an ICMP distributed denial-of-service attack [Colour figure can be viewed at wileyonlinelibrary.com]

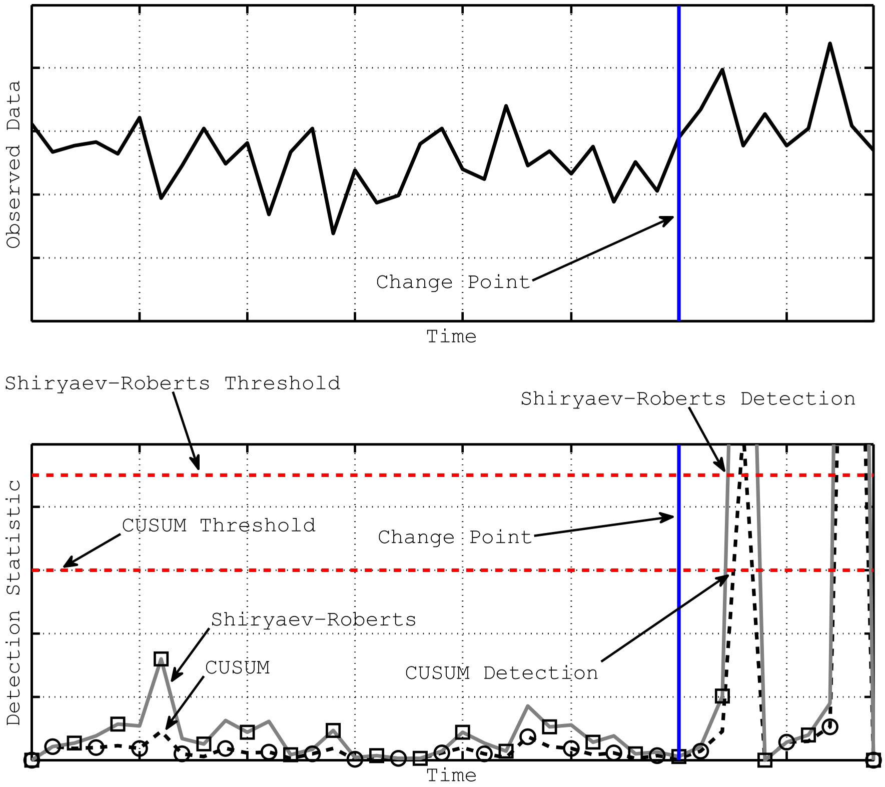


FIGURE 5 Top—real data; bottom—multicyclic CUSUM statistic (black) and log of SR statistic (gray) [Colour figure can be viewed at wileyonlinelibrary.com]

for preliminary detection of attacks with low threshold values. When detection thresholds are low, the anomaly IDS produces intense flows of false alarms. However, these frequent false alarms can be tolerated, since they do not lead to real false alarms that pass the whole system. Once an alarm is raised, a spectral analyzer is triggered. This alarm is either rejected or confirmed as a true detection, in which case, a final alarm is raised.

We begin with explaining the idea of flow-based signature detection techniques that are used for false alarm filtering. These techniques examine patterns embedded in packet arrivals rather than packet contents. Note that this is different than anomaly detection systems that examine packet and bit rates, protocol, port and address decomposition, and daily variations of such quantities. In the signature-spectral approach, we first define important events, then create time series of these events, and finally apply spectral analysis techniques on the time series to characterize traffic. Examples include

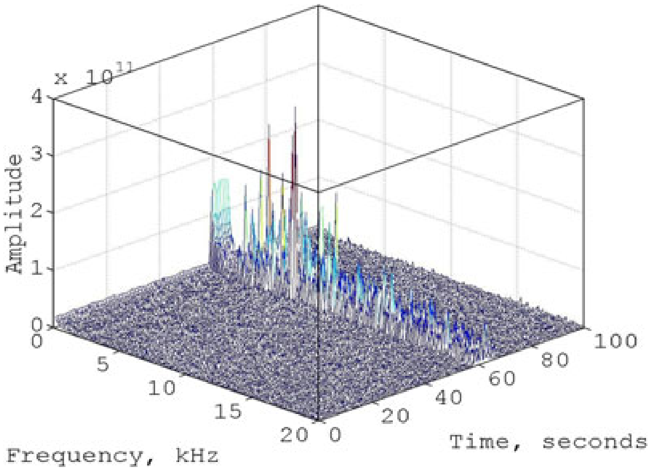


FIGURE 6 Power spectral density for a UDP distributed denial-of-service attack [Colour figure can be viewed at wileyonlinelibrary.com]

simple events such as packet arrivals and retransmissions, but also higher-level events such as connection attempts and failed service requests. Detecting a higher-level event is preferable for complex traffic because it reduces false positives and processing loads.

We now illustrate this idea using an example of an *SSH* dictionary attack. The spectral detection approach works as follows. When an *SSH* dictionary attack takes place, a large amount of *SSH* authorization requests is directed to the target server from the hacker. These requests are usually sent in a periodic manner. This time regularity is mixed with other non-attack traffic toward the target's network but, nonetheless, can be readily detected by spectral analysis because it causes a spike in energy at the frequency of the victim's link.

Another typical example where a spectral approach can be effectively used is detecting DDoS attacks. A DDoS attack sends a large number of packets from several compromised machines (called zombies) with the goal of knocking the target off the network or seriously degrading the service it provides (saturate links). The attack is typically easy to detect at the target where its strength is the highest and sometimes at the source where each zombie blasts the target as fast as possible. However, recently, a new attack tactic has emerged that produces stealthy attacks. Specifically, a very large set of zombies is used, but each zombie attacks with a very low packet rate in order to hide the traffic from the local IDS. Spectral techniques can be used to detect *stealthy low-rate attacks*, especially when combining with change point detection methods.

Yet another problem is *encrypted* attacks. Traffic can be obscured if it is either encrypted or if it uses a proxy. Encrypted or proxied traffic does not pose a problem for spectral analysis techniques. Since we do not rely on packet contents, spectral techniques work with any type of traffic.

The spectral-signature technique requires a time series of packet arrivals. The power spectral density (PSD) for stationary segments is computed by performing a discrete-time fast Fourier transform (FFT) on the autocorrelation function of the attack stream, which is a measure of how similar the attack is to itself shifted in time by a certain offset. The PSD captures the power or strength of individual observable frequencies embedded in the time series formed by the observations. Figure 6 plots the FFT (i.e., PSD vs frequency) for the real network traffic that contains a UDP packet storm DDoS attack. Normal traffic does not produce visible peaks, while when the UDP storm starts a contrast peak immediately appears.

As we stated above, this spectral signature method is combined with the change point anomaly detection method for designing a hybrid anomaly-signature IDS with false alarm filtering and true attack confirmation capability.

To summarize, the hybrid IDS is based on the following principles.

1. *Anomaly IDS—quick detection with high FAR:* In order to detect attacks quickly, the detection threshold in the change point detection module is lowered, which leads to frequent false alarms that are filtered by a signature-spectral IDS block. Thus, the change point detection block is used for quick detection with relatively high FAR and for triggering spectral analysis algorithms.

2. *Signature IDS—false alarm filtering*: To reject false detections, a spectral-based approach is used, eg, Fourier or wavelet spectral analysis module. Therefore, the spectral-signature IDS block is used for false alarm filtering/rejection and true attack confirmation.

