CUSUM Techniques for Timeslot Sequences With Applications to Network Surveillance

Daniel R. Jeske*
Department of Statistics
University of California, Riverside CA 92521

Abstract

We develop two cusum change-point detection algorithms for applications to data network monitoring where various and numerous performance and reliability metrics are available to aid with early identification of realized or impending failures. We confront three significant challenges with our cusum algorithms: 1) the need for nonparametric techniques so that a wide variety of metrics can be included in the monitoring process, 2) the need to handle time varying distributions for the metrics that reflect natural cycles in work load and traffic patterns, and 3) the need to be computationally efficient with the massive amounts of data that are available for processing. The only critical assumption we make when developing the algorithms is that suitably transformed observations within a defined timeslot structure are independent and identically distributed under normal operating conditions. To facilitate practical implementations of the algorithms, we present asymptotically valid thresholds. Our research was motivated by a real-world application and we use that context to guide the design of a simulation study that examines the sensitivity of the cusum algorithms.

*This is joint work with Veronica Montes De Oca (former PhD student at UCR), Wolfgang Bischoff (Faculty of Mathematics and Geography at Catholic University Eichstaett-Ingolstadt), and Mazda Marvasti (CFO of Integrien Corporation).

1. Introduction

Processes that occur over time are frequently monitored for change by metrics that are sampled on a periodic basis. Implicitly, the observer of the data is using the measurements to determine if the process is behaving as expected, or if alternatively, there has been some kind of change that would indicate the process is behaving abnormally. If a change is detected, an in-depth root cause analysis could be initiated and potentially lead to a valuable corrective action. While there is an inconvenience associated with occasional false positive alarms, it is usually far outweighed by the benefit that comes along with identification of true positive alarms.

Basseville and Nikiforov (1993), Lai (1995) and Chen and Gupta (2001) are useful references that describe in some detail a number of alternative change-point algorithms. The algorithms vary with respect to what they assume about the data when the process is operating under normal conditions, and also with respect to the type of change they are especially designed to detect. Four important performance metrics for any change-point algorithm are: 1) false positive rate, 2) false negative rate, 3) detection time and 4) computational complexity. False positive rate is the rate at which the algorithm signals change has occurred when in reality the unusual trend or pattern in the recent data is a random fluctuation within the normal operating behavior. False negative rate is the rate at which a real assignable cause for a change in the process goes undetected by the algorithm. Detection time is a measure of the average time it takes to detect a change for which there is an assignable cause. Finally, computational complexity refers to the demand for CPU cycles that is required when the algorithm is implemented. The most useful change-point detection algorithms are the ones that have low values for all of these performance metrics.

Application of change-point detection algorithms have proliferated into many fields beyond their initial use in engineering disciplines. Illustrative examples include medical applications in Belisle et al. (1998) where the effect of a certain stimulus on neuron behavior is studied and in Staudacher et al. (2005) where heart beat variability during sleep is monitored. Osanaiye and Talabi (1989) and Baron (2001) describe applications to detect disease outbreaks. Pievatolo and Rotonidi (2000) apply a change-point algorithm to distinguish between seismicity levels. Burge and Shawe-Taylor (1997) discuss an application relating to detection of cellular fraud. Chen and Gupta (1997) use change-point algorithms to detect variance changes in stock prices. Xiong and Guo (2004) describe an application in hydrology. Ye et al. (2003) and references therein, and Takeuchi and Yamanishi (2006) discuss applications related to network security.

The most familiar change-point algorithm is the classic cusum algorithm which accumulates deviations relative to a specified target of incoming measurements $\{Y_i\}_{i=1}^{\infty}$ and alarms when the cumulative sum gets too large. First proposed by Page (1954), other useful references include Van Dobben De Bruyn (1968) and Montgomery (1991). A common application is when the process is normally distributed with mean μ and known standard deviation σ. Interest centers on detecting shifts away from a targeted mean μ_0 and the shifts of interest are generally expressed as $\delta\sigma$, where δ is a specified constant. The tracking statistics used for the normal-theory cusum are

$$\begin{aligned} S_i^+ &= \max\{0, S_{i-1}^+ + Y_i - (\mu_0 + \delta\sigma)\} \quad , \quad S_0^+ = 0 \\ S_i^- &= \max\{0, S_{i-1}^- + (\mu_0 - \delta\sigma) - Y_i\} \quad , \quad S_0^- = 0. \end{aligned} \tag{1}$$

If an upward shift is important, the algorithm alarms when $S_i^+ > H^+$, with H^+ determined so that the average run length of observations between alarms when $\mu = \mu_0$ (ARL_0) is equal to a

specified (large) value. Similarly, if a downward shift is important, the algorithm alarms when $S_i^- > H^-$, with H^- similarly determined to control ARL_0. Utilizing either S_i^+ or S_i^- in this way corresponds to implementation of a one-sided cusum change-point algorithm. If shifts in either direction are important, then the two-sided cusum algorithm alarms when either $S_i^+ > H$ or $S_i^- > H$, with H determined to control ARL_0. For a given ARL_0, Monte Carlo techniques are typically used to determine suitable values of H^+, H^- or H. Various analytical approximations to the required thresholds have been proposed in Ewan and Kemp (1960), Brook and Evans (1972), Lucas and Crosier (1982), and Woodall (1983, 1984). Androulidakis et al. (2006) recently used the classic cusum algorithm for network anomaly detection, with particular emphasis on detection of denial of service attacks.

The objective of the work described in this paper is to develop a cusum algorithm for change-point detection that can be applied to data networks that exhibit structured non-stationary patterns in their traffic streams. In particular, traffic on data networks often exhibits a natural weekly cycle with weekdays being pronouncedly different from weekends. Moreover, the hours within a day vary significantly according to traditional work schedules. Figure 1 illustrates the mean and standard deviation of the number of oracle sessions on a server in a large network for each hour in a week (excluding Midnight-1AM, during which time the server was routinely rebooted). The means and standard deviations were estimated from a 12-week snapshot of session counts that were collected every five minutes. Figure 2 shows similar data for the number of live user sessions that were collected every two minutes.

Our work differs from Hajji (2005) who develops a parametric algorithm for network monitoring based an underlying data stream whose distribution is a Gaussian mixture. The

Gaussian mixture distribution is introduced to model random regime switching between traffic distributions and is not suitable for structured timeslot non-stationarity. Our work is related to Kim et al. (2007) who, along with emphasizing the need for change-point detection procedures when monitoring data networks, proposed a distribution-free cusum algorithm that applies in stationary contexts. In a later section of this paper, we consider a particular stationary context as a special case to compare the Kim algorithm to our more general cusum procedure.

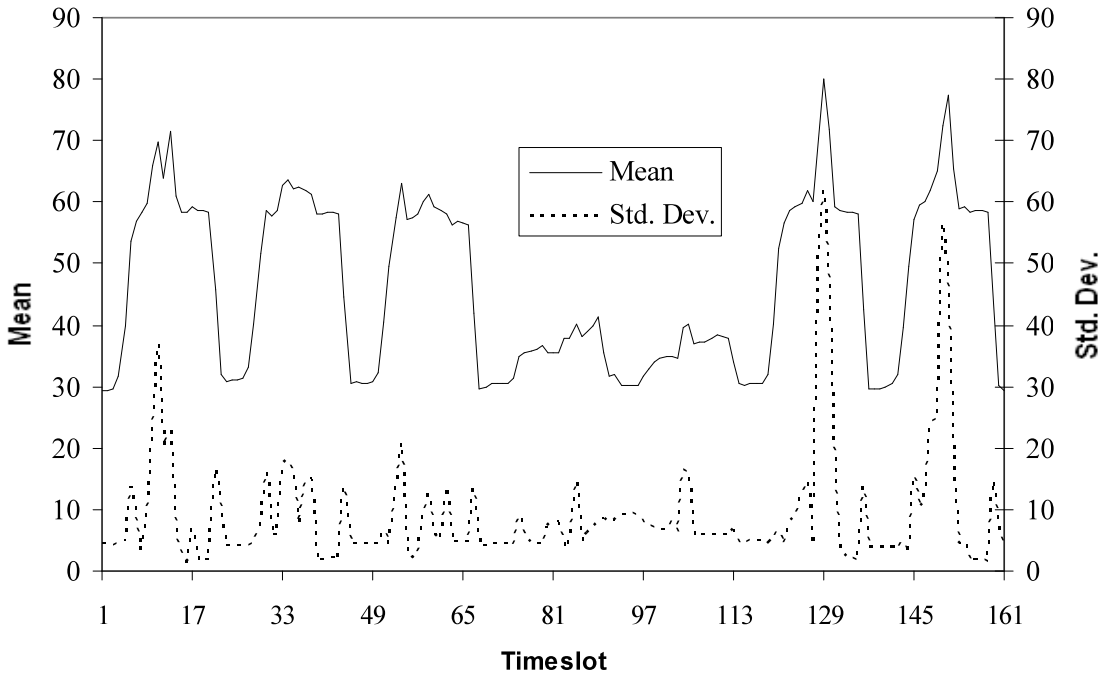

Figure 1. Variability in the Mean and Standard Deviation of the Number of Oracle Sessions on a Network Server

Because numerous reliability and performance metrics are monitored to provide information about realized or impending network failure, a significant contribution with our work is its focus on easing the computations associated with determining the cusum threshold. Two keys to helping us solve this problem are the probability integral transformation and a Brownian motion

central limit theorem. The rest of this paper is organized as follows. In Section 2 we develop two proposed cusum algorithms for our network surveillance context and provide theoretical justification that they have correct (asymptotic) false alarm rate behavior. In Section 3 we illustrate each algorithm with some data from our motivating network monitoring application, and in Section 4 we report results on a fault-injection simulation study.

