Parametric Methods for Anomaly Detection in Aggregate Traffic

Gautam Thatte, Urbashi Mitra, and John Heidemann

***Abstract*—This paper develops parametric methods to detect network anomalies using only aggregate traffic statistics in contrast to other works requiring flow separation, even when the anomaly is a small fraction of the total traffic. By adopting simple statistical models for anomalous and background traffic in the time-domain, one can estimate model parameters in real-time, thus obviating the need for a long training phase or manual parameter tuning. The detection mechanism uses a sequential probability ratio test, allowing for control over the false positive rate while examining the trade-off between detection time and the strength of an anomaly. Additionally, it uses both traffic-rate and packet-size statistics, yielding a bivariate model that eliminates most false positives. The method is analyzed using the bitrate SNR metric, which is shown to be an effective metric for anomaly detection. The performance of the bPDM is evaluated in three ways: first, synthetically generated traffic provides for a controlled comparison of detection time as a function of the anomalous level of traffic. Second, the approach is shown to be able to detect controlled artificial attacks over the USC campus network in varying real traffic mixes. Third, the proposed algorithm achieves rapid detection of real denial-of-service attacks as determined by the replay of previously captured network traces. The method developed in this paper is able to detect all attacks in these scenarios in a few seconds or less.**

I. Introduction

Security in computer networks is an extremely active and broad area of research, as networks of all sizes are targeted daily by attackers seeking to disrupt or disable network traffic. A successful denial-of-service attack degrades network performance, resulting in losses of several millions of dollars [13]. Development of methods to counter these and other threats is then of high interest. Current countermeasures under development focus on detection of anomalies and intrusions, their prevention, or a combination of both.

In this paper, we present an anomaly detection method that profiles normal traffic; a traffic-rate shift and a change in the distribution of packet-sizes from the nominal condition is flagged as an anomaly. Our anomaly detection problem is posed as a statistical hypothesis test. We develop parametric statistical models for typical and anomalous traffic. Our detection method does not need, or attempt, to model the full traffic patterns, instead it captures key, gross features of the traffic to enable informed decisions about changes in traffic. We underscore that our model does *not* capture all aspects of general Internet traffic. However, we show that, in spite of this known mismatch (confirmed in Appendix B), our model effectively captures changes in the traffic which are associated with network anomalies. Our goal is to see whether these simple, approximate statistical models can yield detection methods of high performance by modeling sufficient, salient features of the traffic.

Our approach has three key features. First, our model for anomaly detection *operates on aggregate traffic, without flow-separation or deep-packet inspection*. Both of these characteristics are essential for a practical and deployable anomaly detection system. Flow separation, per-flow anomaly detection, and deep-packet inspection are difficult or impossible for most backbone routers that have tens to hundreds of thousands of active flows per minute [7]. Since our approach only considers packet headers and timing information, it is robust to traffic concealemnt via encryption or tunneling. While it is true that the source and destination IP addresses of each packet are always available at the routers, port numbers are not available without flow-separation. Some prior work [23], [20] uses features related to the source and destination port numbers and so will not be able to detect anomalies in aggregate or VPN tunneled traffic. Note that operating on aggregate traffic is sufficient to *detect* anomalies; we assume that responses such as filtering can involve heavier weight, per-flow analysis. Instead, we focus on the detection of denial-of-service (DoS), such as TCP SYN and DNS reflector, attacks that employ fixed attack packet sizes, as well as the detection of *smart* attacks (see Section V-F) that use varying packet sizes.

Second, unlike prior anomaly detection approaches, *our method automates training* and does not require hand-tuned or hard-coded parameters. Instead, key algorithmic parameters are automatically calculated based on the underlying model parameters, or estimates thereof, which evolve as a function of network traffic. For instance, the update window size, an algorithmic parameter which is described in Section IV, is computed based on the average sample number (ASN) function. The latter is a function of the underlying model parameters, which is derived in Appendix C. Our automation significantly eases deployment and operation in networks where traffic and anomalies inevitably evolve over time.

Third, we employ both the packet-rate and sample entropy of the packet-size distribution statistics to ensure robustness against false positives, thus overcoming one of the traditional drawbacks of anomaly detection methods. Combining both these features ensures that the detection of an anomaly is delcared only when an increase in the traffic volume is accompanied by a change in the packet-size distribution. Thus, an increase in the background traffic alone, more often than not, will not be misidentified as an anomaly.

The contribution of our paper is to develop the bivariate

G. Thatte and U. Mitra are with the Ming Hseih Department of Electrical Engineering, University of Southern California, Los Angeles, CA, 90089 USA. e-mail: {thatte,ubli}@usc.edu

J. Heidemann is with the Information Sciences Institute, University of Southern California, Marina Del Rey, CA, 90292 USA. e-mail: johnh@isi.edu

Parametric Detection Mechanism (bPDM), which is completely passive, incurs no additional network overhead, and operates on aggregate traffic. Furthermore, this work suggests it is feasible to detect anomalies and attacks based on aggregate traffic at network edges, and not just near attack victims. Our detection method employs the sequential probability ratio test (SPRT) [40], a time-adaptive detection technique, for the two aggregate traffic features we consider: packet-rate and packet-size. Combining the SPRTs for these two features ensures the bPDM is robust against false positives, yet maintains rapid detection capabilities. We validate the bPDM and quantify the method's effectiveness on controlled synthetic traces, emulated Iperf attacks, and real network attacks. We introduce the bitrate SNR which is found to be an effective metric for evaluation, and superior to the previously proposed metric of packet SNR [16]. Our algorithm also performs comparably to or better than a selected set of existing detection schemes, while mitigating key drawbacks via the features described above.

This paper is organized as follows. Prior work in anomaly detection that is relevant to our research is reviewed in Section II. An overview of sequential detection methods is provided in Section III, and we then develop SPRTs of our method and the bPDM algorithm in Section IV. An evaluation of our detection mechanism using synthetic traces, real attacks, and controlled attacks in real traffic mixes is presented in Section V, as well as a numerical comparison to some existing anomaly detection methods. We conclude in Section VI. Mathematical derivations, including quantification of the model mismatch and estimator performances, are presented in the Appendices.

II. Related Work

In this section, we review the prior art in anomaly and attack detection relevant to our work. The methods described can be broadly classified as techniques requiring flow-separation, spectral or frequency-domain methods, and non-parametric change-point methods.

Methods requiring flow-separation: The techniques in [12], [14], [18], [23], [25], [27], [30], [32], [39] and [41] use certain flow-separated traffic parameters, *e.g.* source and destination IP addresses and port numbers, to detect an attack. The SPRT has been used to distinguish between reduction-of-quality flows and legitimate TCP flows in a distributed setting [8] and fast portscan detection [20].