As mentioned above, detecting intrusions rapidly and with low intensity of false positives becomes exceedingly harder when attackers use encryption or when attack and legitimate traffic are mixed behind a proxy. Most traditional

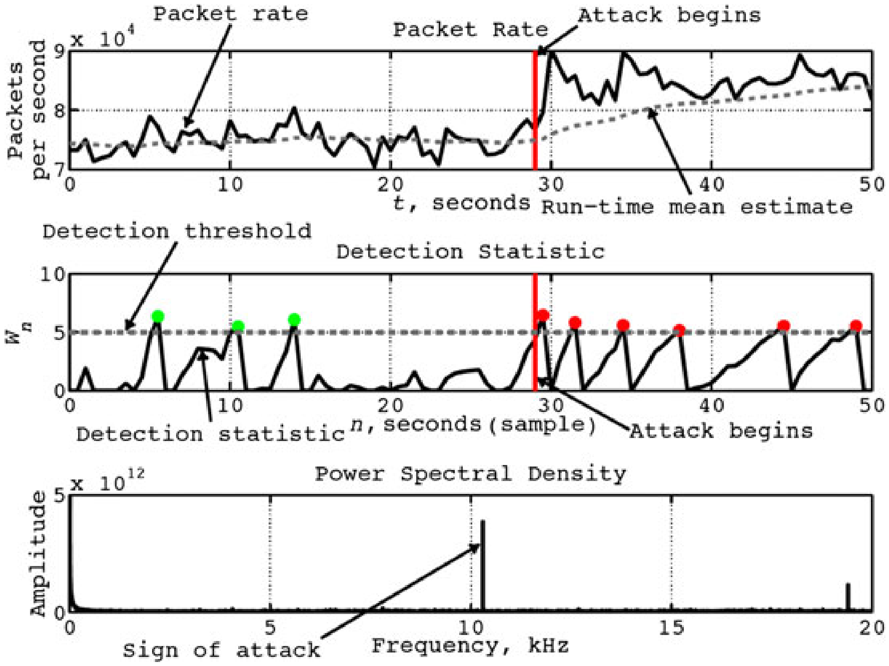


FIGURE 7 Adaptive hybrid intrusion detection system in action for the UDP storm [Colour figure can be viewed at wileyonlinelibrary.com]

signature-based intrusion detection techniques fail. However, encrypted/proxied traffic does not cause any problem for the hybrid anomaly-spectral IDS. Since we do not rely on packet contents, our techniques work perfectly well.

We now present sample testing results reported in Tartakovsky[25] that illustrate the efficiency of the hybrid IDS. We analyze a particularly interesting dataset with a very short DDoS attack on the UDP port 22. The trace was captured on one of the Los Nettos (a regional Internet service provider in LA) private networks (LANDER project). The attack begins about 60 seconds after the beginning of the trace and consists of very short packets (about 15 bytes in size) sent to the victim's UDP port 22 at a rate of about 180 Kbps with the background traffic being about 53 Kbps. The attack is very short (only 10 seconds), which is a challenge for any IDS.

Figure 6 shows spectrum (PSD) for this dataset. The contrast peak in the middle of the plot suggests that this might indeed be an attack. It is this phenomenon that we use to filter false positives in the hybrid system and confirm true attacks.

Figure 7 illustrates the adaptive hybrid IDS at work. This IDS exploits a fully adaptive score-based CUSUM algorithm (with a linear-quadratic score) that constitutes the basis of the anomaly IDS module and an FFT-based spectral-signature module. The first plot shows raw data (packet rate). It is seen that there is a slight change in the mean (and in variance), which is barely visible. The second plot shows the behavior of the multicyclic adaptive CUSUM statistic, which is restarted from scratch (repeated) every time a threshold exceedance occurs. The third plot shows the PSD at the output of the spectral module. The peak appears only when the attack starts (which confirms the attack), while previous threshold exceedances (false alarms) shown by green dots are rejected by the spectral analyzer. The true detections are marked by red dots. Note that the spectral module is triggered only when a threshold exceedance occurs in the anomaly IDS module, which is important for the real-time performance in ultra-high-speed networks since FFT is impossible to implement at Gb rates while CUSUM and SR statistics for change detection are easy to compute at any rate.

3.4 | Open research areas

Since computer networks are very complex and large today, tools that can help diagnose problems in the network, in particular, extract hidden patterns to understand phenomena related to network anomalies are very important. Statistical change point detection and spectral signal processing techniques have great potential in creating such powerful tools. A link is saturated when the offered load at the link exceeds its capacity. A saturated link is usually the bottleneck and it typically implies an abnormal condition in the network.[19] DDoS attacks, considered above, saturate a link near the victim and are only one example of the source for saturation. The results show that the change detection and spectral algorithms can detect the presence of a bottleneck link very fast even for low SNR values when traffic through the bottleneck link is a small portion of the total traffic at the monitoring point. However, the techniques have to be strengthened in the following directions:

1. To model the processes that govern the generation of bottleneck traffic signatures and use these models to design more sophisticated detection algorithms that take time-varying and correlated factors into account. In particular, more general statistical models with dependent observations governed by hidden Markov models are in order.
2. To apply the detection algorithms in more diversified environments, including different monitoring points and different bottleneck locations.
3. To study other periodic patterns such as protocol behavior, spam campaigns, and unauthorized break-ins.

Most organizations run some version of spam filters at their local networks.[34] These filters examine the content of each message as well as the IP address where the message came from, and if they match known spam signatures, the message is marked as spam. These techniques work quite well, but they are typically expensive both in initial and operational costs. In addition, block lists rely on information that was gathered ahead of time and thus might be stale.

Fighting spam at the network level and looking for spam behavior is a challenge. Monitoring network traffic has several advantages: (i) it requires no message content examination and thus guards privacy; (ii) spammers can be detected almost instantly based on their network behavior; (iii) collateral damage is reduced because dynamic addresses released by spammers can be removed from block lists quickly; and (iv) IP addresses can be blocked before connections are accepted, saving resources at the mail server.

An important question is what features are useful for detecting spammers? We propose to investigate such features as the autonomous system the IP address belongs to, message size, blocked connections, and message length, which can be determined from network traffic. Then, change point detection methods can be used to detect when traffic patterns from a particular host match known spammer patterns. Combining these features is, of course, important, and multiple detection processes may need to be active at the same time. In summary, change point detection techniques can be used to learn and detect patterns in network traffic from spammers. Detecting spammers at the network level has several advantages, such as no privacy issues, near real-time detection, and minimizing collateral damage. This is an open and novel research area.

Yet another open challenging problem is the rapid detection of unauthorized break-ins. Indeed, unauthorized tampering with, or breaking into, a system represents a high computer security risk. Such a scenario usually involves 2 stages. In the first stage, the hacker launches a dictionary attack attempting to guess a username and password. In the second stage (assuming that the hacker was successful in gaining access to the machine), the hacker performs suspicious activities on the machine, including downloading malware and opening up a backdoor. Potentially, both stages can be detected by the anomaly IDS or anomaly-signature IDS. The initial study was performed by Tartakovsky,[25] where an approach correlating changes in network traffic to detect attempted and successful break-ins was proposed. Attempted break-ins are detected by searching for traffic patterns corresponding to dictionary attacks. Successful break-ins are detected by noting when a user successfully logs in (signified by a successful connection and exchange of application data), and then detecting subsequent suspicious network activity using the anomaly-signature IDS. The important problem of how to distinguish malicious activity from normal user logins remains open.

4 | MONITORING RELIABILITY OF DATA NETWORKS

4.1 | Objectives

Surveillance of data networks involves the use of algorithms to determine when a perceived anomaly in the monitored metrics is a significant departure from what could be expected due to natural randomness. An algorithm might be constructed for each metric that will be monitored, and for each metric, a threshold is also specified. When a metric crosses its threshold, an alarm is raised that alerts the IT staff to the possibility of a failure. Investigations then proceed to determine if the alarm is a false alarm or if a failure condition exists that needs to be cleared. Effective data network surveillance algorithms maintain a FAR that is manageable for the IT staff and, at the same time, give fast signals of true out of control conditions.

4.2 | Literature review

Data network surveillance algorithms can be rule-based or statistical-based. In rule-based approaches, the formulation of metrics and their thresholds rely on subject matter expertise from data network managers. Barford et al[13] mentioned that a common technique for handling data network surveillance is periodically plotting data and using subject matter expert rules to determine if those data are consistent with expectations. Rules derived this way can be effective but they can also be skewed by the experience of the subject matter experts involved. Feather et al[35] used historical data to establish in-control thresholds for the data stream. The thresholds are used to encode a feature vector that represents the current behavior of the system. The feature vector is input to a pattern matching system to determine if it resembles a pattern that is a priori known to be associated with a specific fault. The adequacy of this approach will rely heavily on how consistently a given fault will reproduce the same pattern and on the depth of the library of fault patterns.

Data network surveillance can alternatively be viewed through the lens of statistical process monitoring (SPM) methods. In this way, a variety of SPM tools become available for potential use, such as Shewhart, EWMA, and CUSUM control charts. The characteristics of data network traffic hamper a conventional use of these charts. Specifically, data network traffic is highly correlated and is nonstationary, and the data are often counts. The nonstationarity is seen in day-to-day and hour-to-hour trends. It is often possible to define a suitable level of periodicity in data network traffic. However, data network traffic metrics may not consistently follow a known distribution. Brutlag[36] discussed these points. Finding a data transformation that deals with these characteristics simultaneously is extremely challenging.