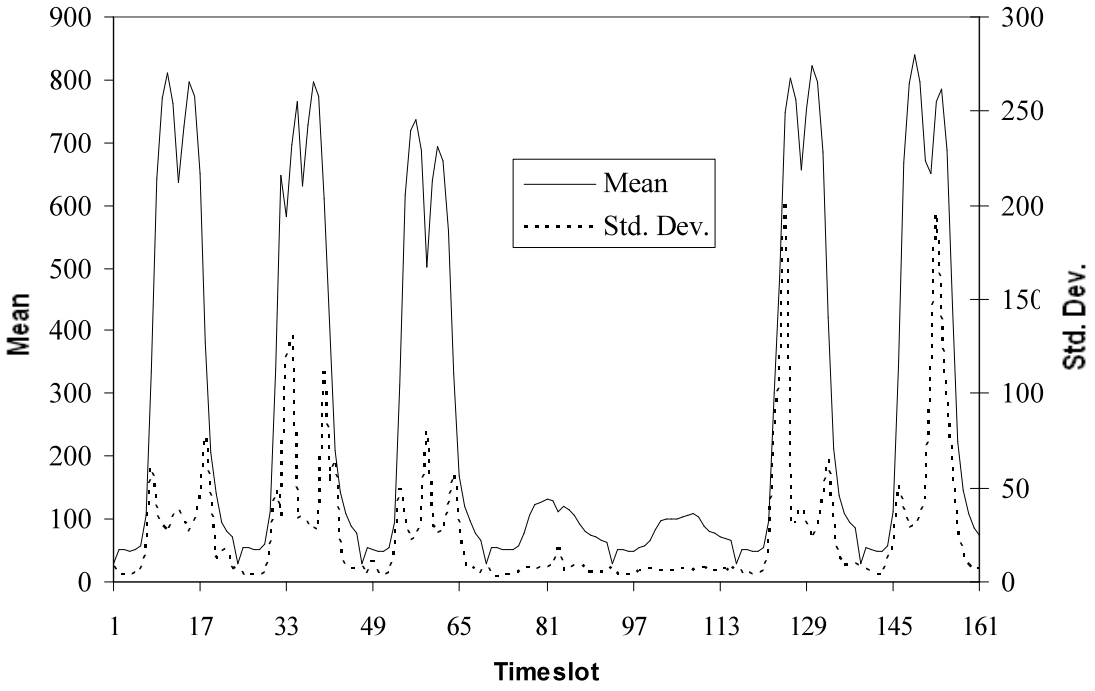

Figure 2. Variability in the Mean and Standard Deviation of the Number of Live Sessions on a Network Server

2. Proposed Cusum Algorithms

2.1 Preliminaries

Both cusums presented in this section account for non-stationarity in the data stream through the use of a defined timeslot structure and the use of historical data collected by a sliding window scheme. Our motivating example in Section 1 was illustrative of weekly cycles of 161

hourly timeslots. More generally, let *m* denote the number of timeslots per cycle. Within each timeslot, we collect and utilize *s*-cycle sliding windows of historical data $\boldsymbol{H} = \{X_{jk} : 1 \le j \le m, 1 \le k \le n_j\}$, where n_j denotes the number of historical observations available for timeslot j. We assume that under normal operating conditions observations are independent and that within a timeslot they are also identically distributed according to heterogeneous distribution functions $\{F_j\}_{j=1}^m$. In some applications, including our own as is discussed in Section 3, a transformation of the observations may be necessary to make this assumption plausible.

As Willemain and Runger (1996) did in a simpler context, we use the empirical cumulative distribution functions $\{\hat{F}_j\}_{j=1}^m$ to estimate $\{F_j\}_{j=1}^m$, and intend to infer from them what the data stream would look like during a normal operating cycle. We assume the historical data is continuously screened to remove data corresponding to intervals of failure. See, for example, Montes De Oca (2008) for one implementation of such a screening algorithm. The width *s* of the sliding window is chosen to facilitate large enough samples $\{n_j\}_{j=1}^m$ so that $\{\hat{F}_j\}_{j=1}^m$ provide adequate estimators of $\{F_j\}_{j=1}^m$. Our simulation analyses of the cusums will provide insight on suitable sample sizes $\{n_j\}_{j=1}^m$. The timeslot structure in the historical data accounts for variation in the data within a cycle, while use of the sliding window updating scheme accounts for gradual effects of network churn on the $\{F_j\}_{j=1}^m$.

In what follows, we denote the number of observations collected during a monitoring cycle for the metric under study as *N*, and let $\{Y_i\}_{i=1}^N$ denote these data. The mapping between Y_i and the timeslot to which it belongs is denoted by $\tau_i \in \{1,\ldots,m\}$. The null hypothesis that $\{Y_i\}_{i=1}^N$

comes from the set of the distributions $\{F_j\}_{j=1}^m$ is denoted by H_0 , and we use $P_0(\cdot)$ to denote a probability calculated under H_0. Whereas the normal-theory cusum in (1) measures change with respect to differences on the raw data scale, the two cusums proposed in this section measure change on a scale defined by empirical probability integral transformations of the data. Each of the proposed cusums is defined to restart at the end of each cycle using the updated $\{\hat{F}_j\}_{j=1}^m$ distributions that result from use of the *s*-cycle sliding window. In this context, rather than controlling ARL_0, it is more intuitive to control the probability of a false alarm occurring during a given monitoring cycle. That is, given that the $\{Y_i\}_{i=1}^N$ sequence comes from the same set of distributions $\{F_j\}_{j=1}^m$ as the current set of historical data, the cusum thresholds are determined so that the probability of a false alarm during the monitoring cycle is equal to a specified γ.

2.2 Transformed Cusum

The Transformed Cusum (TC) tracking statistics are a generalization of (1) defined by $T_0^- = 0$, $T_0^+ = 0$ and

$$T_i^+ = \max\left(0\,,\, T_{i-1}^+ + \hat{F}_{\tau_i}(Y_i) - \alpha\right) \quad , \quad T_i^- = \max\left(0\,,\, T_{i-1}^- + 1 - \alpha - \hat{F}_{\tau_i}(Y_i)\right) \qquad (i = 1,\ldots,N) \qquad (2)$$

where $0 < \alpha < 1$ is a suitably chosen reference value (e.g., $\alpha = 0.9$). Analytical approximations to the required thresholds for the TC algorithms are derived as follows. Considering an arbitrary observation Y_i, which maps to timeslot τ_i, let $\{X_{\tau_i(k)}\}_{k=1}^{n_{\tau_i}}$ denote the ordered values of the historical data associated with its timeslot. Define $X_{\tau_i(0)} = -\infty$ and $X_{\tau_i(n_{\tau_i}+1)} = \infty$. It follows that

$$P_0\left(\hat{F}_{\tau_i}(Y_i) = s/n_{\tau_i} \;|\boldsymbol{H}\right) = F_{\tau_i}(X_{\tau_i(s+1)}) - F_{\tau_i}(X_{\tau_i(s)}) \quad , \quad (s = 0\,,\ldots,n_{\tau_i})\,. \qquad (3)$$

The right-hand-side of (3) can be viewed as spacings in a sample of n_{τ_i} observations from a uniform distribution from which it follows that their means and variances are $1/(n_{\tau_i}+1)$ and $n_{\tau_i}/[(n_{\tau_i}+1)^2(n_{\tau_i}+2)]$, respectively. Hence, if the depth of the historical data as measured by the n_{τ_i} is sufficiently large we can approximate the distributions in (3) by

$$P_0\left(\hat{F}_{\tau_i}(Y_i) = s/n_{\tau_i} \mid \boldsymbol{H}\right) \doteq 1/(n_{\tau_i}+1) \qquad (s = 0, \ldots, n_{\tau_i}). \tag{4}$$

Consequently, conditional on $\boldsymbol{H}$, if the $\{n_j\}_{j=1}^m$ are large then the random variables $\hat{F}_{\tau_i}(Y_i)$ are approximately discrete uniform random variables under H_0 and the cusums in (2) are thus asymptotically distribution free.