These methods use header information and flow-separated features to detect anomalies and attacks, and in comparison to methods that classify outliers based only on traffic volume [34], [35], are more far more accurate while also yielding a lower false positive rate. On the other hand, the main disadvantages of flow-separation are its inherent complexity at the router and its inability to process encrypted traffic. Our work operates on aggregate traffic, using the traffic volume (specifically, the packet rate) to detect attacks, with the improvement that incorporation of the entropy of the packet-size, which does not require flow-separation, reduces the false positive rate and allows us to discriminate between true attacks and non-malicious changes in traffic.

Non-parametric methods: This class of methods does not assume an underlying model, but rather tailors its detection mechanism to the data. A variety of non-parametric methods employ CUSUM to implement change-point detection. The CUSUM algorithm [6] involves the calculation of a cumulative sum of the weighted observations. When this sum exceeds a certain threshold value, a change in value is declared. Prior work has focused on detecting SYN attacks using both aggregate traffic [34] and flow-separated traffic [41]. The work of [37] focuses on anomaly detection using features and statistics of the IP layer. Kalman filtering to detect anomalies using IP address filtered traffic is considered in [31]. A key drawback of the CUSUM algorithm is that the intensity of the anomaly needs to be known *a priori*; in most cases, the solution to this problem requires empirically designed thresholds that necessitate significant human effort before the scheme is initially deployed. In contrast, our detection mechanism automatically calculates key algorithmic parameters based on the underlying model.

Spectral methods: Spectral techniques have been widely used in many other fields to distinguish hidden patterns and trends from a noisy background. In the past few years, researchers have begun to apply these methods to analyze network traffic. Spectrum-based approaches have been used to detect features with near-periodic signatures, such as bottlenecks in the link layer, the effects of the TCP windowing mechanism and DoS attacks [16], and traffic anomalies [5]. They have also been employed for attack fingerprinting [17]. However, the detection accuracy of spectral methods degrades as the periodicities in the attack weaken, and most methods are more computationally expensive than corresponding time-domain techniques, especially when high speed aggregate traffic must be analyzed.

Our previous work [35] developed the parametric Modeled Attack Detector (MAD), which could rapidly detect low-rate attacks but required a dedicated training phase to learn the background traffic parameters, and which was susceptible to a few false positives. The bPDM discussed in this paper employs richer models that circumvent the need for a training phase. Combining the packet rate and packet size distribution nearly eliminates false positives. We present the bPDM in Section IV, but first provide an overview of sequential detection, which is the underlying framework of our anomaly detection method.

III. Background in Sequential Detection Methods

Hypothesis testing exploits prior knowledge of statistical descriptions of data in order to decide amongst a candidate set of populations [22]. In our problem setup, we have two hypotheses:

$$\begin{aligned} & H_0: && \text{No anomaly,} \\ \text{and} \quad & H_1: && \text{Presence of an anomaly in traffic.} \end{aligned}$$

The conditional probability density when hypothesis H_i is true is denoted $p(x|H_i)$ for $i = 0, 1$. We assume independent and identically distributed observations $\{x_k, k = 1, 2, \ldots\}$ which are drawn from one of the two probability distributions.

Given the two hypotheses and thus two decision choices, there are four possible scenarios of which we focus on two. A *false positive* (FP) or *false alarm* is declared when the

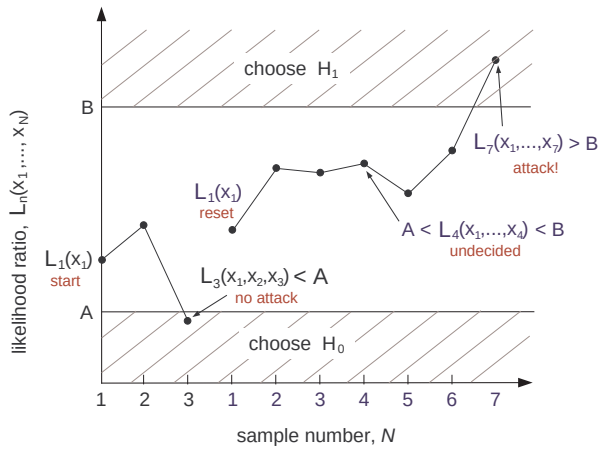

Fig. 1. Depiction of the sequential probability ratio test (SPRT).

algorithm selects H_1 when H_0 is in fact true; choosing H_0 even though H_1 is true is termed a *false negative* (FN). The probabilities of these two scenarios,

$$\alpha = P_{\text{FP}} = \Pr[H_1|H_0] \quad \text{and} \quad \beta = P_{\text{FN}} = \Pr[H_0|H_1], \quad (1)$$

are used to specify the performance criterion of the sequential detection test. The bPDM employs the sequential probability ratio test (SPRT) [40] in order to quickly detect an attack.

The *likelihood ratio* is used to implement the SPRT. Given N independent and identically distributed observations $\mathbf{x} = \{x_1, \ldots, x_N\}$, the likelihood ratio $L_N(\mathbf{x})$ is defined as

$$L_N(\mathbf{x}) = \prod_{k=1}^{N} \frac{p(x_k|H_1)}{p(x_k|H_0)} = \frac{p(x_k|H_1)}{p(x_k|H_0)} \cdot L_{N-1}(x_1, \ldots, x_{N-1}), \quad (2)$$

where the second equality illustrates that the likelihood ratio can be easily updated given a new observation.

Given a new observation, the likelihood ratio is compared to two thresholds A and B which correspond to choosing H_0 or H_1, respectively. Figure 1 depicts a realization of the SPRT wherein if $A < L_N(x_1, \ldots, x_N) < B$, the sequential test continues, and an additional observation x_{N+1} is taken as is the case with L_4 in Figure 1. But if $L_N(x_1, \ldots, x_N) \geq B$ or $L_N(x_1, \ldots, x_N) \leq A$, then the test terminates and we choose hypothesis H_1 if the former, or hypothesis H_0 if the latter, is true. In Figure 1, we see that $L_3 < A$, and thus H_0 is chosen; then, the sequential test and likelihood ratio are reset since an anomaly was not detected, and the SPRT continues. When the likelihood ratio crosses either threshold, at say sample m, the sequential test is reset by computing the updated likelihood ratio as $L(x_{m+1})$ instead of $L(x_1, \ldots, x_m, x_{m+1})$. We then see that $L_8 > B$, so H_1 is chosen indicating that an anomaly has been detected. We can either stop the test now (as shown in Figure 1), or reset the SPRT and see whether the likelihood ratio crosses the threshold B again, potentially confirming the presence of an anomaly. This latter methodology is employed in the design of our detection mechanism, detailed in Section IV.

Ideally, the boundaries A and B are selected to minimize the probability of error for all possible values of N; however this formulation of the problem is generally intractable and thus we use Wald's approximations [40] to approximate

$$B \cong (1-\beta)/\alpha \quad \text{and} \quad A \cong \beta/(1-\alpha), \quad (3)$$

which are a function of the required detection performance parameters from (1). We observe that the approximate values of A and B are independent of $p(x|H_i)$. The number of samples required for a particular test to make a decision is a random number. Thus, we examine the average value of this random number, referred to as the average sample number (ASN) function, to measure the efficacy of the SPRT. For the binary hypothesis test, the ASN function is denoted $\mathbb{E}_i(N)$ for hypothesis H_i, and is derived for our models in Appendix C.