Thottan and Ji[37] presumed that a stream of data network traffic can be divided into batches that follow piecewise normal theory autoregressive models. Likelihood ratio tests (LRTs) on residuals from these models are used to detect changes in the data stream. The plausibility that observations within a batch can be modeled as a stationary normal process is a key assumption with this method. Cao et al[38] developed a state-space model to describe a time-varying data network stream. After an application-dependent transformation, monitoring statistics are considered as following a normal distribution with a constant variance. The time-dependent patterns in the data are captured by B-spline functions. A CUSUM statistic is used to monitor deviations of the data stream relative to baseline forecasts. A concern with this approach is that traffic counts with time-varying mean and variance are not easily transformed into data that follows a normal distribution.

Jeske et al[39] and Montes de Oca et al[40] defined a time slot structure on the data stream and assumed that after a suitable application-dependent transformation, the data within a time slot are independent and identically distributed (iid). Historical data are used to estimate the time slot distributions nonparametrically, and then, a CUSUM tracking statistic based on the empirical probability integral transformations is proposed. The plausibility of finding an iid inducing transformation within each time slot is a potential concern with this method.

Rather than trying to transform the data to accommodate model assumptions, a different approach directly models the data network traffic with discrete models. Lambert and Liu[41] used a time-varying negative binomial distribution for the data within each time slot. They argue that using time-varying parameters for the negative binomial distributions mitigates the need to account for correlation in the data. An EWMA tracking statistic based on an approximate probability integral transformation is proposed. A potential concern with this approach is the premise that modeling correlation in the data is not necessary.

Fu and Jeske[1] used a generalized linear mixed model (GLMM) to model the data stream. A similar time slot structure is used, with negative binomial distributions, but the time slots are associated with an autocorrelated sequence of latent random effects. Repeated LRTs are used to detect change. The importance of explicitly employing an autocorrelation structure in the random effects was demonstrated.

Readers are also referred to US patent literature for a number of additional monitoring methods. Marvasti and Jeske[42] and Gluhovsky et al,[43] and references therein, described a variety of approaches ranging from empirical based pattern recognition systems to systems that utilize statistical predictive modeling. The work of Gluhovsky et al[43] is an interesting implementation where probes are sent into the network to measure user experience with response times. The measured response times are modeled with generalized additive models, from which thresholds that control FAR are developed.

4.3 | Spotlight method

In this section, we elaborate on the methodology proposed in the work of Fu and Jeske.[1] In that paper, GLMMs were proposed as models for data network traffic counts. The random effects in GLMMs can be used to capture both overdispersion and autocorrelation in the data, and Poisson or negative binomial conditional distributions are a natural choice for count data

4.3.1 | Data network traffic model

Figure 2 introduced a data set that was used to motivate modeling counts with a Poisson GLMM that incorporates an hourly time slot structure. Let Y_{ij} denote the jth count with the ith time hour. Here, $i = 1, \dots, 168$ (the number of hours

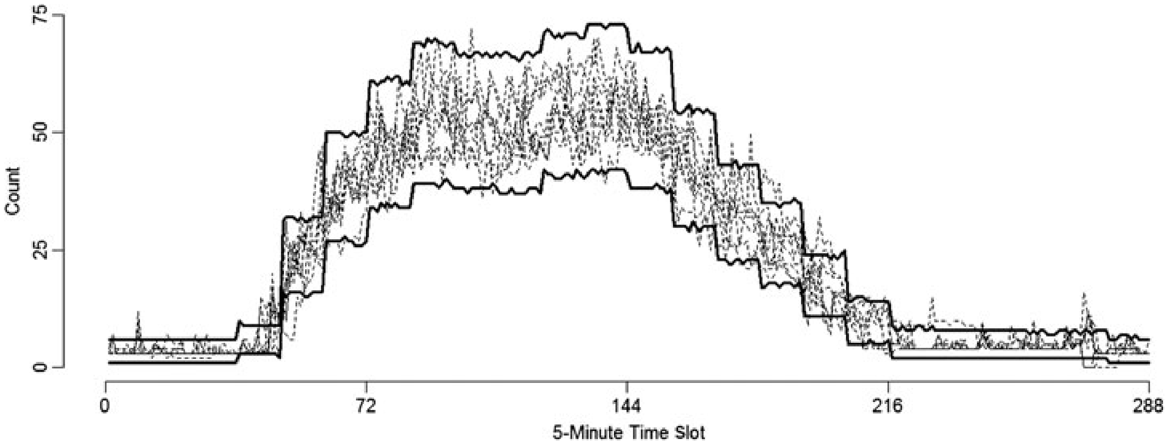


FIGURE 8 Eight Mondays of observed counts (dashed lines) with the lower and upper 10th percentiles of 1000 traces generated from the fitted generalized linear mixed model

in a week) and $j = 1, \dots, 12$ (the number of 5-minute measurements in a given hour). As an illustrative GLMM, let β_i be a fixed effect and S_i be a zero mean normally distributed random effect for hour i. Conditional on all the random effects $\boldsymbol{S} = (S_1, \dots, S_{168})'$, Y_{ij} are independently distributed as Poisson with means μ_{ij}, where $\log\mu_{ij} = \beta_i + S_i$. Observations in different weeks are modeled as independent, but observations within a week are correlated through a covariance structure selected for $\boldsymbol{S}$. With a simple autoregressive covariance structure, i.e., $\text{Cov}(S_i, S_{i'}) = \sigma^2\rho^{|i-i'|}$, it can be shown that $\text{Corr}(Y_{ij}, Y_{i'j'}) = \exp(\sigma^2\rho^{|i-i'|} - 1)/\sqrt{p_i p_{i'}}$, where $p_i = \exp(-\beta_i - \sigma^2/2) + \exp(\sigma^2) - 1$.

Figure 8 demonstrates the goodness of fit of the Poisson GLMM using a sample of 8 traces (light gray lines) of Monday counts. Based on those data, the fitted Poisson GLMM yielded $\hat{\sigma}^2 = 0.017$, $\hat{\rho} = 0.656$, and estimates $\left\{\hat{\beta}_i\right\}_{i=1}^{24}$ (starting midnight: 1 am) of 1.28, 1.23, 1.18, 1.75, 3.13, 3.61, 3.83, 3.96, 3.93, 3.92, 3.99, 4.01, 3.93, 3.71, 3.46, 3.24, 2.84, 2.27, 1.64, 1.61, 1.56, 1.51, 1.52, and 1.31. The dark lines in Figure 8 are the lower 10th and upper 90th percentile at each time point of 1000 simulated traces from the fitted Poisson GLMM. Approximately, 84% of the data falls between the model-based percentiles, indicating only slightly more variability in the model than what is represented in the sample of data.

4.3.2 | Tracking statistic

We describe the tracking statistic identified as T_n^{JLR} in the work of Fu and Jeske.[1] This tracking statistic is motivated as an approximation to a repeated joint LRT based on the concept of h-likelihood.[44] The tracking statistic starts off each monitoring week with a value of zero. It is designed to accumulate values throughout the week in such a way that negative values are expected when observed traffic patterns match what are considered normal traffic patterns and increasingly large positive values are expected when abnormal traffic patterns are observed. To avoid build-up of negative values during the time the network is behaving as expected, the tracking statistic is truncated below at zero. In our example, the tracking statistic is evaluated every 5 minutes when a new count of users on the server becomes available. When T_n^{JLR} crosses a specified threshold, an alarm is triggered for follow-up investigation.

It is assumed there is sufficient in-control data to estimate the fixed effects $\{\beta_i\}_{i=1}^{168}$ and (σ^2, ρ) with negligible sampling errors, and therefore, these parameters can be assumed to be known. The algorithm is designed to detect when the fixed effects increase by a multiplication factor c. That is, the out of control traffic model is obtained by replacing the $\{\beta_i\}_{i=1}^{168}$ parameters in the in-control model by $\{c\beta_i\}_{i=1}^{168}$, where c is some constant, $c > 1$ indicating interest in detecting upward shifts in the counts. Straightforward modifications to the tracking statistic can be made if interest is in detecting downward shifts in the counts.