The stopping rule for an upper one-sided TC algorithm is $\min\left\{1 \le i \le N : T_i^+ > t_\gamma^+\right\}$, where t_γ^+ is selected so the conditional (on $\boldsymbol{H}$) probability of a false alarm under H_0 is γ. To approximate t_γ^+, a large number *M* of sample paths $\{T_i^+\}_{i=1}^N$ can be simulated using independent discrete uniform distributions for the $\{\hat{F}_{\tau_i}(Y_i)\}_{i=1}^N$ according to (4). Denote the ordered maximum values of each sample path by $\{t_{i:M}\}_{i=1}^M$. A Monte Carlo estimate of t_γ^+ is $t_{100\gamma:M}$. The stopping rule for a lower one-sided TC algorithm is $\min\left\{1 \le i \le N : T_i^- > t_\gamma^-\right\}$, and it can be shown that $t_\gamma^- = t_\gamma^+$. The stopping rule for a two-sided TC algorithm is $\min\left\{1 \le i \le N : T_i^- > t_\gamma \text{ or } T_i^+ > t_\gamma\right\}$ and a Monte Carlo estimate of the threshold is obtained by simultaneously simulating sample paths $\{T_i^+\}_{i=1}^N$ and $\{T_i^-\}_{i=1}^N$ and defining the $\{t_{i:M}\}_{i=1}^M$ to be the ordered maximum values across both sample paths. The Monte Carlo estimate of t_γ is again $t_{100\gamma:M}$. Theorem 1 summarizes the results that have been derived in this section.

Theorem 1. a) The TC tracking statistics defined by (2) are asymptotically (as $\min n_j \to \infty$) distribution free under H_0; b) The thresholds t_γ^+ and t_γ^- for conditional (on $\boldsymbol{H}$) nominal size γ one-sided upper and one-sided lower TC algorithms, defined by $\min\left\{1 \le i \le N:\ T_i^+ > t_\gamma^+\right\}$ and $\min\left\{1 \le i \le N:\ T_i^- > t_\gamma^-\right\}$, respectively, can both be approximated by $t_{100\gamma:M}$, where $\{t_{i:M}\}_{i=1}^M$ are the ordered maximum values of a large number *M* of simulated sample paths for $\{T_i^+\}_{i=1}^N$ using the discrete uniform distributions for $\hat{F}_{\tau_i}(Y_i)$ shown in (4); c) The threshold t_γ for a conditional (on $\boldsymbol{H}$) nominal size γ two-sided TC, defined by $\min\left\{1 \le i \le N:\ T_i^- > t_\gamma \text{ or } T_i^+ > t_\gamma\right\}$, can be approximated by $t_{100\gamma:M}$, but with $\{t_{i:M}\}_{i=1}^M$ redefined to be the ordered maximums across simultaneous pairs of sample paths $\{T_i^+\}_{i=1}^N$ and $\{T_i^-\}_{i=1}^N$.

We note that since the TC algorithms have asymptotic conditional false alarm rate equal to γ, they also have unconditional false alarm rates equal to γ. In addition, we can see that the Monte Carlo algorithm for determining the threshold only depends on $\boldsymbol{H}$ through the timeslot sample sizes $\{n_j\}_{j=1}^m$. Thus, the alert thresholds for the cusum algorithms will not usually need to be updated at each cycle.

2.3 Brownian Motion Cusum

Define $U_l = \max\left[0,\ \hat{F}_{\tau_l}(Y_l) - \alpha\right]$ and $V_l = \max\left[0,\ 1 - \alpha - \hat{F}_{\tau_l}(Y_l)\right]$. The Brownian Motion Cusum (BMC) tracking statistics are defined by

$$Z_i^+ = \frac{1}{\sqrt{N}} \sum_{l=1}^{i} \frac{U_l - E_0(U_l \mid \boldsymbol{H})}{\sqrt{V_0(U_l \mid \boldsymbol{H})}} \quad , \quad Z_i^- = \frac{1}{\sqrt{N}} \sum_{l=1}^{i} \frac{V_l - E_0(V_l \mid \boldsymbol{H})}{\sqrt{V_0(V_l \mid \boldsymbol{H})}} \qquad (i = 1, \ldots, N) \qquad (5)$$

where $E_0(\cdot|\boldsymbol{H})$ and $V_0(\cdot|\boldsymbol{H})$ denote the conditional mean and variance operators, given $\boldsymbol{H}$, under H_0. Analytical approximations to the required thresholds for the BMC are derived as follows. Conditional on $\boldsymbol{H}$, the terms in each of the cusum statistics are independent and are functions of $\hat{F}_{\tau_l}(Y_l)$. Using the approximation in (4) to the conditional distribution of $\hat{F}_{\tau_l}(Y_l)$, it follows that if the $\{n_j\}_{j=1}^m$ are large and comparable in magnitude then the terms in the cusum statistics are approximately independent and identically distributed. Hence, from Billingsley (1971, section 7), it follows that conditional on $\boldsymbol{H}$ the BMC sample paths converge in distribution (as $N \to \infty$) to a standard Brownian motion sample path defined on the time interval $[0,1]$. It then follows from Feller (1966, p. 329) that

$$
\begin{aligned}
&P_0\left\{\max_{1\le i\le N} Z_i^+ < z_{\gamma/2} \mid \boldsymbol{H}\right\} \doteq 1-\gamma \\
&P_0\left\{\max_{1\le i\le N} Z_i^- < z_{\gamma/2} \mid \boldsymbol{H}\right\} \doteq 1-\gamma \\
&P_0\left\{\max_{1\le i\le N} Z_i^- < z_{\gamma/4} \text{ or } \max_{1\le i\le N} Z_i^+ < z_{\gamma/4} \mid \boldsymbol{H}\right\} \doteq 1-\gamma
\end{aligned}
$$

where $z_\gamma = \Phi^{-1}(1-\gamma)$.

Stopping rules for upper and lower one-sided BMC that have asymptotic conditional false alarm rates equal to γ during a monitoring cycle are $\min\left\{1 \le i \le N : Z_i^+ > z_{\gamma/2}\right\}$ and $\min\left\{1 \le i \le N :\ Z_i^- > z_{\gamma/2}\right\}$, respectively. Similarly, the stopping rule for a two-sided BMC that has asymptotic conditional false alarm rate equal to γ is $\min\left\{1 \le i \le N :\ Z_i^+ > z_{\gamma/4} \text{ or } Z_i^- > z_{\gamma/4}\right\}$. Because the stopping rules have conditional asymptotic false alarm rates equal to γ, their unconditional asymptotic false alarm rates are equal to γ as well.

Finally, we turn to the calculation of the conditional means and variances needed for the tracking statistics associated with the BMC. Because both U_l and V_l are simple functions of $\hat{F}_{\tau_l}(Y_l)$, the calculations are straight forward using the approximation to the conditional distributions shown in (4). In particular,

$$E_0(U_l \mid \boldsymbol{H}) = E_0(V_l \mid \boldsymbol{H})$$
$$\doteq \frac{\left(n_{\tau_l} - \lceil \alpha n_{\tau_l} \rceil + 1\right)\left(n_{\tau_l} + \lceil \alpha n_{\tau_l} \rceil - 2\alpha n_{\tau_l}\right)}{2n_{\tau_l}(n_{\tau_l}+1)}$$

$$V_0(U_l \mid \boldsymbol{H}) = V_0(V_l \mid \boldsymbol{H})$$
$$\doteq \frac{2\left(n_{\tau_l} - \lceil \alpha n_{\tau_l} \rceil\right)\left(n_{\tau_l}^2 + n_{\tau_l}\lceil \alpha n_{\tau_l} \rceil + \lceil \alpha n_{\tau_l} \rceil^2\right) + 3\left(n_{\tau_l}^2 + \lceil \alpha n_{\tau_l} \rceil^2\right) + \left(n_{\tau_l} - \lceil \alpha n_{\tau_l} \rceil\right)}{6n_{\tau_l}^2(n_{\tau_l}+1)}$$
$$+ \frac{\alpha\left(n_{\tau_l} - \lceil \alpha n_{\tau_l} \rceil + 1\right)\left(\alpha n_{\tau_l} - n_{\tau_l} - \lceil \alpha n_{\tau_l} \rceil\right)}{m(m+1)}$$
$$- \left[\frac{\left(n_{\tau_l} - \lceil \alpha n_{\tau_l} \rceil + 1\right)\left(n_{\tau_l} + \lceil \alpha n_{\tau_l} \rceil - 2\alpha n_{\tau_l}\right)}{2n_{\tau_l}(n_{\tau_l}+1)}\right]^2 \qquad (6)$$

where $\lceil x \rceil$ denotes the smallest integer larger than or equal to x. Theorem 2 summarizes the results that have been derived in this section.

Theorem 2. a) The BMC tracking statistics defined by (5) and (6) are asymptotically (as $\min n_j \to \infty$ and $N \to \infty$) distribution free under H_0; b) Asymptotic conditional (on $\boldsymbol{H}$) nominal size γ one-sided upper and one-sided lower BMC algorithm stopping rules are defined by $\min\left\{1 \le i \le N : Z_i^+ > z_{\gamma/2}\right\}$ and $\min\left\{1 \le i \le N :\ Z_i^- > z_{\gamma/2}\right\}$, respectively; c) An asymptotic conditional (on $\boldsymbol{H}$) nominal size γ two-sided BMC algorithm stopping rule is defined by $\min\left\{1 \le i \le N :\ Z_i^+ > z_{\gamma/4} \text{ or } Z_i^- > z_{\gamma/4}\right\}$.