IV. The Parametric Model

In this section, we derive the SPRTs for the packet-rate and packet-size features that are the primary components of the bivariate Parametric Detection Mechanism (bPDM). The bPDM operates on a sampled time-series of aggregate network traffic. The parametric models employed to derive the bPDM are *not* representative of *general* Internet traffic, but rather are chosen to differentiate between the presence of an anomaly and the background-only hypotheses.

A classical SPRT assumes known and constant model parameters. In reality, such parameter values are not always available, and thus we consider a *generalized likelihood ratio test* (GLRT), defined as [38]

$$G_N(\mathbf{x}) = \prod_{k=1}^{N} \frac{p(x_k, \widehat{\Theta}_1|H_1)}{p(x_k, \widehat{\Theta}_0|H_0)} \quad (4)$$

where we use the notation $p(x_k, \widehat{\Theta}_i|H_i)$ to denote replacing the true values of the model parameters Θ_i of the conditional probability density $p(x_k|H_i)$ with their maximum likelihood (ML) estimates $\widehat{\Theta}_i$. To form the *generalized SPRT*, the estimated parameters are substituted into the test form as previously described. In particular, we continue taking observations if $A < G_N(\mathbf{x}) < B$, and make a decision, choosing H_0 or H_1 if $G_N(\mathbf{x}) \geq B$ or $G_N(\mathbf{x}) \leq A$, respectively. When implementing the GLRT, the model parameters associated with either or both densities may be estimated. We adopt the notation $\widehat{\theta}_i = \widehat{\theta}|H_i$ to denote the estimate $\widehat{\theta}$ of the parameter θ when H_i is true. Herein, for both the presence of an anomaly and background only hypotheses, the respective model parameters are estimated using the observations in the SPRTs for both our features.

In particular, the model parameters are updated using non-overlapping windows. We initially use fixed-size windows for both hypotheses; a 1-second sliding window ensures that enough data is being collected to derive good estimates of the background and attack parameters, denoted $M_{init} = N_{init} = 1$ second. The offset window to estimate the H_1 parameters uses more recent samples, and thus the change in the model parameters can be detected as evidenced in Section V. Whenever the SPRT crosses the lower threshold, confirming the absence of an attack, the ASN function (see Appendix C)

is computed under hypothesis H_0, and the update window size is reset to

$$M = \min\left\{\frac{\alpha \log B + (1-\alpha)\log A}{\mathbb{E}_0(z)}, M_{init}\right\}. \quad (5)$$

Similarly, when an attack is detected by the bPDM as described in Algorithm 1, the length of the update window for the H_1 parameters is reset to

$$N = \min\left\{\frac{(1-\beta)\log B + \beta \log A}{\mathbb{E}_1(z)}, N_{init}\right\}, \quad (6)$$

where the first argument of the $\min$ functions in (5) and (6) are the ASN functions under hypotheses H_0 and H_1, respectively, and have been derived in Appendix C. We now derive the SPRTs for both the packet-rate and packet-size features, and then describe the bPDM algorithm.

A. The SPRT for the Packet-Rate

The null hypothesis H_0, which represents only background traffic, is modeled using the generalized Poisson distribution (GPD), whose probability density function (pdf) is given by

$$p(x|H_0) = \theta(\theta + \lambda x)^{x-1} e^{-\theta-\lambda x}/x!, \quad (7)$$

where $x \in \{0, 1, \ldots\}$ is the number of packet arrivals in a fixed time interval and $\{\theta, \lambda\}$ are the parameters of the GPD. We model an anomaly or attack stream as a constant rate source with *deterministic, unknown* rate r. A random variable Y drawn from the anomalous distribution is specified as

$$Y = r + X, \quad (8)$$

where X is drawn from the GPD distribution that models the background only hypothesis. For the anomaly hypothesis, we assume that the constant rate anomaly follows the pdf of the shifted GPD (sGPD)[1] given by

$$p(x|H_1) = \theta(\theta + \lambda(x-r))^{x-r-1} e^{-\theta-\lambda(x-r)}/(x-r)! \quad (9)$$

where $x \in \{r, r+1, \ldots\}$ is the number of packet arrivals in a fixed time interval and $\{\theta, \lambda, r\}$ are the parameters of the sGPD. Note that in the case where an anomaly is present, r is the minimum number of packet arrivals in a fixed time interval. For the packet-rate SPRT, under both the GPD and sGPD, x_i is thus the number of packet arrivals in the interval $\left[\frac{i}{p}, \frac{i+1}{p}\right)$, given the sampling rate p.

The SPRT, in the case of the packet-rate feature, requires us to compare the generalized likelihood ratio

$$G_N(\mathbf{x}) = \prod_{k=1}^{N} \frac{p(x_k, \hat{\theta}_1, \hat{\lambda}_1, \hat{r}|H_1)}{p(x_k, \hat{\theta}_0, \hat{\lambda}_0|H_0)} \quad (10)$$

to the threshold given in (3). Note that the densities specified in (10) are the GPD (7) and sGPD (9) with parameter estimates used in lieu of known parameter values. We now derive the estimator structures for the parameters of the GPD and sGPD for the background only and presence of an anomaly hypotheses, respectively.

The mean and variance of the GPD are given as [9]

$$\mu = \theta(1-\lambda)^{-1} \quad \text{and} \quad \sigma^2 = \theta(1-\lambda)^{-3}, \quad (11)$$

and are used to derive the moment estimators of the parameters θ and λ under the H_0 hypothesis, which are given as [9]

$$\hat{\theta}_0 = \sqrt{\frac{\overline{x}^3}{s^2}} \quad \text{and} \quad \hat{\lambda}_0 = 1 - \sqrt{\frac{\overline{x}}{s^2}}, \quad (12)$$

where $\overline{x}$ and s^2 are the sample mean and sample variance, respectively, of an M-sample window. We note that the sample mean and sample variance are computed using their unbiased estimators[2] given by

$$\overline{x} = \frac{1}{M}\sum_{i=1}^{M} x_i \quad \text{and} \quad s^2 = \frac{1}{M-1}\sum_{i=1}^{M}(x_i - \overline{x})^2, \quad (13)$$

respectively. Although both moment and ML estimators are available (see [9] for the ML estimators) in the case of the null hypothesis, we use the former since they are more computationally efficient than the latter.

For the sGPD, the moment estimators of the three model parameters (θ_1, λ_1, r) require computing third and fourth order moments which we observed to required an order of magnitude greater number of samples to compute than the average time to detection. Thus, we present an alternative estimation procedure for the model parameters under the H_1 hypothesis that is computationally lightweight.