Let $1 \leq n \leq 2016$ be the current observation time during a monitoring week and let T_n^{JLR} denote the value of the tracking statistic. Define $1 \leq r_n \leq n$ to be the index of the first observation of the monitoring week after the most recent reset to zero. The observations that go into the calculation of T_n^{JLR} are $\{y_t\}_{t=r_n}^{n}$. Let $\boldsymbol{S}^*$ denote the vector of distinct random effects of the monitoring week for the time slots from which $\{y_t\}_{t=r_n}^{n}$ emanated. Note that, since the number of observations in the monitoring week that are used at any given time is $n - r_n + 1$, the dimension of $\boldsymbol{S}^*$ will usually be small. The value of the tracking statistic at time n is

$$T_n^{JLR} = \max\left\{0, \log \frac{\max\limits_{\boldsymbol{s}^*}\left(\prod\limits_{t=r_n}^{n} f\left(y_t|c\beta_{i_t}, s_{i_t}\right) g\left(\boldsymbol{s}^*|\sigma^2\rho\right)\right)}{\max\limits_{\boldsymbol{s}^*}\left(\prod\limits_{t=r_n}^{n} f\left(y_t|\beta_{i_t}, s_{i_t}\right) g\left(\boldsymbol{s}^*|\sigma^2\rho\right)\right)}\right\},$$

where $1 \leq i_t \leq 168$ denotes the time slot that observation y_t comes from, $f(y_t \mid \beta_{i_t}, s_{i_t})$ is the Poisson probability function with mean $\exp(\beta_{i_t} + s_{i_t})$ evaluated at y_t, and $g(\boldsymbol{s}^* \mid \sigma^2, \rho)$ is the multivariate normal probability density function of $\boldsymbol{S}^*$.

4.3.3 | Implementation

An alarm indicating a potential out of control network condition is triggered when $T_n^{JLR} > h$, where h is selected to control the FAR. A simple way to determine h is to simulate network traffic from the fitted in-control GLMM for a large number of monitoring weeks, and for each week identify the maximum of the T_n^{JLR} values. Using the upper α percentile of these maximum values as h ensures a FAR of α.

Fu and Jeske[1] discussed a simulation study that shows T_n^{JLR} has good sensitivity for detecting injected traffic pattern anomalies. They also provide guidelines on how to determine when the size of the in-control data is adequate for fitting the GLMM with negligible sampling error. The in-control data sets need to be large, on the order of tens of thousands of observations, but this is consistent with most SPM applications that use Phase I data sets to estimate in-control parameters.[45,46] In terms of computational complexity, the Monte Carlo method for finding h can take 2 to 3 hours if using a single processor computer. However, the calculation can be done offline. Computing the tracking statistic T_n^{JLR} at each time point is an online calculation and can be done within 5 ms.

4.4 | Open research areas

A GLMM appears to be a realistic and flexible model for capturing a variety of data network traffic patterns. On the downside, it is a large model in terms of the number of parameters. A simpler tracking statistic that does not involve computing LRTs, and/or a more parsimonious data network traffic model that lends itself to a simpler tracking statistic would be useful.

Nonparametric data network surveillance methods would be a useful area to explore. The nonparametric CUSUM method in the work of Jeske et al[39] is a start in this regard, but it does not handle an unspecified autocorrelation structure. Recent work using discrete wavelet transformations seems promising.[47]

The number of potential metrics to be used by a data network surveillance system could be very large if consideration is given to measuring server metrics and link utilization metrics everywhere in the network. The data network surveillance methods that have been discussed here are all univariate. Additional research on how to implement a dimension reduction strategy to downsize the number of monitored metrics, and/or how to control an inflated FAR that will result from unadjusted univariate monitoring would be useful, as would be some research that links FAR to the traditional SPM average run length performance metric.

5 | MONITORING FOR CHANGE IN DYNAMIC NETWORKS

5.1 | Objectives

As we have seen, the term network surveillance has different connotations in different contexts. In this section, we regard a network as a system of interactions between several actors. The actors may be humans in a social network, genes in a biological network, or neurons in a network of the brain. Recognizing that the pattern of interactions in such networks may evolve over time, we refer to them as dynamic networks. In this situation, interest lies in identifying instances or periods of unusual levels of interaction among the actors in a network, and we use the term network surveillance to describe the collection of statistical strategies that do exactly this.

To this end, we mathematically represent a network with n actors as a graph with n nodes or vertices, where an edge between 2 nodes signifies interaction between them. In the context of network surveillance, we assume that temporal "snapshots" of the dynamic network are available, and we let $G_t = ([n], W_t)$ denote a network at time point t. Here, $[n]$ represents the collection of nodes and $W_t = \{w_t(u, v) : u, v \in [n]\}$ represents the edge weights that quantify the strength of the relationship between nodes $u, v \in [n]$. Depending on the context and available data, the edge weights $w_t(u, v)$ may be recorded as binary indicators taking on the value of 1 only if nodes u and v share a specified level of interaction at time t. In other situations, the edge weights may be discrete valued and count the number of interactions between nodes u and v at time t. Here, we consider undirected graphs, meaning that no information regarding the direction of interaction is stored in the edge weights.

Supposing we observe a dynamic network prospectively through time (i.e., $G_1, G_2, G_3, \ldots$), network surveillance provides a formal methodology for identifying the time point(s), i.e., t^*, at which the level of interaction between a small or large number of actors in the network has significantly changed. Network surveillance strategies are typically used for the detection of periods of increased interaction, though they may also be used to identify unusually low levels of interaction as well.

In Section 5.2, we review a few existing methods of network surveillance, and in Section 5.3, we highlight one surveillance strategy in particular that follows a 2-stage framework based on SPM techniques. Specifically, we discuss the approach proposed by Wilson et al[48] that utilizes Shewhart and EWMA control charts to monitor parameter estimates from the degree-corrected stochastic block model (DCSBM)[49] for the fast detection of a variety of local and global network changes.

5.2 | Literature review

It is important to note a distinction between our focus of prospectively detecting change in a dynamic network as new information becomes available versus identifying a point in time when a significant change occurred in a dynamic network by considering only historical data. Notable works on the latter include the work of Peel and Clauset,[50] who described a Bayes factor testing approach under the generalized hierarchical random graph model, and the work of Bhamidi et al,[51] who investigated the preferential attachment dynamic network model. The remainder of this section will be devoted to prospective network surveillance strategies.

Several review papers have recently been published that discuss available methods, current challenges, and future research in the field of network surveillance. In this section, we review a handful of network surveillance methodologies that were chosen specifically to exemplify different areas of emphasis in network surveillance applications. For a more comprehensive review of this topic, see the works of Savage et al,[52] Ranshous et al,[53] Bindu and Thilagam,[54] and Woodall et al.[55]

An example of a 2-stage network surveillance framework is proposed by McCulloh and Carley.[56] These authors used a SPM methodology to monitor topological metrics such as average closeness and average betweenness with CUSUM and EWMA control charts. In their strategy, they suggest that five or more graphs should be used to establish a baseline for the dynamic network, but realistically many more graphs are required to accurately characterize the distribution of typical behavior.

Priebe et al[57] similarly monitor topological summary statistics that they refer to as "scan statistics," which describe the density of the graph. However, rather than using a fixed Phase I period to establish a baseline of typical variation, they recommend using a moving window of length 20. The advantage of a moving window approach is that the limits of typical variation can adapt to the network as it evolves over time. However, as the window moves along, observations corresponding to an undetected network change will be incorporated into the baseline, making it nearly impossible to detect a significant change in the network if the change is not identified almost immediately.

Sparks and Wilson[58] generalized the univariate EWMA strategies for Poisson counts considered in the works of Weiß,[59,60] Sparks et al,[61,62] and Zhou et al[63] to a multivariate setting for network surveillance. Similar EWMA control charts have been successfully applied to space-time monitoring of crime (see the works of Zeng et al,[64] Kim and O'Kelly,[65] Neill,[66] and Nakaya and Yano[67]). Sparks and Wilson[58] and Mei[68] are motivated by the identification of significant changes in teams of actors, where the team is possibly unknown.

Azarnoush et al[69] proposed a surveillance strategy for detecting anomalies in attributed dynamic networks, which are networks with covariate information (i.e., attributes) associated with each node. Their methodology uses logistic regression to predict the probability of the existence of an edge between 2 nodes and then applies a LRT to compare the fitted logistic regression models from one time point to the next. A significantly large value of the LRT statistic indicates a significant change in the network.

Wilson et al[48] proposed a surveillance approach that applies well-known SPM techniques to the estimated parameters of a dynamic random graph model for the observed network. The authors specifically describe surveillance of a degree corrected stochastic block model; however, the chosen random graph model is a member of a larger family of possible random graph models. We describe this technique in more detail in Section 5.3.