3. Example Application

Returning to our motivating example in Section 1, we illustrate the TC and BMC algorithms for the live sessions metric. Recall that Figure 2 shows a snapshot estimate of the means and standard deviations for the $m = 161$ timeslots associated with this metric, which is collected every 2 minutes. In Sections 2.2 and 2.3 we assumed the observations within a timeslot were independent. It is more plausible for our application that the data exhibits autocorrelation, and thus we transform observations within timeslots to break up this autocorrelation. Let $r = N/m$ denote the number of observations in the sequence $\{Y_i\}_{i=1}^{N}$ that belong to the j-th timeslot and let $Y_j = (Y_{j1}, \ldots, Y_{jr})'$ denote these observations. For our application we assume the joint distribution for the components of Y_j is such that all of the marginals are identical with mean μ and $\mathrm{Cov}(Y_{js}, Y_{jt}) = \sigma_j^2 \rho_j^{t-s}$, for $t \geq s$ and $|\rho_j| < 1$. For this type of structure we transform the incoming data within each timeslot according to

$$
\begin{aligned}
Z_{j1} &= Y_{j1} \\
Z_{j2} &= \frac{Y_{j2} - \rho_j Y_{j1}}{\sqrt{1-\rho_j^2}} + \mu_j \left(1 - \frac{1-\rho_j}{\sqrt{1-\rho_j^2}} \right) \\
&\vdots \\
Z_{jr} &= \frac{Y_{jr} - \rho_j Y_{j,r-1}}{\sqrt{1-\rho_j^2}} + \mu_j \left(1 - \frac{1-\rho_j}{\sqrt{1-\rho_j^2}} \right),
\end{aligned}
\tag{7}
$$

It is easily verified that $Z_{j1}, \ldots, Z_{jr}$ are uncorrelated and identically distributed with mean μ_j and variance σ_j^2, and thus the iid assumption is at least more plausible. In this context, we note that the $\{F_j\}_{j=1}^{161}$ introduced earlier correspond to the distribution functions of the transformed

data sequence. Note that when the $\{\rho_j\}_{j=1}^m$ are all zero, the transformed data coincides with the original data. The parameters $(\mu_j, \rho_j)_{j=1}^m$ required by (7) can be estimated from the historical data. For example, re-indexing the historical data by $\boldsymbol{H} = \{X_{ijk} : 1 \le i \le s, 1 \le j \le m, 1 \le k \le n_{ij}\}$ where i denotes the week within the s-week sliding window and n_{ij} denotes the number of observations during the j-th timeslot of the i-th week, we could use estimates of the form

$$\begin{aligned} \hat{\mu}_j &= \textstyle\sum_{i=1}^{s} [\sum_{k=1}^{n_j} X_{ijk} / n_{ij}] / s \\ \hat{\rho}_j &= \textstyle\sum_{i=1}^{s} [\sum_{k=2}^{n_{ij}} (X_{ijk} - \hat{\mu}_j)(X_{ij,k-1} - \hat{\mu}_j) / \sum_{k=1}^{n_j} (X_{ijk} - \hat{\mu}_j)^2] / s. \end{aligned} \tag{8}$$

To illustrate the algorithms we used 13 weeks of live session data as follows. Weeks 1-12 were used as the initial $s = 12$ week window of historical data. The raw data [screened as described in Montes De Oca (2008)] from weeks 1-12 was used to obtain $(\hat{\mu}_j, \hat{\rho}_j)_{j=1}^{161}$ according to (8) and was then transformed according to (7) to create the empirical distribution functions $\{\hat{F}_j\}_{j=1}^{161}$. Week 13 was used as the monitoring week. The data from the monitoring week was sequentially transformed according to (7) using the estimates $(\hat{\mu}_j, \hat{\rho}_j)_{j=1}^{161}$ as defined in (8). The transformed data stream, along with the empirical distribution functions $\{\hat{F}_j\}_{j=1}^{161}$, were the inputs to the TC and BMC algorithms. Figure 3 illustrates the autocorrelation functions for the raw data and the transformed data from representative historical data where each row in the figure represents a particular timeslot during a particular week. It can be seen from Figure 3 that the transformation is successful at breaking the (exponentially decaying) autocorrelation in the raw data.

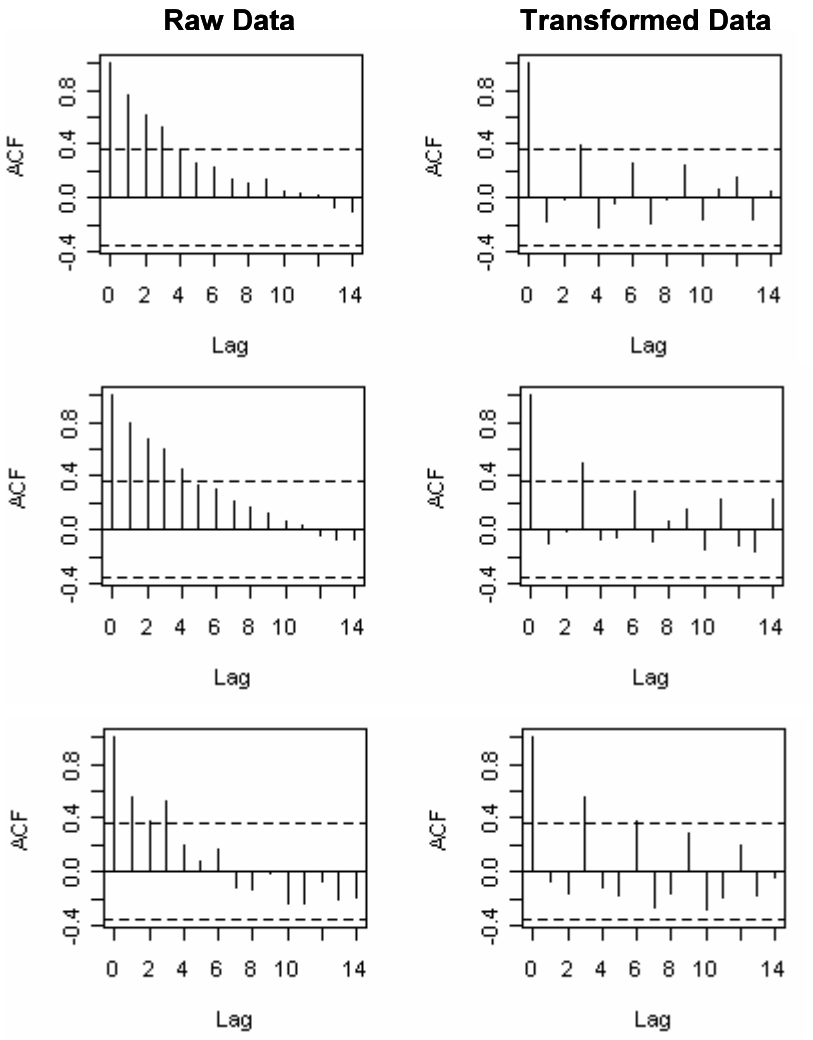

Figure 3. Autocorrelation Functions of Raw Data (Column 1) and Corresponding Transformed Data (Column 2) for Three Timeslot-Week Combinations (Rows)

Figure 4 shows the sample path of the two-sided TC algorithm $\{\max(T_i^-, T_i^+)\}_{i=1}^{4380}$, using $\alpha = 0.9$, for week 13. Also shown in the figure is the raw data trace and the nominal 10% threshold H that was estimated using the Monte Carlo method outlined in Theorem 1 with $M = 100,000$. No data were available on July 19th. Figure 4 reveals eight alarms for TC during the last half of the monitoring week (all contained within the four circled regions). Figure 5 is a similar plot showing the sample path of the two-sided BMC algorithm $\{\max(Z_i^-, Z_i^+)\}_{i=1}^{4380}$, again

with $\alpha = 0.9$. In this case, Theorem 2 implies the threshold is $z_{.025} = 1.96$. The BMC algorithm detects two anomalies during July 24th and one on July 25th.

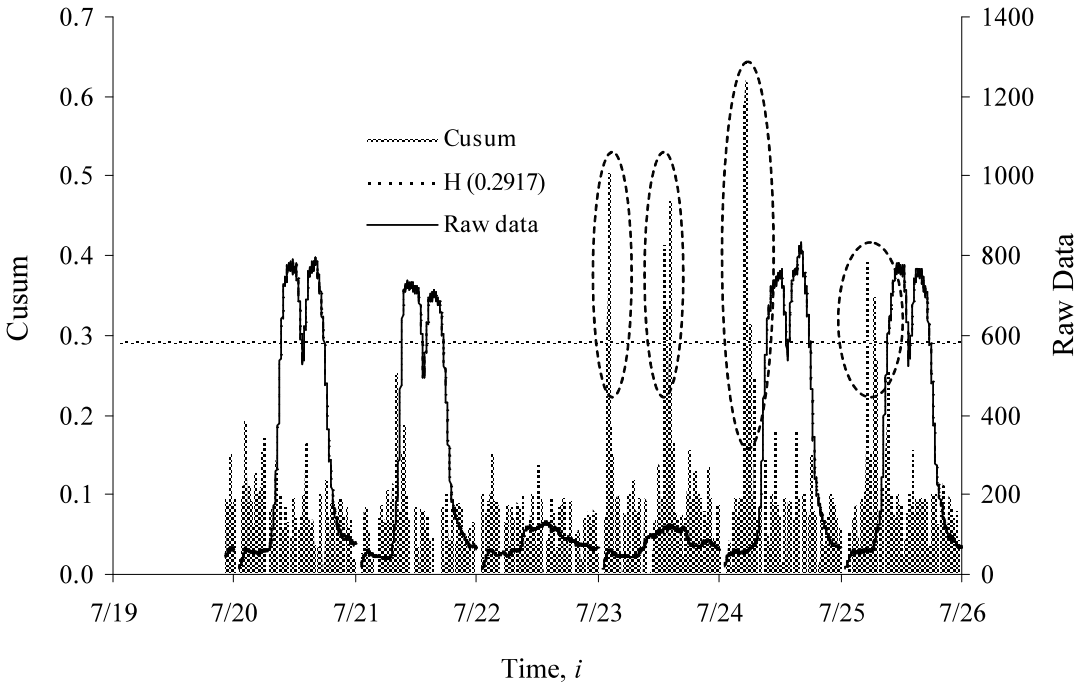

Figure 4. Two-sided TC $\{\max(T_i^-, T_i^+)\}_{i=1}^{4380}$ for Live Sessions Monitoring Week with nominal 10% Monte Carlo Threshold