From the construction of our anomaly model in (8), we would expect that an unbiased estimate of r can be obtained by simply using the difference in the average traffic levels in the anomalous and background only cases. Furthermore, since we are deriving moment estimators in a SPRT framework, we employ the estimator

$$\hat{r} = \max\Big\{\lfloor -\hat{\theta}_0/(1-\hat{\lambda}_0) + \overline{x}\rfloor, \min\{x_1, \ldots, x_M, x_{M+1}, \ldots, x_{M+N}\}\Big\} \quad (14)$$

where $\hat{\theta}_0$ and $\hat{\lambda}_0$ are as defined in (12). The estimate $\hat{r}$ is computed using both the N-sample sliding window and the M-sample growing window. Since $Y - r$ is a generalized Poisson distributed random variable, the other two sGPD model parameters are estimated using the GPD estimator structures in (12) and are given as

$$\hat{\theta}_1 = \sqrt{\frac{(\overline{x} - \hat{r})^3}{s^2}} \quad \text{and} \quad \hat{\lambda}_1 = 1 - \sqrt{\frac{\overline{x} - \hat{r}}{s^2}} \quad (15)$$

where $\hat{r}$ is the estimate given in (14), and $\overline{x}$ and s^2 are computed via (13) using the N-sample sliding window. Note that the $\min\{\cdot\}$ function in (14) is a constraint due to the fact that the support of the sGPD is over $\{r, r+1, r+2, \ldots\}$.

Employing the GPD/sGPD hypothesis test lets us detect a change in the mean of the traffic, but an increase in the mean does not always occur due to a malicious anomaly or

[1] In our previous work [35], we modeled the presence of an anomaly using the simpler shifted Poisson distribution. The richer, generalized Poisson model is employed herein to circumvent the need for a dedicated training phase, and allow all the model parameters to be estimated online.

[2] An estimator is defined as *unbiased* if the estimator's expected value is equal to the true value of the parameter being estimated, *i.e.* $\mathbb{E}\{\hat{\theta}\} = \theta$ [22].

attack. Flash crowds, which might occur due to the Digg or SlashDot effect, are not malicious traffic [19]. Since our detection methods are designed to operate on aggregate traffic, to allow for analysis of encrypted traffic, we cannot employ the source and destination IP addresses or port numbers to develop a more robust detection scheme. Thus, packet-size information is now incorporated to reduce the number of false positives.

B. Incorporating the Packet-Size SPRT

The packet-size distribution of nominal Internet traffic has been characterized in [29] as mostly bimodal at 40 bytes and 1500 bytes (with 40% and 20% of packets, respectively). An examination of our background trace data, which include Ethernet and VLAN headers, validates the bimodal distribution but with differing means (68 bytes and 1518 bytes) and relative numbers of each packet type.

We expect packet-size distribution information to be effective in attack detection, since a broad class of attacks use a single packet-size; *e.g.* DNS reflector attacks use the maximum packet-size and TCP SYN attacks use the minimum packet size. Thus, the influx of attack packets, in the case of these types of attacks, will alter the relative number of a specific packet-size with respect to the packet-size distribution of nominal traffic. As such, the sample entropy of the packet-size distribution can be used to distinguish between the no attack and attack hypotheses.

In the bPDM framework, recall that x_i represents the number of packet arrivals in the interval $\left[\frac{i}{p}, \frac{i+1}{p}\right)$. Let $\mathbb{S}_i$ denote the set of distinct packet sizes that arrive in this interval, and q_j denote the proportion of packets of size j to the total number of packets in the same interval. Thus, the sample entropy y_i is computed as

$$y_i = -\sum_{j \in \mathbb{S}_i} q_j \log q_j \ . \tag{16}$$

The sample entropy is modeled using the Gaussian distribution given by

$$p(y|H_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{1}{2\sigma_i^2}(y-\mu_i)^2\right] \tag{17}$$

for both the background ($i = 0$) and attack ($i = 1$) hypotheses. Thus, the log-likelihood ratio (LLR), given N observations, is specified as

$$\log L(\mathbf{y}) = a_2 \sum_{i=1}^{N} y_i^2 + a_1 \sum_{i=1}^{N} y_i + a_0, \tag{18}$$

where $a_2 = \frac{1}{2\sigma_0^2} - \frac{1}{2\sigma_1^2}$, $a_1 = \frac{\mu_1}{\sigma_1^2} - \frac{\mu_0}{\sigma_0^2}$, and $a_0 = N\left[\frac{\mu_0^2}{2\sigma_0^2} - \frac{\mu_1^2}{2\sigma_1^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right)\right]$. As in the case of the GPD/sGPD hypothesis test, the model parameters in the case of the sample entropy are estimated in real time using the sliding and growing update windows. Since the sample entropy is modeled using the Gaussian distribution, the parameter estimators for μ and σ^2 for each of the hypotheses are the sample mean $\overline{x}$ and sample variance s^2, given in (13), using the respective update windows. The resulting SPRT requires that we continue to take more observations if

$$\log(A) < \log G(\mathbf{y}) < \log(B) \ , \tag{19}$$

where $G(\mathbf{y})$ is the generalized likelihood ratio associated with the packet-size SPRT. $\log G(\mathbf{y})$ is of the form in (18), but the constants a_2,a_1 and a_0 are defined in terms of the parameter estimates $\{\hat{\mu}_0, \hat{\sigma}_0^2\}$ and $\{\hat{\mu}_1, \hat{\sigma}_1^2\}$ instead of the true parameter values.

Given two features, ideally we would compute a joint density to determine a single bivariate SPRT. However, given the mixed nature of the two features (discrete packet arrivals and continuous entropies) computing this joint density appears to be intractable. Instead, we now describe our bPDM algorithm, which effectively combines the two SPRTs to yield an anomaly detection mechanism that has a low false positive rate.

C. The bPDM Algorithm

The bPDM combines the SPRTs of the packet-rate and packet-size features. Recall that the bPDM must be initially deployed in the absence of an anomaly. Once the initial parameter estimates have been computed, subsequent observations are used to update the parameter estimates for both hypotheses and compute the likelihood ratios. For each of the SPRTs, the likelihood ratio is updated given each new observation as described in (2). The H_0 and H_1 parameters are estimated using a fixed number of samples.

During the operation of the bPDM, if only one of the SPRTs (packet-rate or packet-size) crosses the upper threshold B, then we declare an initial warning and continue computing the likelihood ratio after resetting the corresponding SPRT. For example, an increase in the packet-rate without a significant change in the sample entropy of the packet-size distribution may be due to a normal non-malicious increase in traffic. Thus, an attack is declared only if an initial warning is followed by the other SPRT crossing the upper threshold, *i.e.* we declare an attack only if *both* the packet-rate and packet-size SPRTs "coincidentally" cross the upper threshold. Requiring the SPRTs to cross the upper threshold at the same sample is too restrictive, being the equivalent of millisecond accuracy; thus, we define the "hold time," $\tau_H = 0.1$ second, and require that the SPRTs cross the upper threshold within τ_H samples of each other. Consequently, a false positive is said to have occurred when both SPRTs coincidentally cross the upper threshold *and* there is no anomaly present in the traffic.