5.3 | Spotlight method

In this section, we highlight the approach proposed by Wilson et al[48] in which a DCSBM[49] is used to model each snapshot G_t of the dynamic network, and Shewhart and EWMA control charts[70] are used to monitor estimates of the parameters associated with the DCSBM. In Section 5.3.1, we provide an overview of control charts and SPM, and in Section 5.3.2, we provide details about the DCSBM and the surveillance methodology. Then, in Section 5.3.3, we illustrate its use on the Enron data introduced in Section 2.3.

5.3.1 | Statistical process monitoring

To perform surveillance, one must specify a statistic S_t or a vector of statistics $\boldsymbol{S}_t$ that provides information about local or global interactions in the network G_t. Once the statistic has been selected, SPM techniques can be used to identify anomalous behavior in S_t. The objective of SPM is to distinguish unusual variation from typical variation in an ordered sequence of observations of S_t. For a comprehensive review of this field, see the work of Woodall and Montgomery.[71,72] Within this framework, surveillance consists of 2 phases, i.e., Phase I and Phase II.

In Phase I, the statistic S_t is calculated for all graphs G_t constituting the dynamic network for time points $t = 1, 2, \ldots, m$. The mean μ and variance σ^2 of S_t are estimated from the m sampled statistics, and a region of typical variation $R(\hat{\mu}, \hat{\sigma}^2)$ is calculated using these estimates. The bounds of this region are referred to as the upper and lower control limits, and variation within these limits is regarded as typical. In Phase II, S_t is calculated for each new graph G_t, with $t > m$, and a new graph G_t is deemed "typical" if $S_t \in R(\hat{\mu}, \hat{\sigma}^2)$. If S_t falls outside the control limits, G_t is considered "anomalous" and we say that the control chart has *signaled*. Such a signal indicates that a significant change in interaction within the network has occurred. This information is summarized with a control chart: a time series plot of S_t constructed with upper and lower control limits depicting $R(\hat{\mu}, \hat{\sigma}^2)$, the region of typical variation. Figure 9 illustrates the distinction between Phase I and Phase II and the process by which a control chart is used to determine whether a graph G_t is anomalous, and hence, whether a change in the network has occurred.

Figure 9A depicts changing levels of interaction within the dynamic network over time with solid circles and asterisks representing nodes in 2 different communities. Figure 9B depicts the plotting of S_t over time. Open circles correspond to time points at which the interaction within the network is considered typical, and closed circles depict time points at which

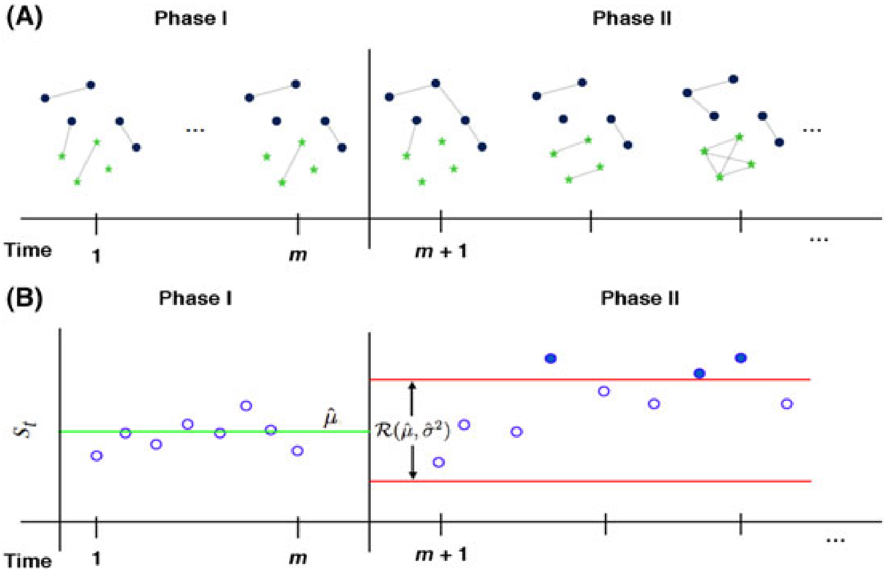


FIGURE 9 Example illustrating network surveillance using the statistic S_t and the distinction between phase I and phase II [Colour figure can be viewed at wileyonlinelibrary.com]

the level of interaction is anomalous since they lie outside $R(\hat{\mu}, \hat{\sigma}^2)$, indicated by horizontal red lines. In both Figures, the sequence of networks and, hence, the sequence of statistics S_t, are partitioned into 2 time periods corresponding to Phase I and Phase II.

The performance of the control chart in Phase II depends largely on the definition of $R(\hat{\mu}, \hat{\sigma}^2)$ and the accuracy with which it characterizes typical behavior. The definition of $R(\hat{\mu}, \hat{\sigma}^2)$ will depend on the type of control chart and the type of data being plotted, which, in turn, are guided by the types of changes one wishes to detect. We discuss these choices further in Section 5.3.2. To ensure that the control limits accurately represent typical variation in S_t, we require that the Phase I data provide precise estimates of μ and σ^2. The importance of effectively collecting and analyzing baseline data during Phase I is discussed in the works of Jones-Farmer et al[45] and Jeske.[46]

Note that the choice of statistic S_t is flexible; one may choose a topological metric that summarizes the connectivity of the nodes in the network such as density, centrality, or modularity measures, or one may choose to model the dynamic network using a random graph model and take S_t to be an estimator of one or more parameters that govern the model. These 2 approaches can be thought of as nonparametric and parametric alternatives of one another. The spotlight method employs the parametric approach as it has 2 distinct advantages. First, the parameters of a random graph model typically have useful interpretations, and a significant change in these parameters provides more information than just when the network has changed, it also provides valuable insight into where and how the network has changed. Second, the performance of a network surveillance strategy should be tested using computer simulation in which the time and nature of the change are controlled by the investigators. Random graph models provide a means to easily simulate complex and realistic networks for these types of investigations.

Thus, the spotlight network surveillance methodology is carried out in 2 stages. First, one chooses a suitable statistic S_t that adequately summarizes salient features of the dynamic network. In particular, estimates of the parameters associated with some appropriately chosen parametric random graph model are proposed. Next, one chooses a SPM strategy that is appropriate for the chosen statistic S_t and that is capable of quickly detecting the types of changes of interest.

5.3.2 | The surveillance methodology

We begin by describing the degree corrected stochastic block model, which is used to model the interactions among actors within a dynamic network. The DCSBM itself is a probability distribution on the family of undirected graphs with discrete-valued edge weights, and it is characterized by parameters that capture 2 important aspects of real-world networks: (i) community structure and (ii) degree heterogeneity.

The communities of a network refer to subgraphs (i.e., subsets of nodes) that are more densely connected to each other than to the other nodes in the network. Empirically, a network G can be partitioned into $k \geq 1$ disjoint vertex sets $n = V_1 \cup V_2 \cup \cdots \cup V_k$, where the level of interaction between actors within communities is larger than between communities. The DCSBM easily accounts for this type of structure with a $k \times k$ connectivity matrix P, where the entries $P_{r,s} > 0$ express the propensity of connection between nodes in communities r and s. Note that, for purposes of network surveillance, the community membership of each node is assumed known, but in practice, an appropriately chosen community detection algorithm must determine it. See the works of Porter et al[73] and Fortuno[74] for reviews of available methods.

While it is realistic to believe that community membership influences an actor's propensity to interact, it is also reasonable to assume that not all actors within a given community have the *same* propensity to interact. The *degree* $d(u)$ of a node $u \in [n]$ is the total number of interactions that u takes part in

$$d(u) = \sum_{v \in [n]} w(u, v).$$

The DCSBM accommodates degree heterogeneity with the degree parameter $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)$, which allows for a different interaction propensity for each of the n actors in the network. Note that the DCSBM is not identifiable without some constraint on $\boldsymbol{\theta}$, and different authors impose different constraints. Wilson et al[48] require the sum of θ_u in the same community to equal the number of nodes in that community, namely,

$$\sum_{u : c_u = r} \theta_u = n_r$$

for all $r = 1, 2, \ldots, k$, where n_r denotes the number of nodes in community r and c_u denotes the community that node u belongs to. Yu et al,[75] on the other hand, requires $\sum_{u : c_u = r} \theta_u = 1$.

The choice of constraint simply scales the value of each θ_u and affects the value of the expectation and variability of an edge weight between nodes. For instance, the value of θ_u can be viewed as a probability that an edge connects to node u given that an edge connects to community r.