Figure 6 shows zoom-in detail for three of the alarms that occurred during the monitoring week. Consider the first row of Figure 6, which corresponds to the period where TC alarms on July 25th. In the first two columns the raw data and transformed data are overlaid as traces of dark points upon traces of light gray points that depict the individual 12 weeks of historical data. Data for the first hour of each day is not available due to routine scheduled maintenance activity and this explains the break in the points at the beginning of July 25th. The last two columns in Figure 6 show the sample paths for the TC and BMC tracking statistics, and the dashed lines in these figures correspond to the respective alarm thresholds.

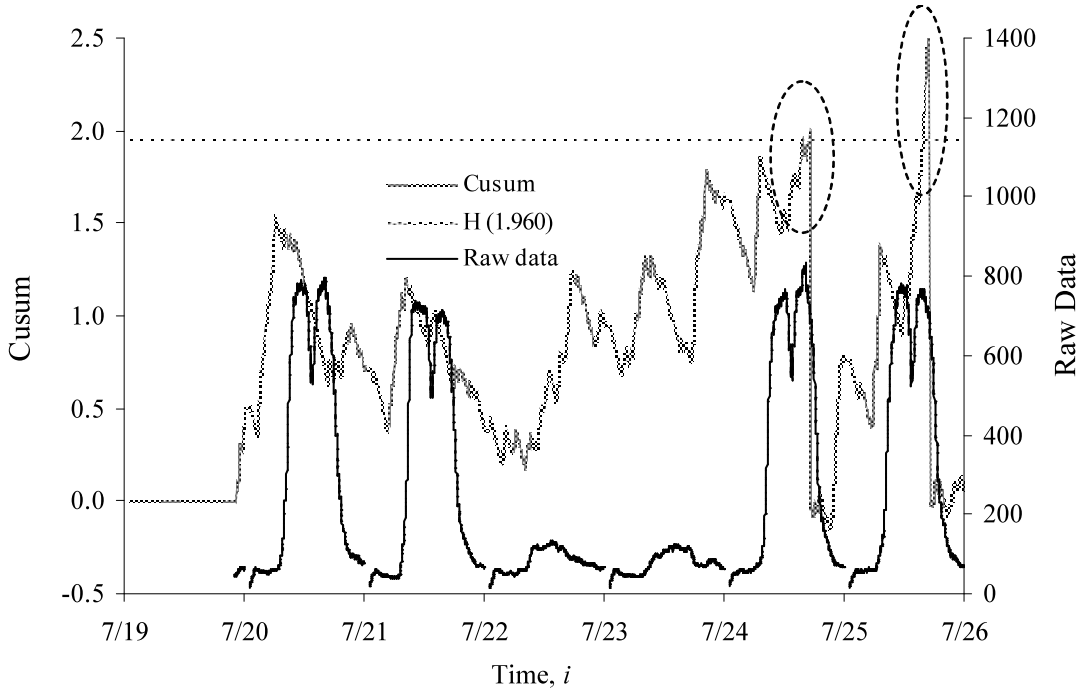

Figure 5. Two-sided BMC $\{\max(Z_v^-, Z_v^+)\}_{v=1}^{4380}$ for Live Sessions Monitoring Week with nominal 10% asymptotic threshold

The second row in Figure 6 zooms in more to further magnify details that cannot be easily seen in the first row. What can be seen in the first row is that toward the end of July 24th the raw and transformed data traces are consistently below the historical data. During this time, Z_i^- is building up a substantial positive value, though not a value large enough to alarm. In the second row, we can see a bit more clearly than in the first row that starting around 4am on July 25th, the monitoring week data is consistently above the historical data. However, by that time the diminishing value of Z_i^- is still larger than Z_i^+ because of its prior run up, and the net value of the BMC tracking statistic is Z_i^-. It would take a relatively long run of elevated observations before the net value of the BMC statistic switches to Z_i^+ and possibly cross the alarm threshold. The TC tracking statistic, on the other hand, perpetually truncates its value at zero and does not

have the same potential to build up history that is hard to overcome. Consequently, TC is able to alarm on the relatively short run of elevated observations. BMC probably would have alarmed had it not been for the fact it was relatively far below the threshold just before the start of the anomalous activity.

The third row in Figure 6 shows zoom-in detail for the period where BMC alarms on July 24th but TC does not alarm. The raw data indicate that the monitoring week data is below most of the historical data before the alarm, while the corresponding transformed data is more consistent with the historical data. Looking at the raw data, there is an unusually low observation around 4:30pm that (because of the autocorrelation) pulls the subsequent observations down with it. The transformed data trace shows this unusually low observation more or less as a stand-alone anomaly. Suspicion arises as to whether BMC alarms only because it was wandering fairly close to its threshold and normal random fluctuations caused the algorithm to alarm.

Finally, the fourth row in Figure 6 shows zoom-in detail for a period where neither the BMC algorithm nor the TC algorithm alarms. The raw data during the afternoon of July 24th is consistently below the historical data and gives a striking suggestion of anomalous activity. However, after transforming the data to account for autocorrelation, the hint of anomalous activity is removed. The autocorrelation in the raw data enabled a single observation to influence the pattern for an extended period of time and application of the cusum algorithms on the raw data stream would have resulted in an unnecessary alarm. Each algorithm perceived this fact by operating on the transformed data and consequently did not alarm.

What we see in our illustrative examples is that the BMC algorithm can have a tendency to build up history during normal operating conditions through simple wandering. When too much history is built up, the algorithm will react slowly to real anomalous conditions.

Consequently, it may take the BMC algorithm too long to find alarms. A fault-injection simulation study will be used in Section 4 to more thoroughly compare the TC and BMC algorithms with respect to anomaly detection time.

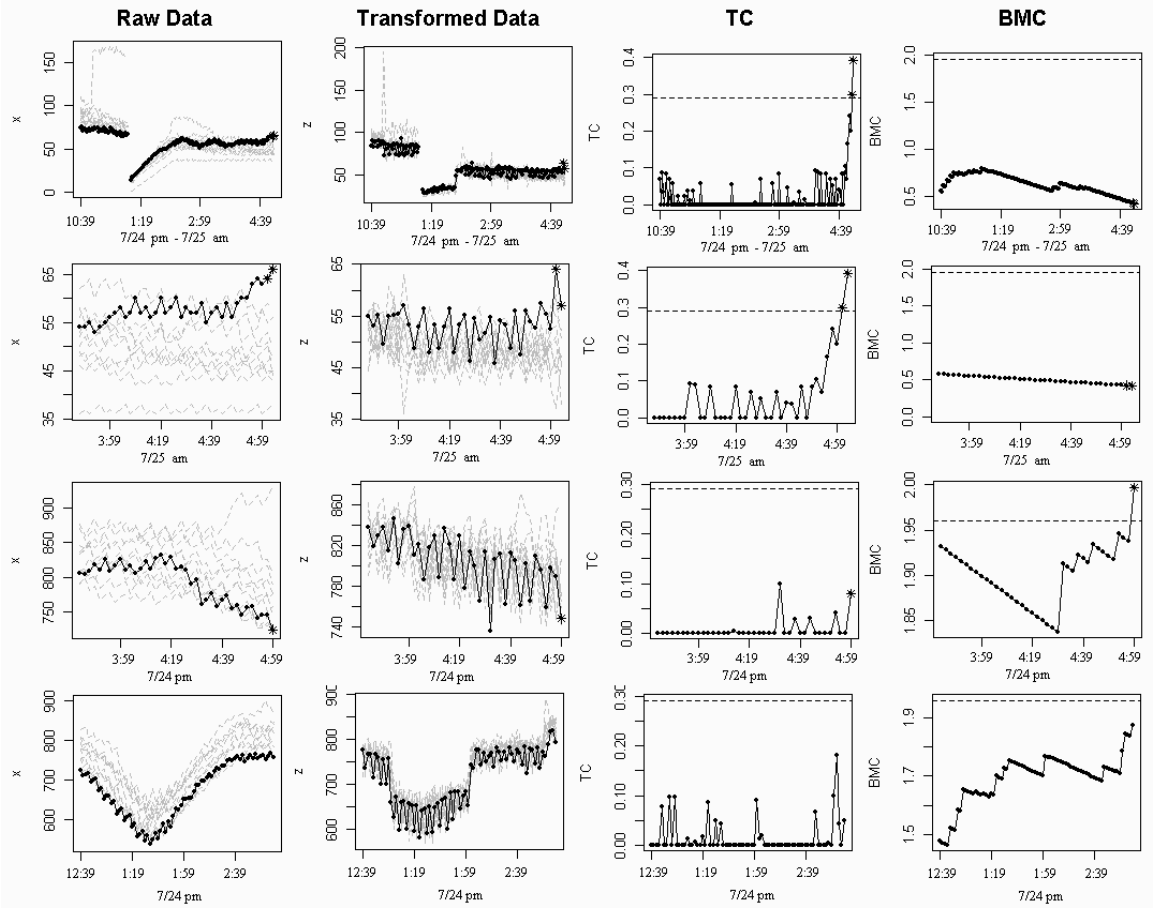

Figure 6. A Closer Look at Three Events During the Monitoring Week:
TC-only Alarms (Rows 1 and 2), BMC-only Alarms (Row 3), Neither Alarms (Row 4)