Algorithm 1 provides an overview of the bPDM algorithm, which operates on each sample of the time-series generated from the network traces. The subscripts $_{pr}$ and $_{ps}$ refer to the packet-rate and entropy of the packet-sizes, respectively. When the bPDM is initially deployed, the update window sizes are set to $M = \min\{\mathbb{E}_0(N), 1000 \text{ samples}\}$ and $N = 1000$ samples in Line 6. Specifically, we use the ASN function under H_1, denoted $\mathbb{E}_0(N)$, to shorten the update window in order to achieve quicker detection. As the network traffic evolves, the ASN function is *automatically* recomputed, as detailed in Appendix C, and thus the optimal window size is

Algorithm 1 Outline of the bivariate Parametric Detection Mechanism (bPDM) algorithm

1 Load time-series for packet-rate and sample entropy of packet-size distribution
2 Set $P_{FP} = 10^{-8}, P_{FN} = 10^{-7}, \tau_H = 100$
3 Compute SPRT thresholds A and B using (3)
4 Estimate parameters $\{\theta_0, \lambda_0, \mu_0, \sigma_0^2\}$ using (12), (13)
5 Compute ASN function $\mathbb{E}_0(N)$ using (26)
6 Set H_0 and H_1 update window sizes, $M = \min\{\mathbb{E}_0(N), M_{init}\}$ and $N = N_{init}$, respectively
7 Initialize $LLR_{pr/ps} = 0, flag_{pr/ps} = 0$
8 **for** $i = M + N + 1, \ldots$ **do**
9 **if** $flag_{pr/ps} = 0$ **then**
10 Compute *pr/ps* H_0 parameters $\{\theta_0, \lambda_0\}$ and $\{\mu_0, \sigma_0^2\}$ using (12) and (13), respectively
11 **end if**
12 Compute H_1 parameters $\{\theta_1, \lambda_1, r\}$ and $\{\mu_1, \sigma_1^2\}$ using (14), (15), (13)
13 Update $LLR_{pr}[i]$ using (2) with densities (7),(9)
14 Update $LLR_{ps}[i]$ using (18)
15 **if** $LLR_{pr}[i] < \log(A)$ **then**
16 Reset $LLR_{pr}[i+1] = 0$
17 Reset $flag_{pr} = 0$ to update *pr* H_0 parameters
18 **end if**
19 **if** $LLR_{ps}[i] < \log(A)$ **then**
20 Reset $LLR_{ps}[i+1] = 0$
21 Reset $flag_{ps} = 0$ to update *ps* H_0 parameters
22 **end if**
23 **if** $LLR_{ps/pr}[i] > \log(B)$ **then**
24 Set $flag_{ps/pr} = 1$ to stop H_0 parameter update
25 Declare initial warning!
26 **if** $LLR_{pr/ps}[i + \tau_H] > \log(B)$ **then**
27 **if** H_1 is true **then**
28 Anomaly detected! Recompute ASN function $\mathbb{E}_1(N)$ with $\hat{r}$ using (27)
29 Reset H_1 window, $N = \min\{\mathbb{E}_1(N), N_{init}\}$
30 $LLR_{pr/ps}[i+1] = 0$
31 **else**
32 False positive!
33 $LLR_{pr/ps}[i+1] = 0$
34 **end if**
35 **end if**
36 $LLR_{ps/pr}[i+1] = 0$
37 **end if**
38 **end for**

re-determined whenever the SPRT crosses the lower threshold. Lines 10–15 describe updating the background parameters only if neither SPRT has crossed the upper threshold. If the possibility of an attack exists, the background parameters are not updated to avoid using attack samples in the estimates of the H_0 parameters. The attack parameters are continuously updated as in Line 16. Lines 19–26 describe the reseting the flags that control updating the H_0 parameters when the absence of an attack is confirmed by either of the packet-rate or packet-size SPRTs. Lines 27–42 detail the anomaly

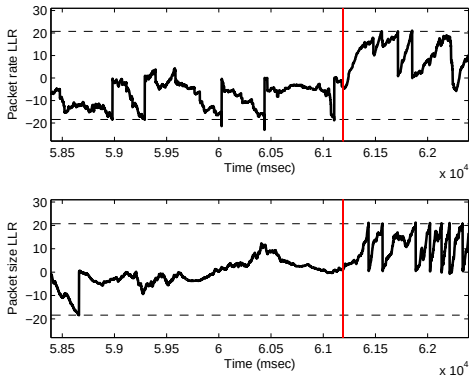

Fig. 2. SPRTs for the packet rate and packet size features for the Iperf attack with bitrate SNR 0.056: an attack is declared when *both* SPRTs cross the upper threshold B, 695 msec after start of attack.

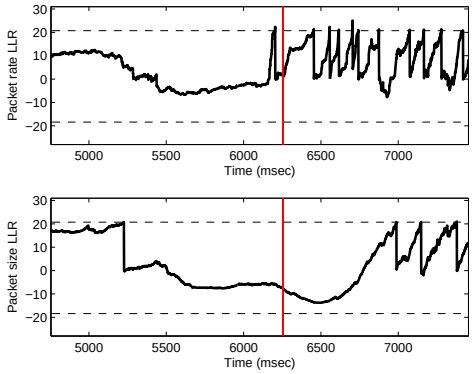

Fig. 3. SPRTs for the packet rate and packet size features for a synthetic TCP SYN attack with bitrate SNR 0.005. Non-coincidental crossings are simply flagged as warnings.

detection mechanism described earlier, which includes not updating H_0 parameters when a warning is raised to ensure that the background parameter estimates are not computed using attack samples.

D. An Illustrative Example

In order to highlight the facets of the bPDM, we consider the detection of a simulated Iperf attack with bitrate SNR 0.056 using our detection mechanism. Figure 2 shows the SPRT outputs that result from detecting this Iperf attack. We declare the presence of an attack 695 msec after the start of the attack, although the packet-size SPRT crosses the upper threshold before this point in time. This delay is due to the fact that the bPDM requires that both SPRTs coincidentally cross the upper threshold. Although it is the case that the time to detection could have been reduced had we used only the packet-rate SPRT in the case of Figure 2, it would have yielded a false positive in the case of Figure 3. Therein, we find that the crossing of the upper threshold before the start of the attack is flagged as a warning, but an attack is not declared. Thus, the bPDM reduces false positives by leveraging both the packet-rate and packet-size features of aggregate traffic.