Given $\boldsymbol{\theta}$, P and the community assignment of each node, the edge weights $\{w(u, v) : u, v \in [n]\}$ are modeled using a Poisson distribution

$$w(u, v) \sim \text{POI}\left(\theta_u \theta_v P_{c_u, c_v}\right).$$

For a more detailed treatment of the degree corrected stochastic block model, see the work of Karrer and Newman.[49] Maximum likelihood estimation may be used to obtain the following estimates of the DCSBM parameters

$$\hat{\theta}_u = \frac{d(u)}{n_r^{-1} \sum_{v : c_v = c_u} d(v)} \quad , \quad \hat{P}_{r,s} = \frac{m_{r,s}}{n_r \, n_s},$$

where

$$m_{r,s} = \sum_{u : c_u = r} \sum_{v : c_v = s} w(u, v)$$

is the total weight of edges between communities r and s (twice the weight of edges when $r = s$).

The surveillance strategy we spotlight here monitors these estimates in the following manner. Each of the $\binom{k}{2}$ unique entries of $\hat{P}$ is monitored via control charts. Signals on these control charts indicate changes in the level of interaction within and between communities. To monitor changes in the communities without monitoring each $\hat{\theta}_u$ individually, the following statistic is monitored to capture changes in the overall variability of interactions within community $r = 1, 2, \ldots, k$:

$$s_r = \sqrt{\frac{1}{n_r - 1} \sum_{u : c_u = r} \left(\hat{\theta}_u - 1\right)^2}.$$

Alternatively, one may monitor the vector $\boldsymbol{\theta}$ using a multivariate control chart as proposed by Yu et al.[75] Thus, a total of $\binom{k}{2} + k$ statistics (and hence, control charts) are monitored. To detect sudden large changes in each S_t, Shewhart control chart for individual observations is used with control limits calculated as

$$R\left(\hat{\mu}, \hat{\sigma}^2\right) = \hat{\mu} \pm 3\hat{\sigma}.$$

Here, μ and σ are estimated using the m Phase I graphs as described in Section 5.3.1. In particular, $\hat{\mu}$ is the sample mean of these statistics and $\hat{\sigma}$ is a moving range estimate of the standard deviation of the m statistics available from the Phase I graphs.

To detect small- to medium-sized changes that are persistent, an EWMA control chart is recommended. Instead of monitoring the statistics S_t directly, the EWMA control chart is a time series plot of Z_t, an EWMA of the S_t is

$$Z_t = \lambda S_t + (1 - \lambda) Z_{t-1},$$

where $0 < \lambda \leq 1$ is a smoothing constant and $Z_0 = \hat{\mu}$ is commonly chosen as the starting point for the moving average. The control limits associated with this control chart are calculated as

$$R\left(\hat{\mu}, \hat{\sigma}^2\right) = \hat{\mu} \pm 3\hat{\sigma} \sqrt{\frac{\lambda}{2 - \lambda}},$$

where values $0.05 \leq \lambda \leq 0.25$ have been found to work well in practice.

A signal on any of these charts indicates that the interaction patterns within the network have changed in some way. The particular statistic that signals provides information about the type and location of this change. Run length analyses have shown that this methodology is able to quickly and accurately identify local and global changes in interaction levels as well as community merges. However, the methodology in its current form does not account for inflated FAR that may occur due to the fact that many metrics are simultaneously being monitored. Extensions to this work should strive to improve upon this shortcoming. We discuss other areas of future research in Section 5.4.

5.3.3 | The Enron example

As an illustration of the spotlight methodology, we apply the DCSBM method to the Enron email corpus described in Section 2.3. We first estimated the communities of the aggregate network, which consists of the sum of each edge weight over all 143 weeks of email communication, using the walktrap community detection method[76] available in the R package igraph. This method identified 12 nontrivial communities. The DCSBM was fit to each network of weekly email counts under these communities, and Shewhart control limits were calculated for each coefficient estimate. In Figure 10, we illustrate the average rates of communication among several of these communities. The 3 dotted vertical lines indicate important transitions for Enron, including (i) the launch of Enron online, (ii) the company's filing of bankruptcy, and (iii) the hiring of Stephen Cooper as CEO, respectively. The surveillance method illustrates that there were significant changes in the email communication activity among the employees surrounding these events. Noticeably, communication tends to increase among employees after the launch of Enron online. Furthermore, for many of the Enron communities, there tends to be a spike in communication the week before bankruptcy followed by very little activity between this and the hiring of Mr. Cooper. In all situations, email communication was rare after the hiring of Stephen Cooper, perhaps signifying that the end of the company was near.

5.4 | Open research areas

The DCSBM methodology presented above represents one practically useful technique among a general family of methods for surveillance. The recommended 2-stage surveillance framework relies on (1) selecting a parametric random graph model for modeling the features of the dynamic network and (2) selecting a control chart for statistical monitoring and the identification of changes in the model's parameters. The rich literature on both random graph models and SPM provides a large number of choices for both (1) and (2) above. These choices, in turn, provide a large number of surveillance strategies that can be tailored to meet a variety of network surveillance objectives. As such, further investigation is warranted.

As new surveillance strategies become available, it is important to compare their performance with existing strategies. While the performance of many existing surveillance methodologies is demonstrated on specific examples, this alone is not sufficient.[53] Computer simulation is required to systematically evaluate method performance; only in such a controlled environment can an investigator be sure if and when a change in the network actually occurred, and whether the surveillance strategy identified the change in a timely manner. Analyses that evaluate run length properties and the detection power of a proposed methodology serve as another research opportunity.

Several other research problems that should also be considered include (i) evaluating the effect of community detection algorithms on the performance of a surveillance methodology that assumes community labels are fixed, (ii) evaluating the loss of information when an unweighted graph is used in place of a weighted one, (iii) the development of surveillance

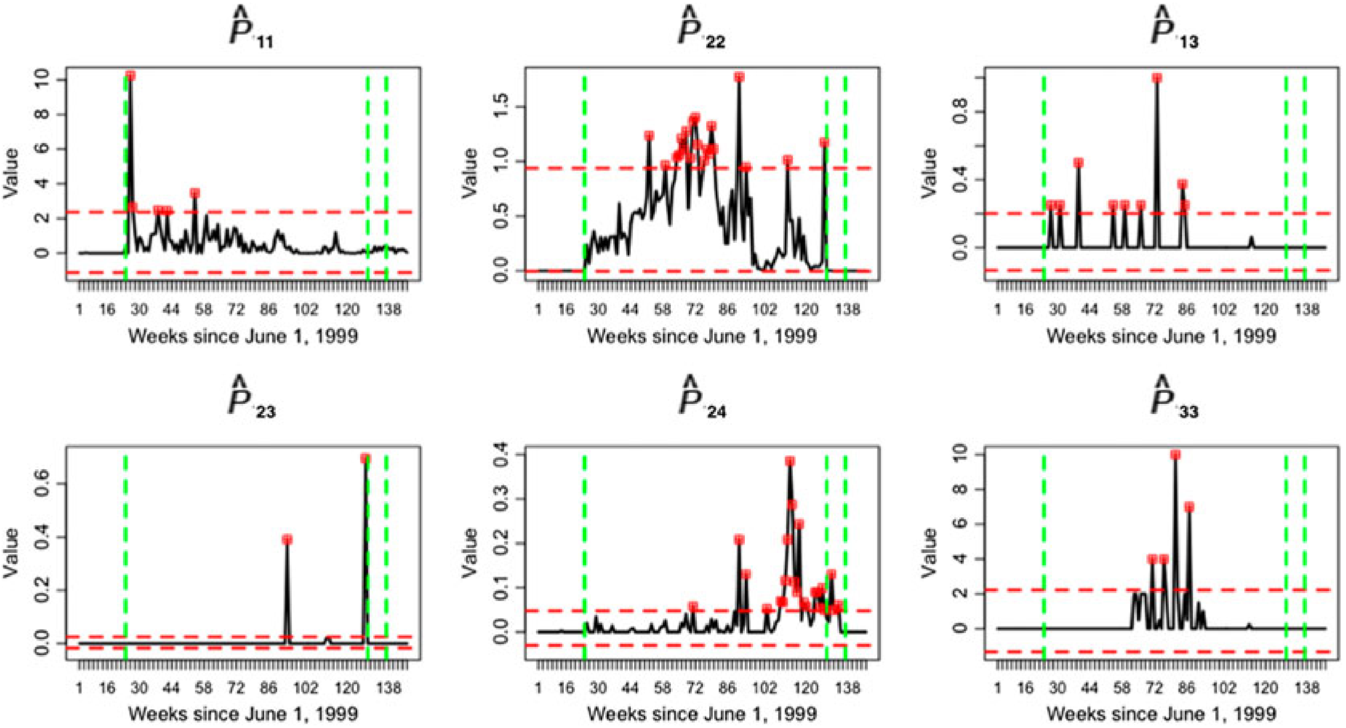


FIGURE 10 Shewhart control charts for the estimated coefficients of the degree corrected stochastic block model on the Enron email network [Colour figure can be viewed at wileyonlinelibrary.com]

strategies that account for the direction of interaction (i.e., directed as opposed to undirected graphs), and (iv) the development of surveillance strategies that account for the dependence between graphs at different time points, as opposed to assuming they are iid.