4. Performance Analysis Cusum Algorithms

In this section we report results on a simulation study to assess the performance of the cusum algorithms in terms of false alarm and true positive rates. The motivating application in Section 1 was used to guide the simulation design. In particular, we defined a cycle as one week and used $m = 161$ hourly timeslots. We selected $N \in \{1932, 4830\}$ to correspond to the oracle

sessions and live sessions metrics which are collected every five minutes and every two minutes, respectively. We assumed the depths of the historical data in each timeslot, measured by $\{n_j\}_{j=1}^{161}$, were all the same $n_j = n$, and to gain insight on the adequacy of the asymptotic theory associated with the cusums we varied $n \in \{180, 360, 720\}$.

We assume the transformations defined by (7) are implemented to correct for possible within timeslot autocorrelation. Therefore, for our simulation study we pick distribution models for the (possibly negative) transformed data sequences. We chose to use normal and Laplace distributions. Under H_0 the transformed sequences are iid within timeslots, and the timeslots have the same means and variances as the original data streams. Figures 1 and 2 were used to determine the null means and variances. In all of our simulation studies, we implemented the TC and BMC algorithms using $\alpha = 0.9$.

The results and findings from the simulation study did not depend very much on whether the normal or Laplace distribution was used, nor did they change based on whether the oracle sessions or live sessions metric was used to drive the simulation parameters. Thus, to simplify the presentation and summary of the study we only report the results obtained from using the normal distribution and the parameters associated with the live sessions metric. A complete summary of the simulation study can be found in Montes De Oca (2008).

4.1 Conditional False Alarm Rates

To assess the conditional false alarm rate of each algorithm, 25 different sets of 12-week historical data were simulated from the normal distributions using the means and standard deviations shown in Figure 2. For each of the 25 sets of historical data, 1000 monitoring weeks (week 13) were simulated from the same set of normal distributions, and sample paths for the

two-sided cusum algorithms were constructed. The proportion of sample paths that exceeded their nominal 10% alert threshold at least once during the monitoring week was recorded as the conditional false alarm rate for each of the 25 different sets of historical data. Figure 7 shows complementary empirical distribution functions (ECDF) of the 25 conditional false alarm rates for each algorithm for the three cases $n \in \{180, 360, 720\}$, and also for the case $n = \infty$. The case $n = \infty$ corresponds to using the analytical normal cumulative distribution functions rather than the empirical distribution functions $\{\hat{F}_j\}_{j=1}^{161}$ when evaluating the cusum sample paths. The four cases of *n* in Figure 7 illustrate the convergence of the conditional false alarm rate for each algorithm to the nominal value of .10. The case $n = \infty$ is not a degenerate distribution at .10 only because of the finiteness of N when constructing the sample paths. In the case $n = 180$, we see that the BMC algorithm has a preferred distribution of conditional false alarm rates since the TC algorithm shows a skew toward high values. For the other values of *n*, both algorithms have conditional false alarm rate distributions that are satisfactorily centered on .10. However, the distributions for the TC algorithm are more concentrated on .10.

4.2 Unconditional False Alarm Rates

The means of the observations that were used to draw the ECDFs of Figure 7 provide simulation estimates of the unconditional false alarm rates for each algorithm when the nominal false alarm rate is .10. Table 1 summarizes these values, and additionally reports the results of simulations run for the cases where the nominal false alarm rate was .01 and .05. We conclude from Table 1 that, except for the case $n = 180$, both algorithms achieve unconditional false alarm rates that are very close to the nominal rate.

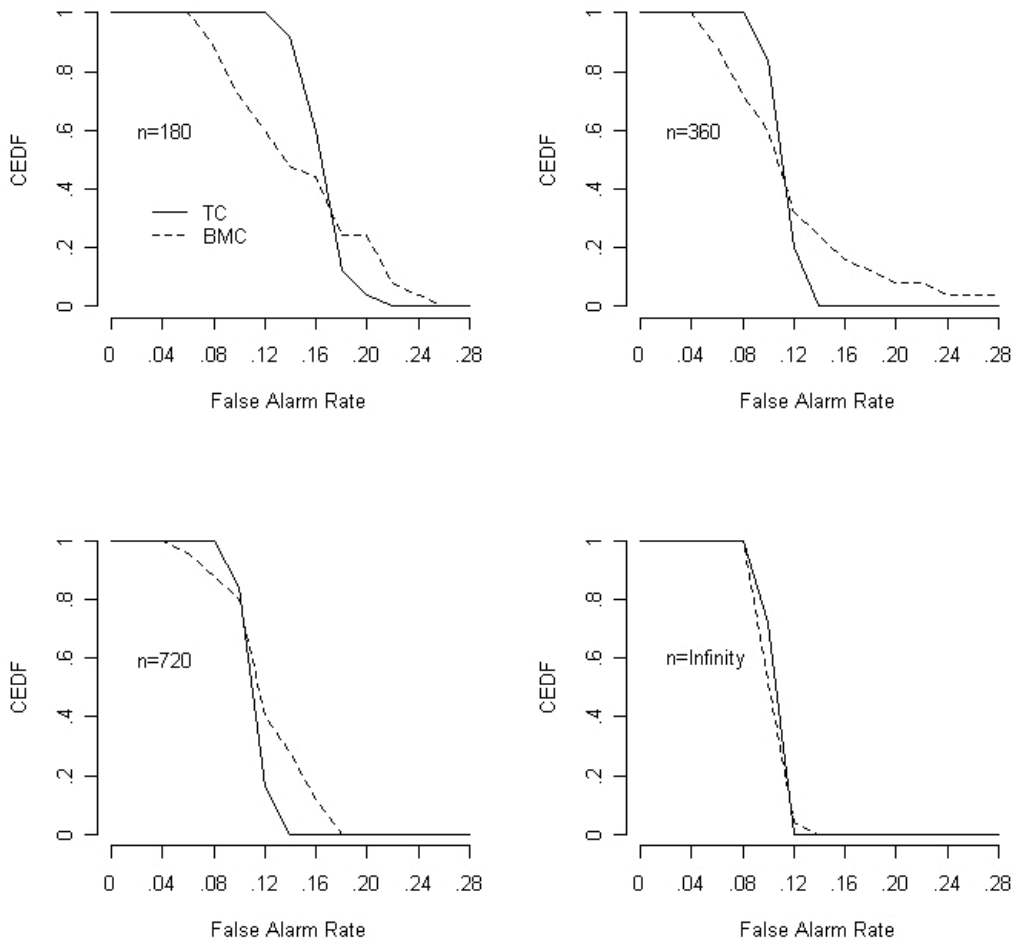

Figure 7. Simulated Conditional False Alarm Rate Distributions for Nominal 10% Two-Sided Cusum Algorithms.

n	$\gamma = 0.01$		$\gamma = 0.05$		$\gamma = 0.10$	
	TC	BMC	TC	BMC	TC	BMC
180	0.025	0.022	0.077	0.081	0.164	0.145
360	0.015	0.014	0.058	0.063	0.111	0.116
720	0.011	0.014	0.057	0.064	0.112	0.119
∞	0.010	0.011	0.053	0.053	0.104	0.102

Table 1. Simulated Unconditional False Alarm Rates for Two-Sided Cusum Algorithms

The theoretical analysis of the two cusum algorithms in Section 2 showed both algorithms have correct false alarm rate behavior in an asymptotic sense. The simulation results in Figure 7 and Table 1 suggest that for finite sample sizes the algorithms maintain satisfactory false alarm rate behavior provided the depth of the historical data is on the order of a few hundred observations per timeslot.

4.3 Fault-Injection Study

To compare the cusum algorithms with respect to how sensitive they are to departures from H_0, we modified the simulation methods by using a fault-injection technique to seed faults within the monitoring weeks. Recall from Section 4.1 that the basic simulation design consists of generating 25 sets of historical data, and then for each set of historical data 1000 monitoring weeks were simulated. In Sections 4.1 and 4.2, the monitoring week data was generated under the same set of normal distributions that the historical data was generated from. What we do differently in this section is generate monitoring week data that is seeded with faults.