V. Performance Evaluation and Analysis

In the following sections, we employ synthetic traces, real network attacks, and emulated Iperf attacks in varying traffic mixes to investigate the effects of background and attack traffic levels on the time to detection. We show that the performance of the bPDM is comparable to or better than selected alternate detection schemes (Section V-D), and that time to detection is influenced by bitrate SNR. We define bitrate SNR in Section V-A, show how how it is affected by attack and background traffic rates and hopcounts (Sections V-B, V-C, and V-I). We also compare it to previously used packet SNR (Section V-G), Finally, we validate our synthetic attacks (Section V-H) and that bPDM works with minimal training (Section V-E), and is robust to countermeasures (Section V-F).

A. Evaluation of the bPDM

The basic principles of detection theory teach that the time to detection of a signal in noise is related to the signal-to-noise ratio (SNR) [38]. However, for anomaly detection, there is no clear notion of what an appropriate SNR measure would be. We present the *bitrate SNR* metric, which is defined as

$$\text{bitrate SNR} = \frac{\text{Anomalous traffic level}}{\text{Background traffic level}} = \frac{\sum_{S\in\mathcal{S}_A} M_S S}{\sum_{S\in\mathcal{S}_B} M_S S}, \tag{20}$$

where $\mathcal{S}_A$ is the set of attack packet-sizes, $\mathcal{S}_B$ is the set of background packet-sizes, and M_S is the number of packets of size S in bits.

In this section, we evaluate the bPDM using a set of synthetic traces and emulated Iperf attacks, and find that as the bitrate SNR increases, the time to detection decreases. This trend is also shown to be true for the underlying theoretical model of the bPDM.

1) Evaluation using simulated synthetic traces: The bPDM is first evaluated using a set of synthetic attacks [4] that allow us to control the attack rate and methodically evaluate the bPDM. The attack traces use 196 megabits per second (Mbps) background traffic taken from our network. After 6–8 seconds of background traffic, we add in constant rate attacks at various rates using Stream Merger [21]. Focusing on low-rate attacks, our traces employ attacks that range from 1 Mbps to 120 Mbps, in addition to the 196 Mbps background traffic. The artificial attacks model TCP SYN attacks that use a fixed attack packet size of 68 bytes [7].

Figure 4 plots the bPDM times to detection for the set of synthetic TCP SYN attacks as a function of the bitrate SNR, in addition to the detection times for emulated attacks and the theoretical model discussed in following sections.

The bPDM was run on 8 distinct synthetic traces of a specific bitrate SNR, and the mean values of the detection times are plotted in Figure 4 along with error bars that represent the standard deviation associated with the mean detection time. We see that as the bitrate SNR increases, the bPDM time to detection of the synthetic attacks decreases.

2) Evaluation using emulated Iperf traces: We next consider a more realistic scenario wherein controlled attacks in *varying* traffic mixes are detected by our algorithm. Specifically, we employ 80-second Iperf attacks that use 345-byte fixed-size packets sent from Colorado State University (CSU) to the University of Southern California (USC); their generation is detailed in Appendix A. As before as the bitrate SNR increases, the time to detection of these emulated Iperf attacks decreases.

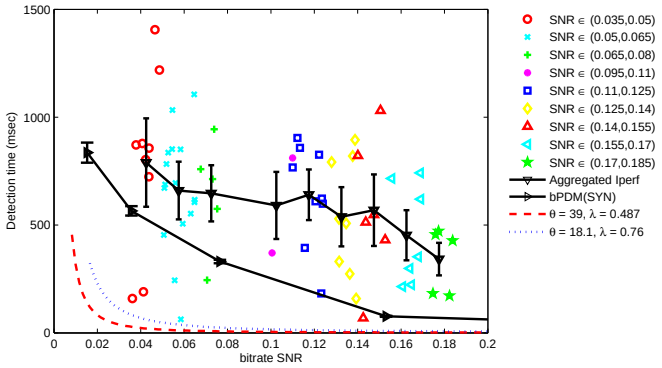

Fig. 4. Comparing detection time for the emulated Iperf attacks and the synthetic TCP SYN attacks, with the Iperf attacks grouped by similar bitrate SNR values. Theoretical detection times also plotted for comparison.

The detection times for the 67 individual Iperf attacks are plotted as open symbols in Figure 4. The individual attacks are grouped by bitrate SNR to allow us to investigate the relationship between detection time and bitrate SNR. The data is partitioned in 0.015 bin increments, so that ten bins span the bitrate SNR range from 0.035 to 0.185. Each bin is plotted using a different symbol in Figure 4, *i.e.* data points that have bitrate SNR values between 0.035 and 0.05 are represented using red circles, data with bitrate SNR values between 0.05 and 0.065 by cyan x's, and so on. The aggregated Iperf line in Figure 4 plots the mean of each bin, and the error bars give the standard deviation for each bin. We find that the time to detection decreases as the bitrate SNR increases for these Iperf attacks, as it did for the synthetic attacks. The large error bars of the aggregated Iperf plot prevent further statistical analysis.

3) Comparing simulated and emulated traces to theory: We have found that the bPDM time to detection decreases as the bitrate SNR increases in the case of both the simulated and emulated attacks. In this section, we show that the time to detection for the underlying theoretical model follows the same general trend. We recall that the sequential probability ratio test (SPRT), described in Section III, is employed by the bPDM for both the packet-rate and packet-size features. For the packet-rate SPRT, which is based on the generalized Poisson distribution (GPD) model as in (7) and (9), the theoretical time to detection is the average sample number (ASN) function under hypothesis H_1, and is derived in Appendix C. The ASN under H_1 is a function of the shifted GPD (sGPD) model parameters $\{\theta, \lambda, r\}$.

The same set of sGPD parameters is used to derive the bitrate SNR, as defined in (20). The mean of the GPD is $\theta/(1-\lambda)$ (see (11)), which corresponds to the number of packets in the background traffic. Similarly, the attack parameter r corresponds to the number of attack packets. Furthermore, we assume that the attack uses constant 544-bit packets, and adopt a simplified model for the background traffic wherein 66.6%

of packets are 480-bit, and 33.3% packets are 12000-bit. Thus, the bitrate SNR is computed as

$$\text{bitrate SNR}\Big|_{\text{bPDM}} = \frac{r \cdot 544}{(2/3 \cdot 480 + 1/3 \cdot 12000)\theta/(1-\lambda)}, \quad (21)$$

where for a fixed θ and λ, a greater r corresponds to a higher bitrate SNR. The theoretical detection times for $\{\theta = 39, \lambda = 0.487\}$ and $\{\theta = 18.1, \lambda = 0.76\}$, which correspond to the parameter values for the synthetic TCP SYN attacks and a 30 Mbps Iperf attack, respectively, are plotted in Figure 4 as a dashed red line and a dotted blue line, respectively.

We see that the theoretical time to detection trends as in the experimental cases: the time to detection decreases as the bitrate SNR increases. Thus, we find that attacks with higher bitrate SNR values are detected more quickly for the simulated and emulated attacks, which is consistent with what is predicted by the underlying theoretical model.