Furthermore, future work should further investigate network monitoring of attributed networks—networks with possibly dynamic node and edge covariates—as well as complex multiplex networks that change through time. Finally, the theoretical analysis of online surveillance methods for dynamic networks is lacking. This analysis will require appropriate modeling of dynamic networks with change, which the DCSBM provides an initial effort.

6 | SUMMARY

Network surveillance is a broad term, but it generally refers to monitoring a network of interconnected entities, looking for unexpected changes that precipitate a root cause investigation. We have illustrated network surveillance applications in the context of network security, network reliability, and social networks.

To a large extent, the statistical tools used for network surveillance are the same type of tools used in SPM applications. However, network surveillance contexts usually bring unique challenges that inhibit a straightforward application of the familiar SPM tools. It was seen in our examples that data network applications are challenged by how to characterize nonstationary and correlated count data, as well as unknown prechange and postchange parameters and even unknown models. Social networks have similar traffic characteristics and, furthermore, often have rapidly changing architectures. Network security applications are fraught with a wide variety of masking techniques employed by would-be perpetrators. Selecting appropriate metrics and dealing with high-dimensional and high-frequency data structures will be typical.

Development and implementation of the monitoring methods needed in network surveillance applications can be expected to be an iterative and custom process. Our hope is that our review of the field, particularly our illustrative applications, can serve as a useful starting point for practitioners who are interested in developing network surveillance algorithms.

ACKNOWLEDGEMENTS

In the past, the work of Alexander G. Tartakovsky related to network security was supported in part by the US Army Research Office under grant W911NF-14-1-0246 at the University of Southern California and in part by the US Department of Energy Phase I SBIR contract at Argo Science Corp. Currently, his work is supported in part by the Ministry of Education and Science of the Russian Federation through the Russian Academic Excellence Project 5-100 and the Arctic Project at Moscow Institute of Physics and Technology.

REFERENCES

1. Fu Y, Jeske DR. SPC methods for non-stationary correlated count data with application to network surveillance. *Appl Stoch Models Bus Ind*. 2014;30(6):708-722.
2. Moody J, Mucha PJ. Portrait of political party polarization. *Netw Sci*. 2013;1(1):119-121.
3. Taylor IW, Linding R, Warde-Farley D, et al. Dynamic modularity in protein interaction networks predicts breast cancer outcome. *Nat Biotechnol*. 2009;27:199-204.
4. Debar H, Dacier M, Wespi A. Toward a taxonomy of intrusion detection systems. *Comput Netw*. 1999;31(8):805-822.
5. Kent S. On the trial of intrusions into information systems. *IEEE Spectr*. 2000;37(12):52-56.
6. Scarfone K, Mell P. Guide to Intrusion Detection and Prevention Systems (IDPS). Gaithersburg, MD: National Institute of Standards and Technology; 2007. NIST SP 800-94.
7. Tartakovsky AG. Rapid detection of attacks in computer networks by quickest changepoint detection methods. In: Adams N, Heard N, eds. *Data Analysis for Network Cyber-Security*. London, UK: Imperial College Press; 2014:33-70.
8. Tartakovsky AG, Veeravalli VV. Asymptotically optimal quickest change detection in distributed sensor systems. *Seq Anal*. 2008;27(4):441-475.
9. Tartakovsky AG, Rozovskii BL, Shah K. A nonparametric multichart CUSUM test for rapid intrusion detection. Paper presented at: Joint Statistical Meetings; 2005; Minneapolis, MN.
10. Tartakovsky AG, Rozovskii BL, Blažek R, Kim H. Detection of intrusions in information systems by sequential change-point methods. *Stat Methodol*. 2006;3(3):252-340.
11. Mirkovic J, Dietrich S, Dittrich D, Reiher P. *Internet Denial of Service: Attack and Defense Mechanisms*. Upper Saddle River, NJ: Prentice Hall; 2005.

12. Cheng C-M, Kung HT, Tan K-S. Use of spectral analysis in defense against DoS attacks. Paper presented at: IEEE Global Communications Conference; 2002; Taipei, Taiwan.
13. Barford B, Kline J, Plonka D, Ron A. A signal analysis of network traffic anomalies. Paper presented at: Internet Measurement Workshop; 2002; Marseille, France.
14. Hussain A, Heidemann J, Papadopoulos C. A framework for classifying denial of service attacks. Paper presented at: ACM SIGCOMM Conference; 2003; Karlsruhe, Germany.
15. Lakhina A, Crovella M, Diot C. Mining anomalies using traffic feature distributions. Paper presented at: ACM SIGCOMM Conference; 2005; Philadelphia, PA.
16. Partridge C, Cousins D, Jackson AW, Krishnan R, Saxena T, Strayer WT. Using signal processing to analyze wireless data traffic. Paper presented at: ACM Workshop on Wireless Security; 2002; Atlanta, GA.
17. Huang P, Feldmann A, Willinger W. A non-intrusive, wavelet-based approach to detecting network performance problems. Paper presented at: ACM SIGCOMM Internet Measurement Workshop; 2001; San Francisco, CA.
18. Li L, Lee G. DDoS attack detection and wavelets. Paper presented at: 12th International Conference on Computer Communications and Networks; 2003; San Francisco, CA.
19. He X, Papadopoulos C, Heidemann J, Mitra U, Riaz U. Remote detection of bottleneck links using spectral and statistical methods. *Comput Netw*. 2009;53(3):279-298.
20. Bartlett G, Heidemann J. Papadopoulos C. Understanding passive and active service discovery. Paper presented at: 6th AMC SIGCOMM Conference on Internet Measurement Conference; 2007; San Diego, CA.
21. Hussain A, Heidemann J, Papadopoulos C. Identification of repeated denial of service attacks. Paper presented at: IEEE Conference on Computer Communications; 2006; Barcelona, Spain.
22. Mirkovic J, Reiher P, Papadopoulos C, et al. Testings a collaborative DDoS defense in a red team/blue team exercise. *IEEE Trans Comput*. 2008;57(8):1098-1112.
23. Mitra U, Heidemann J, Ortega A, Papadopoulos C. Detecting and identifying malware: a new signal processing goal. *IEEE Signal Process Mag*. 2006;23(5):107-111.
24. Marchette D. *Computer Intrusion Detection and Network Monitoring: A Statistical View-point*. New York, NY: Springer; 2001.
25. Tartakovsky AG. *Hybrid Intrusion Detection System Integrating Anomaly and Signature Intrusion Detection Methods*. Rolling Hills Estates, CA: Argo Science Corp; 2010. Phase I Final Technical Report.
26. Tartakovsky AG, Polunchenko AS. Decentralized quickest change detection in distributed sensor systems with applications to information assurance and counter terrorism. Paper presented at: 13th Annual Army Conference on Applied Statistics; 2007; Houston, TX.
27. Tartakovsky AG, Veeravalli V. Change-point detection in multichannel and distributed systems with applications. In: Mukhopadhyay N, Datta S, Chattopadhyay S, eds. *Applications of Sequential Methodologies*. New York, NY: Marcel Dekker; 2004.
28. Tartakovsky AG, Rozovskii BL, Blažek R, Kim H. A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods. *IEEE Trans Signal Process*. 2006;54(9):3372-3382.
29. Tartakovsky AG, Polunchenko AS, Sokolov G. Efficient computer network anomaly detection by changepoint detection methods. *IEEE J Sel Top Signal Process*. 2013;7(1):4-11.
30. Tartakovsky AG, Nikiforov I, Basseville M. *Sequential Analysis: Hypothesis Testing and Changepoint Detection*. Boca Raton, FL: CRC Press; 2014.
31. Polunchenko AS, Tartakovsky AG, Mukhopadhhyay N. Nearly optimal change-point detection with an application to cybersecurity. *Seq Anal*. 2010;31(3):409-435.
32. Antonatos S, Anagnostakis KG, Markatos EP. Polychronakis M. Performance analysis of content matching intrusion detection systems. Paper presented at: Symposium on Applications and the Internet; 2004; Tokyo, Japan.
33. Xinidis K, Charitakis I, Antonatos S, Anagnostakis KG, Markatos EP. An active splitter architecture for intrusion detection and prevention. *IEEE Trans Dependable Secure Comput*. 2006;3(1):31-44.
34. DiBenedetto S, Massey D, Papadopoulos C, Walsh P. Analyzing the Aftermath of the McColo Shutdown. Paper presented at: Workshop on Trust and Security in the Future Internet in conjunction with SAINT 2009; 2009; Seattle, WA.
35. Feather FW, Siewiorek R, Maxion D. Fault detection in ethernet networks using anomaly signature matching. Paper presented at: ACM SIGCOMM '93 Conference on Communications Architectures, Protocols and Applications; 1993; San Francisco, CA.
36. Brutlag JD. Aberrant behavior detection in time series for network monitoring. Paper presented at: 14th Systems Administration Conference; 2000; New Orleans, LA.
37. Thottan M, Ji C. Adaptive thresholding for proactive network problem detection. Paper presented at: IEEE 3rd International Workshop on Systems Management. 1998; Newport, RI.
38. Cao J, Chen A, Bu T, Buvaneswari A. Monitoring time-varying network streams using state-space models. Paper presented at: IEEE INFOCOM. 2009; Rio de Janeiro, Brazil.
39. Jeske DR, Montes de Oca V, Bischoff W, Marvasti M. CUSUM techniques for timeslot sequences with applications to network surveillance. *Comput Stat Data Anal*. 2009;53:4332-4344.
40. Montes de Oca V, Jeske DR, Zhang Q, Rendon C, Marvasti M. A CUSUM changepoint detection algorithm for non-stationary sequences with application to network surveillance. *J Softw Syst*. 2010;83:1288-1298.
41. Lambert D, Liu C. Adaptive thresholds: monitoring streams of network counts. *J Am Stat Assoc*. 2006;101(473):78-88.
42. Marvasti M, Jeske DR. Nonparametric method for determination of anomalous event states in complex systems exhibiting non-stationarity. US patent 8,275,563. September 25, 2012.