In particular, each of the 1000 monitoring weeks corresponding to a particular set of historical data is seeded with 5 faults of a particular pattern. Two types of fault insertion were considered. In the first case, the fault patterns were inserted randomly into the monitoring week traffic stream, while in the second case they were inserted into relatively heavy-traffic timeslots. Faults were generated by changing the mean values of the timeslot distributions. The magnitude and duration of the change in mean values describe the fault pattern. The magnitude of increase was varied among the values 25%, 50%, 75% and 100%. The duration of the increase was varied among the values 8, 16, 30, 60 and 120 minutes. At the conclusion of a fault pattern, the cusum tracking statistics were reset to zero.

For a given fault pattern and a given set of historical data, there were 5000 opportunities for the cusum algorithm to detect a fault (5 opportunities in each of the 1000 simulated monitoring weeks). The proportion of these 5000 opportunities that were detected by the cusum algorithm was recorded, with one such proportion for each of the 25 sets of historical data. We observed little variability in these proportions across the different sets of historical data and thus only report the average (across the 25 sets of historical data) fraction of faults detected.

We note that if positive autocorrelation is present (the most likely type) the sensitivity of the cusums is somewhat muted. This is seen by examining the way in which an increase in the mean of the raw data propagates through the transformations defined by (7). For example, if the mean of the raw data in the j-th timeslot increases from μ_j to $(1+r)\mu_j$, then only the first transformed value in that time slot has a mean equal $(1+r)\mu_j$ while the remaining values have a mean equal to $[1+r\sqrt{(1-\rho_j)/(1+\rho_j)}]\mu_j$. Thus, for positive values of ρ_j the percent increase in the mean of the remaining transformed values is smaller than *r*. In our simulation study we will examine this muting effect by considering two cases involving the values of $\{\rho_j\}_{j=1}^{161}$ that are called for in (7). In the first case, we take all the ρ_j to be zero (corresponding to no autocorrelation and no muting effect) and in the second case we take all the ρ_j equal to the values we estimated in Section 3 from our 12 week snapshot of real data.

We report fault-injection results only for the case $n = 360$. Other cases of *n* were explored and showed similar results. We only show results for random fault insertions since the results for peak-period fault insertion were similar. We also used the simulation results to estimate the average detect time of the fault. Detection time could be easily measured since we knew in advance the start time of the fault and could then merely compute the difference between this

time and the time the cusum crossed its threshold. We note that detect time is partially controlled by the frequency at which the metric is measured.

Fault Pattern		Detection Percentage		Average Detect Time (min)	
Mean Increase	Duration (min)	TC	BMC	TC	BMC
25%	8	68.5	0.2	6.6	3.6
	16	81.4	0.4	7.5	9.6
	30	88.4	2.3	8.8	22.4
	60	95.7	45.9	11.4	49.7
	120	99.4	91.6	13.9	65.9
50%	8	90.7	0.3	6.2	4.0
	16	96.6	0.7	6.5	8.9
	30	99.1	3.2	6.9	23.1
	60	100.0	70.2	7.3	49.5
	120	100.0	99.9	7.3	56.2
75%	8	97.4	0.3	6.1	4.2
	16	99.8	0.5	6.2	9.9
	30	100.0	3.4	6.2	23.2
	60	100.0	78.2	6.2	49.5
	120	100.0	100.0	6.3	53.4
100%	8	99.6	0.2	6.02	3.7
	16	100.0	0.5	6.04	10.7
	30	100.0	3.8	6.04	23.3
	60	100.0	80.7	6.04	49.7
	120	100.0	100.0	6.04	52.8

Table 2. Fault Insertion Simulation Study: Nominal 10% Two-Sided Cusum Algorithms, Random Fault Injection, All $\rho_j = 0$

Fault Pattern		Detection Percentage		Average Detect Time (min)	
Mean Increase	Duration (min)	TC	BMC	TC	BMC
25%	8	5.3	0.1	7.5	2.7
	16	15.1	0.2	10.9	4.1
	30	26.2	0.3	16.1	15.0
	60	39.7	1.2	26.2	44.7
	120	57.3	14.7	44.9	91.6
50%	8	36.1	0.2	7.1	3.0
	16	56.3	0.3	8.9	8.6
	30	68.3	1.2	11.3	20.2
	60	80.0	18.9	16.4	49.6
	120	91.6	63.5	25.0	74.4

75%	8	60.1	0.2	6.6	4.1
	16	73.5	0.5	7.6	9.0
	30	81.7	2.1	9.1	22.2
	60	91.2	39.2	12.9	49.7
	120	97.9	83.9	18.0	67.9
100%	8	70.9	0.2	6.3	3.9
	16	82.1	0.4	7.1	9.0
	30	89.3	2.5	8.4	22.1
	60	96.5	49.3	11.1	49.8
	120	99.4	92.7	13.2	64.3

Table 3. Fault Insertion Simulation Study: Nominal 10% Two-Sided Cusum Algorithms, Random Fault Injection, Autocorrelation Case

Examining Table 2 we see that the TC algorithm has significantly higher detection percentages and lower detect times than the BMC algorithm. Comparing Tables 2 and 3, the effect of positive autocorrelation is seen to be substantial in terms of lowering the sensitivity of both cusum algorithms. The results in this section suggest that TC is preferable to BMC in terms of sensitivity and fault detection time, and that a couple hundred historical observations in each timeslot should be sufficient for the algorithm to achieve the correct false alarm rate. We note that in the case of no autocorrelation (Table 2), the low variability for the average detect time for the TC algorithm for fault patterns with large increases in the mean is a consequence of the increment for the TC algorithm being close to its maximum value of 0.1 for three consecutive (spaced 2 minutes apart) observations and therefore exceeding the threshold value of 0.2917. This effect is not observed in the autocorrelation case (Table 3) because, as discussed above, the magnitude of the mean increase is muted when autocorrelation is present.

4.4 Special Case of One Timeslot

For the special case of a single timeslot, the normal operating conditions emit a stationary data stream. For stationary data streams with an arbitrary (but known) autocorrelation structure, Kim et al. (2007) proposed a distribution-free cusum algorithm based on the tracking statistics

$R_i^{\pm} = \max\left(0\,,\, R_i^{\pm} \pm (Y_i - \mu_0) - K\right)$, for $i = 1, 2, \ldots$ and with $R_0^{\pm} = 0$. Here, μ_0 is the in-control process mean and K measures the magnitude of the shift (measured as multiples of the process standard deviation σ_Y so that $K = k\sigma_Y$ for some constant *k*) that is deemed important to detect. The algorithm alarms if $R_i^{+} \geq H$ or $R_i^{+} \geq H$, where H is a constant determined (e.g., by simulation) so that the average run length, ARL, equals a specified value, say ARL_0. An analytical approximation to the required *H* is given and can be used when *k* is small.

Following Kim et al. (2007), we consider the case where the data stream follows a first-order autoregressive model, AR(1), where $Y_i = \mu + \rho(Y_{i-1} - \mu) + \varepsilon_i$, where $\{e_i\}_{i=1}^{\infty}$ are independent and identically distributed normal random variables with mean zero and variance σ_e^2. For this process, we have $\sigma_Y^2 = \sigma_e^2 / (1 - \rho^2)$. For $\mu_0 = 5$ and for each value of $\rho \in \{0\,,\, 0.25\,,\, 0.5\,,\, 0.7\}$, the Kim algorithm was implemented $k = 0.1$ and taking $\sigma_e^2 = 1 - \rho^2$ to render $\sigma_Y^2 = 1$. In each case, the value of *H* (which depends on the value of ρ) was selected via a-priori simulation analyses to assure that $ARL_0 = 2000$. We note that 2000 is approximately the number of observations per week in the application described in Section 3. The values of *H* were found to be 20.1, 30.9, 47.5 and 73.5, respectively. For each value of ρ, and for shifts in the null mean of the AR(1) process belonging to the set $\{0\,,\, 0.25\,,\, 0.5\,,\, 0.75\,,\, 1.0\,,\, 1.5\,,\, 2.0\,,\, 2.5\,,\, 3.0\,,\, 4.0\}$, 5000 sample paths were generated and checked at each stage for alarms arising from the Kim algorithm. With 5000 sample paths, the coefficient of variation of the estimated ARL values are all less than 0.01. Columns 2, 4, 6 and 8 show the estimated *ARL* values of the Kim algorithm under these shifts.

Based on the earlier conclusion of this section that the TC algorithm is better than the BMC algorithm, we consider only the TC algorithm in comparison with the Kim algorithm. When the process is in-control, the transformed observations $\{Z_i\}_{i=1}^{\infty}$ are independent and identically distributed normal random variables with mean μ_0 and variance σ_Y^2 (and for our example, $\sigma_Y^2 = 1$). Hence, F is correspondingly the distribution function $F(u) = \Phi\left[(u - \mu_0)/\sigma_Y\right]$. In this comparison, F is known and the tracking statistics for the TC algorithm are thus $T_i^+ = \max\left(0, T_{i-1}^+ + F(Z_i) - \alpha\right)$ and $T_i^- = \max\left(0, T_{i-1}^- + (1-\alpha) - F(Z_i)\right)$. The threshold H required to make $ARL_0 = 2000$ for the TC algorithm is easily obtained via simulation since the $F(Z_i)$ quantities are independent and identically distributed as Uniform(0,1) random variables when the process in in-control, and it correspondingly follows that the value of H is independent of ρ.