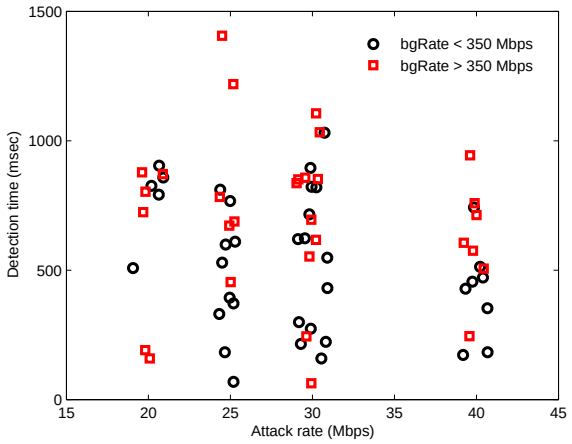

Fig. 5. Detection time for the Iperf attacks as a function of attack rate (in Mbps), grouped by high- and low-levels of background traffic.

B. Effect of Attack rate (Mbps) on Time to Detection

In the previous section, we saw that the time to detection decreases as the bitrate SNR increases. The bitrate SNR as defined in (20) consists of two components: the attack rate (in Mbps) and the background traffic (in Mbps). We now investigate the effect of each of the individual components on the time to detection, and find that for a constant level of background traffic, the time to detection decreases as the attack rate increases. The effect of varying background traffic for a constant attack rate is considered in the next section.

As in Section V-A2, we again aggregate the emulated Iperf data, this time to better examine the effect of the attack rate. Specifically, we group the detection times of emulated attacks by level of background traffic: data points with background traffic of less than 350 Mbps constitute the first group (low-level), and data points with background traffic greater than 350 Mbps are the second group (high-level)[3]. The detection times of the Iperf attacks, grouped by background traffic levels, are plotted as a function of the attack rate (in Mbps) in Figure 5. The attack rates for the data points in Figure 5 are either 20, 25, 30 or 40 Mbps, but are plotted with random shifts ($\in (-1, 1)$ Mbps) to improve the visibility of the data points.

To measure the association of the time to detection with the attack rate, we compute the Pearson product-moment correlation coefficient r [33] of the time to detection and backgroud traffic level, as well as the time to detection and attack strength. The correlation coefficient is independent of the scale of measurement, and its value ranges from -1.00 to +1.00[4]. An r value of 0.00 represents no correlation between the two variables, while a value of -1.00 or +1.00 indicates perfect predictability.

[3]We considered finer groupings of background traffic (100-200 Mbps, 200-300 Mbps, etc.), but the results were inconclusive due to insufficient data points in each bin.

[4]Given the variables Y and X, we define the standardized variables $Z_Y = (Y - \overline{Y})/S_Y$ and $Z_X = (X - \overline{X})/S_X$, where ($\overline{Y}, \overline{X}$ and (S_Y, S_X) represent the sample means and standard deviations of the variables Y and X, respectively. The Pearson r is then computed as $r = \sum Z_X Z_Y/(N-1)$ [33].

For this grouping of the data points, the correlation coefficients between detection time and attack rate (Mbps) for the high- and low-level background traffic are -0.3050 and -0.0781, respectively. The correlation coefficients and their associated p-values, along with the sample size for each group, are listed in Table I. The p-value is a measure of statistical significance, *i.e.* the probability that the result occured due to chance rather than an underlying cause. A p-value of less than 0.10 indicates that there is statistical evidence for the model being considered, or hypothesis being proposed, at the 10% significance level. We see that the detection time and attack rate are weakly negatively correlated with statistical signficance for $R_{bg} < 350$, suggesting that for a specific background level of traffic, the time to detection decreases as the attack rate increases.

Note that this weak negative correlation between time to detection and attack rate also holds in the case of the set of synthetic TCP SYN attacks discussed in Section V-A1 above, where the time to detection decreased as the bitrate SNR increased. This result is intuitive because in both cases, the effect of attack rate is examined for a given level of background traffic.

Interestingly, for a higher level of background traffic ($R_{bg} > 350$), the correlation becomes very small ($(r = -0.0781) \sim 0$) and looses statistical signficance ($(p = 0.68) > 0.10$). In order to support the claim that the decrease in correlation is reflective of a legitimate trend, and not merely an artifact, we now consider the variances of the detection times, grouped by background traffic level, as a function of the attack rate. Table II shows that for the higher-level of background traffic, the variance in detection time is greater than in the low-level background traffic case for all the attack rates. In other words, as the level of background traffic increases, the attack rate is less predictive of the time to detection of the bPDM.

C. Effect of Background traffic (Mbps) on Time to Detection

Now we consider the second component of the bitrate SNR, and investigate the effect of the background traffic (in Mbps)

TABLE I
CORRELATION COEFFICIENTS AS A FUNCTION OF BACKGROUND RATE, R_{bg}.

Bin (Mbps)	$R_{bg} < 350$	$R_{bg} > 350$
Sample size	37	30
r	-0.3050	-0.0781
p-value	0.0664	0.6815

TABLE II
VARIANCE OF DETECTION TIMES AS A FUNCTION OF BACKGROUND RATE, R_{bg}, GROUPED BY ATTACK RATE R_{att}.

$R_{att}(Mbps)$	40	30	25	20
$R_{bg} < 350$ Mbps	$3.4e4$	$8.15e4$	$5.8e4$	$2.44e4$
$R_{bg} > 350$ Mbps	$4.8e4$	$10.0e4$	$13.2e4$	$11.4e4$

on the detection time. We find that, for a constant attack rate, the time to detection increases as the background traffic increases. The detection times, grouped by attack rate, are plotted as a function of the background traffic in Figure 6.

As in the previous section, in order to measure the association of the time to detection to the background traffic, we compute the correlation coefficients for each attack rate, which are listed in Table III. We see that for the 40, 30 and 25 Mbps attack rates, the correlation coefficient is positive, and this implies that the time to detection increases as the background traffic increases. Yet we find that the correlation coefficient in the case of the 20 Mbps attack is negative. However, this anomalous result may be explained by the relatively small sample size; furthermore, we note that the p-value is significantly greater than that in the other cases, which means that the possibility of this conclusion being a result of chance is much higher. More data is therefore required to reach a firmer conclusion. Thus, we find that for higher attack rates, the time to detection generally increases as the background traffic level increases.

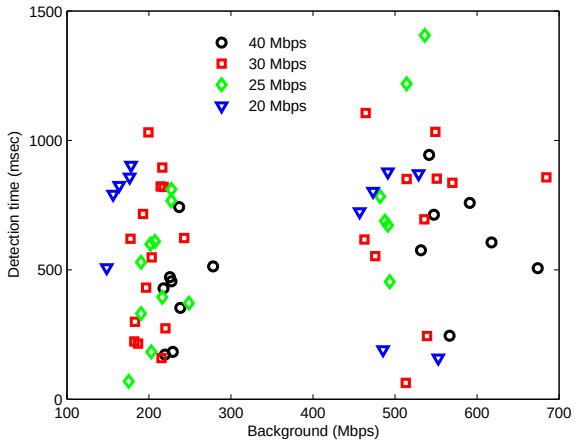

Fig. 6. Detection time for the Iperf attacks as a function of background traffic (in Mbps), grouped by attack rate.