43. Gluhovsky I, Hoffman AJ, Lee HM, Yashchin E. System and method of predicting future behavior of a battery of end-to-end probes to anticipate and prevent computer network performance degradation. US patent 7,081,823. July 25, 2006.
44. Lee Y, Nelder JA, Noh M. H-likelihood: problems and solutions. *Stat Comput*. 2007;17(1):49-55.
45. Jones-Farmer LA, Woodall WH, Steiner SH, Champ CW. An overview of phase I analysis for process improvement and monitoring. *J Qual Technol*. 2014;46(3:265-280.
46. Jeske DR. Determining the phase 1 study sample size to control the accuracy of the conditional in-control ARL of a normal-theory CUSUM. *Qual Reliab Eng Int*. 2016;32:2499-2504.
47. Zhou Y, Li J, Jeske DR. A wavelet-based nonparametric CUSUM control chart for Autocorrelated processes with applications to network surveillance. Paper presented at: Joint Statistical Meetings; 2017; Baltimore, MD.
48. Wilson JD, Stevens NT, Woodall WH. Modeling and estimating change in temporal networks via a dynamic degree corrected stochastic block model. 2016. https://arxiv.org/abs/1605.04049
49. Karrer B, Newman ME. Stochastic block models and community structure in networks. *Phys Review E*. 2011;83(1). https://doi.org/10.1103/PhysRevE.83.016107
50. Peel L, Clauset A. Detecting change points in the large-scale structure of evolving networks. Paper presented at: 29th AAAI Conference on Artificial Intelligence; 2015; Austin, TX.
51. Bhamidi S, Jin J, Nobel AB. Change point detection in network models: Preferential attachment and long-range dependence. 2015. https://arxiv.org/abs/1508.02043
52. Savage D, Zhang X, Yu X, Chou P, Wang Q. Anomaly detection in online social networks. *Soc Netw*. 2014;39:62-70.
53. Ranshous S, Shen S, Koutra D, Harenberg S, Faloutsos C, Samatova NF. Anomaly detection in dynamic networks: a survey. *Wiley Interdiscip Rev Comput Stat*. 2015;7(3):223-247.
54. Bindu PV, Thilagam PS. Mining social networks for anomalies: methods and challenges. *J Netw Comput Appl*. 2016;68:213-229.
55. Woodall WH, Zhao MJ, Paynabar K, Sparks R, Wilson JD. An overview and perspective on social network monitoring. *IISE Trans*. 2017;49(3):354-365.
56. McCulloh I, Carley KM. Detecting change in longitudinal social networks. *J Soc Struct*. 2011;12:1-37.
57. Priebe CE, Conroy JM, Marchette DJ, Park Y. Scan statistics on Enron graphs. *Comput Math Organ Theory*. 2005;11(3):229-247.
58. Sparks R, Wilson, JD. Monitoring communication outbreaks among an unknown team of actors in dynamic networks. 2016. https://arxiv.org/abs/1606.09308
59. Weiß CH. Controlling processes of Poisson counts. *Qual Reliab Eng Int*. 2007;23(6):741-754.
60. Weiß CH. EWMA monitoring of correlated processes of Poisson counts. *Qual Technol Quant Manag*. 2009;6(2):137-153.
61. Sparks RS, Keighley T, Muscatello D. Improving EWMA plans for detecting unusual increases in Poisson counts. *Adv Decis Sci*. 2009;2009:1-16.
62. Sparks RS, Carter C, Graham PL, et al. A strategy for understanding the sources of variation in syndromic surveillance for bioterrorism and public health incidence. *IIE Trans*. 2010;42:613-631.
63. Zhou Q, Zou C, Wang Z, Jiang W. Likelihood-based EWMA charts for monitoring Poisson count data with time-varying sample sizes. *J Am Stat Assoc*. 2012;107(499):1049-1062.
64. Zeng D, Chang W, Chen H. A comparative study of spatio-temporal hotspot analysis techniques in security informatics. Paper presented at: 7th International IEEE Conference on Intelligent Transportation Systems; 2004; Washington, WA.
65. Kim Y, O'Kelly M. A bootstrap based space–time surveillance model with an application to crime occurrences. *J Geogr Syst*. 2008;10(2):141-165.
66. Neill DB. Expectation-based scan statistics for monitoring spatial time series data. *Int J Forecasting*. 2009;25(3):498-517.
67. Nakaya T, Yano K. Visualizing crime clusters in a space-time cube: an exploratory data-analysis approach using space-time kernel density estimation and scan statistics. *Trans GIS*. 2010;14(3):223-239.
68. Mei, Y. Quickest detection in censoring sensor networks. Paper presented at: IEEE International Symposium on Information Theory; 2011; Saint Petersburg, Russia.
69. Azarnoush B, Paynabar K, Bekki J, Runger G. Monitoring temporal homogeneity in attributed network streams. *J Qual Technol*. 2016;48:28-43.
70. Montgomery DC. *Introduction to statistical quality control*. 7th ed. New York, NY: John Wiley & Sons; 2013.
71. Woodall WH, Montgomery DC. Research issues and ideas in statistical process control. *J Qual Technol*. 1999;31(4):376-386.
72. Woodall WH, Montgomery DC. Some current directions in the theory and application of statistical process monitoring. *J Quality Technology*. 2014;46(1):78-94.
73. Porter MA, Onnela J-P, Mucha PJ. Communities in networks. *Notices Am Math Soc*. 2009;56:1082-1097.
74. Fortunato S. Community detection in graphs. *Phys Rep*. 2010;486(3):75-174.
75. Yu L, Woodall WH, Tsui KL. Detecting node propensity changes in dynamic degree corrected stochastic block models. Paper presented at: 5th International Symposium on Statistical Process Monitoring; 2017; Seoul, South Korea.
76. Pons P, Latapy M. Computing communities in large networks using random walks. Paper presented at: International Symposium on Computer and Information Sciences; 2005; Istanbul, Turkey.

How to cite this article: Jeske DR, Stevens NT, Tartakovsky AG, Wilson JD. Statistical methods for network surveillance. *Appl Stochastic Models Bus Ind*. 2018;1–21. https://doi.org/10.1002/asmb.2326