We must now consider how to pick α in the TC algorithm so that a meaningful comparison with the Kim algorithm can be made. To this end, note that the increment in T_i^+ is positive if and only if $Z_i - \mu_0 > F^{-1}(\alpha) - \mu_0$, and comparing this condition to the Kim tracking statistic suggests α should be selected to satisfy $F^{-1}(\alpha) - \mu_0 = K$, or equivalently, $\alpha = F(\mu_0 + K)$. Since for our illustration we have $K = 0.1$, it follows that $\alpha = 0.54$. Through simulation analyses, it was verified that choosing $H = 4.95$ will provide for $ARL_0 = 2000$. Columns 1, 3, 5 and 7 in Table 4 show the ARL for the TC algorithm for the different null mean shifts.

Shift $\frac{\mu-\mu_0}{\sigma_Y}$	$\rho = 0$		$\rho = 0.2$		$\rho = 0.5$		$\rho = 0.7$	
	Kim $K = 0.1$ $H = 20.1$	TC $\alpha = 0.54$ $H = 4.95$	Kim $K = 0.1$ $H = 30.9$	TC $\alpha = 0.54$ $H = 4.95$	Kim $K = 0.1$ $H = 47.5$	TC $\alpha = 0.54$ $H = 4.95$	Kim $K = 0.1$ $H = 73.5$	TC $\alpha = 0.54$ $H = 4.95$
0	1983	2000	1985	2000	2014	2000	2002	2000
0.25	127	129	185	198	281	328	417	529
0.50	52	49	77	69	119	104	181	173
0.75	33	31	49	41	74	59	117	91
1.0	24	23	35	30	55	41	84	61
1.5	16	16	23	20	35	26	54	37
2.0	12	14	17	16	26	20	40	27
2.5	9	12	14	14	21	16	32	21
3.0	8	12	11	13	17	15	26	18
4.0	6	11	9	12	13	13	20	15

Table 4. Comparison of Kim et al. (2007) Algorithm with TC Algorithm for AR(1) Process

While assessing the comparison shown in Table 4, it should be kept in mind that the single timeslot context was selected because this is the only case where the Kim algorithm could be used, while the TC algorithm applies more generally to an arbitrary number of timeslots. Thus, we might expect that the Kim algorithm could have an advantage in this comparison and simply look for the TC algorithm to be respectably competitive. It would appear from examining Table 4 that the TC algorithm is indeed competitive. Beginning with the case $\rho = 0$ we can see the two algorithms are essentially identical in their ARL values, with perhaps a slight edge to the Kim algorithm for larger shifts. As ρ increases, the TC algorithm begins to show a slight advantage over the Kim algorithm, except when the shifts are very small. Intuitively, one might rationalize that the TC algorithm would have an increasing advantage over the Kim algorithm as ρ increases since increments for the TC algorithm are positive if and only the independent quantities $|Z_i - \mu_0|$ exceed *K*, whereas the increments in the Kim algorithm are positive if and only if the dependent quantities $|Y_i - \mu_0|$ exceed *K*.

5. Summary

We have investigated two modified cusum algorithms, TC and BMC, for handling network surveillance data streams. The proposed cusums adapt the structure of the classic cusum to account for a structured time-slot non-stationarity and introduce robustness through the use of a nonparametric approach. Each cusum is computationally tractable in terms of evaluating the necessary alarm threshold, and each was shown to have the correct asymptotic false alarm rate. While our illustrative application to a real data network shows that the implementation of either algorithm is feasible, the fault-injection simulation study shows that the TC algorithm is preferable in terms of higher true positive detection rate and lower mean time to detection. The TC algorithm also compares favorably to the Kim et al. (2007) algorithm when compared within a narrower context where the latter algorithm is also applicable.

REFERENCES

Androulidakis, G., Chatzigiannakis, V., Papavassiliou, S., Grammatikou, M., and Maglaris, V. (2006) "Understanding and Evaluating the Impact of Sampling on Anomaly Detection Techniques," *Military Communications Conference* (MILCOM 2006), pp. 1-7.

Baron, M. (2001), "Bayes and asymptotically pointwise optimal stopping rules for the detection of influenza epidemics," in Case Studies in Bayesian Statistics, (eds. C. Gatsonis, R. E. Kass, A. Carriquiry, A. Gelman, D. Higdon, D. K. Pauler, and I. Verdinelli), 6, 153-163, New York, New York: Spring-Verlag.

Basseville, M. and Nikiforov, I. V. (1993), Detection of Abrupt Changes: Theory and Application, Englewood Cliffs, NJ: Prentice-Hall.

Belisle, P., Joseph, L., Macgibbon, B., Wolfson, D. B., and Berger, R. D. (1988), "Change-point analysis of neuron spike train data," *Biometrics*, 54, 113-123.

Billingsley, P. (1971), Weak Convergence of Measures: Applications in Probability, Society for Industrial and Applied Mathematics, Philadelphia, PA.

Brook, D. and Evans, D. A. (1972), "An Approach to the Probability Distribution of Cusum Run Length, *Biometrika*, 59, 539-549.

Burge, P. and Shawe-Taylor, J. (1997), "Detecting Cellular Fraud Using Adaptive Prototypes," *Proceedings of Workshop on AI approaches to Fraud Detection and Risk Management*, 9-13.

Chen, J. and Gupta, A. (1997), "Testing and Locating Variance Change-points With Application to Stock Prices," *Journal of the American Statistical Association*, 92, 739-747.

Chen, J. and Gupta, A. (2001), "On Change-point Detection and Estimation," *Communication in Statistics – Simulation and Computation*, 30, 665-697.

Ewan, W. D. and Kemp, K. W. (1960), "Sampling Inspection of Continuous Processes with No Autocorrelation between Successive Results," *Biometrika*, 47, 363-380.

Feller, W. (1966), An Introduction to Probability Theory and Its Applications, Vol. 2, John Wiley & Sons, New York, NY.

Hajji, H. (2005), "Statistical Analysis of Network Traffic for Adaptive Faults Detection," *IEEE Transactions on Neural Networks*, 16, 1053-1063.

Kim, S., Alexopoulos, C., Tsui, K., and Wilson, J. R. (2007), "A Distribution-free Tabular CUSUM Chart for Autocorrelated Data," IIE Transactions, 39, 317-330.

Lai, T. L. (1995), "Sequential Change-point Detection in Quality Control and Dynamical Systems," *Journal of the Royal Statistical Society, Series B*, 57, 613-658.

Lucas, J. M. and Crosier, R. B. (1982), "Fast Initial Response for CUSUM Quality Control Schemes: Give Your CUSUM A Head Start," *Technometrics*, 24, 199-205.

Montgomery, D. C. (2005), Introduction to Statistical Quality Control, 5th edition, John Wiley and Sons, New York, New York.

Montes De Oca, V. (2008), Nonparametric Cusums with Applications to Network Surveillance, PhD dissertation, University of California, Riverside, CA.

Osanaiye, P. A. and Talabi, C. O. (1989), "On some non-manufacturing applications of counted data cumulative sum (CUSUM) control chart schemes," *The Statistician*, 38, 251-257.

Page, E. S. (1954), "Continuous Inspection Schemes," *Biometrika*, Vol. 41, pp. 100-115.

Pievatolo, A. and Rotondi, R. (2000), "Analyzing the interevent time distribution to identify seismicity phases: a Bayesian nonparametric approach to the multiple change-points problem," *Applied Statistics*, 49, 543-562.

Staudacher, M., Telser, S., Amann, A., Hinterhuber, H., Ritsch-Marte, M. (2005), "A new method for change-point detection developed for on-line analysis of the heart beat variability during sleep," *Physica A*, 349, 582-596.

Takeuchi, J. and Yamanishi, K. (2006), "A Unifying Framework for Detecting Outliers and Change-points from Time Series," *IEEE Transactions on Knowledge and Data Engineering*, 18, 482-492.

Van Dobben De Bruyn, C. S. (1968), Cumulative Sum Tests: Theory and Practice, Griffin's Statistical Monographs & Courses (A. Stuart, editor), New York: Hafner.

Willemain, T. R. and Runger, G. C. (1996), "Designing Control Charts Using an Empirical Reference Distribution," *Journal of Quality Technology*, 28, 31-38.

Woodall, W. H. (1983), "The Distribution of the Run Length of One-Sided CUSUM Procedures for Continuous Random Variables," *Technometrics*, 25, 295-301.

Woodall, W. H. (1984), "On the Markov Chain Approach to the Two-Sided CUSUM Procedure," *Technometrics*, 26, 41-46.

Wolfe, D. A. and Schechtman, E. (1984), "Nonparametric statistical procedures for the change-point problem," *Journal of Statistical Planning and Inference*, 9, 389-396.

Xiong, L. and Guo, S. (2004), "Trend test and change-point detection for the annual discharge series of the Yangtze River at the Yichang hydrological station," *Hydrological Sciences Journal*, 49, 99-112.

Ye, N., Vilbert, S. and Chen, Q. (2003), "Computer Intrusion Detection Through EWMA for Autocorrelated and Uncorrelated Data," *IEEE Transactions on Reliability*, 52, 75-82.