TABLE III
CORRELATION COEFFICIENTS AS A FUNCTION OF ATTACK RATE.

Attack rate (Mbps)	40	30	25	20
Sample size	15	26	17	11
r	0.4556	0.2849	0.6503	-0.3598
p-value	0.0879	0.1674	0.0064	0.2772

D. Comparing bPDM to Prior Methods

We compare the bPDM to a selected set of detection schemes as described in Section II, and find that our algorithm performs comparably to or better than the other detection mechanisms we consider, while mitigating key drawbacks of the latter. Recall that the bPDM only requires 2-3 seconds of background-only traffic for training, updates its model parameters in real-time, and requires no human intervention when it is initially deployed. First, we focus on the Modeled Attack Detector (MAD) [35], a time-domain sequential scheme which adopts a simpler Poisson model and only uses the packet-rate feature to detect attacks. The MAD requires at least 10-12 seconds of background-only data to initialize the estimate of its background parameter λ (the rate), which does not update during the algorithm's operation. Though the attack parameter r (the rate change due to the attack) in the MAD is updated in real-time, the fact that the background parameter remains static necessitates a longer training phase as compared to the bPDM, which requires 2-3 seconds of training data. Furthermore, significant evolutions of normal network traffic are often flagged by the MAD as attacks since the background parameter is not automatically updated. In contrast, the bPDM updates its model parameters in an online fashion, and employs the packet-size feature to minimize false alarms.

The second scheme we consider is the Periodic Attack Detector [35], a spectral-domain scheme that exploits the near-periodic nature of attacks. The PAD is the sequential version of the spectral-domain scheme developed by He et al [16], *i.e.* the underlying models and development in [16] were adapted into a sequential framework as described in Section III. Like the MAD, the PAD uses a longer length of training data, compared to the bPDM, and then requires that the test data be statistically similar to the training data.

The entropy-based scheme by Feinstein et al [14] is the third scheme we examine. Unlike the bPDM, which develops statistics based only on aggregate traffic features, this third scheme computes the entropy of *flow-separated parameters* and compares the decision statistics against a threshold in a sequential framework. We simulate this scheme by computing the entropy of the destination IP address using non-overlapping batches of 5000 packets. Furthermore, the computationally complexity associated with extracting the flow-separated parameters from the network trace is noticably higher compared to the operation of the bPDM, MAD and PAD.

We compare the performance of these three schemes to the bPDM using the bitrate SNR metric. First, we compare the four methods' performance when tested on a reflector attack [3] with a bitrate SNR of 0.0678, which sends echo reply packets targeted to a victim within Los Nettos and lasts for

TABLE IV
NUMERICAL RESULTS FOR COMPARISONS OF THE BPDM TO OTHER METHODS.

Scheme	# FP	TD	Drawback
bPDM	0	336	Limited training required
MAD [35]	2	280	Longer training phase required
IP Entropy [14]	0	400	Flow-separation required
PAD [16]	1	340	Higher complexity due to FFT and longer training phase required

204 seconds. The results are tabulated in Table IV, where the second and third columns are the number of false positives (# FP) and the time to detection (TD, in msec), respectively. We note that the detection time for the method by Feinstein et al [14] may be shorter or longer if different simulation parameters are employed. Comparison of the four methods shows that the time to detection for the bPDM is comparable to or shorter than those of the other three.

Next, we employ the set of synthetic TCP SYN attacks to compare these three detection schemes to the bPDM. Figure 7 shows the detection time as a function of the bitrate SNR for the bPDM, MAD, and PAD schemes. The IP entropy scheme [14] is not included in the comparison because the synthetic attacks were generated without using source and destination IP addresses and port numbers. The label "bPDM(SYN)" denotes the performance of the bPDM on the set of synthetic TCP SYN attacks, and similarly for MAD and PAD. Each of the three algorithms was run on 8 distinct synthetic traces of a specific bitrate SNR, and the resulting mean values are plotted in Figure 7 along with error bars representing the standard deviation associated with the mean detection time for each bitrate SNR. Notice that, as expected, the time to detection decreases as the bitrate SNR increases. In this particular case, for the set of simulated TCP SYN attacks, the attack level (in Mbps) increases while the background remains constant. We find that the bPDM generally outperforms both the MAD and the PAD, although for lower bitrate SNR values, the detection times are comparable to that of the MAD. The spectral-based PAD consistently has the highest, though comparable, detection times. We note that we achieve these comparable or better detection times without the drawbacks of the other methods, as described in Sections I and II (see Table IV).

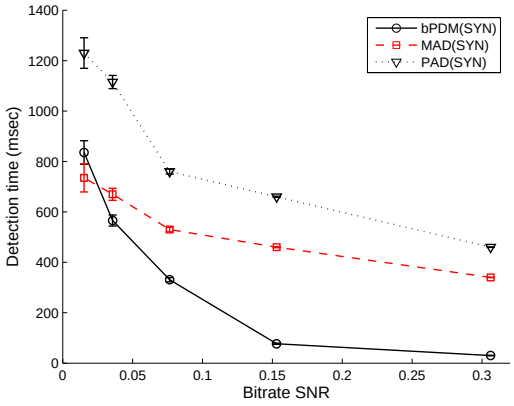

Fig. 7. Comparing the time to detection (in msec) for the bPDM, MAD and PAD detection algorithms using the set of simulated synthetic TCP SYN attacks that employ 68-byte packets.

E. Validating the need for minimal training in bPDM

The previous section showed that the bPDM outperforms the MAD, PAD, and entropy-based Feinstein schemes. In this section, we further explain two important advantages of the bPDM: it requires limited training, and its model parameters automatically update in an on-line fashion, as compared to other existing schemes.

As described in the previous section, the MAD requires a 10-12 second or more training period. We note that the MAD is based on the simpler pure Poisson model, with a single parameter λ that is estimated as the background level of traffic based on the training data, and assumed to remain constant thereafter. The level of traffic with an attack is modeled by $\lambda+r$, wherein r captures the effect of an attack and is updated in real-time. As a result, both marked changes in the level of background traffic and actual attacks are flagged as attacks, since the increase in traffic volume is captured by updating only the attack parameter. In constrast, the bPDM uses the generalized Poisson distribution with two varying parameters (θ, λ) to account for changes in the level of background traffic. These parameters update individually via (12) and (15) to incorporate the changing background traffic. The attack parameter r, as in the case of the MAD, updates as in (14), and is compared to the background to identify the presence of an anomaly. Importantly, in the bPDM, the packet-size SPRT is incorporated in addition to the packet-rate to ensure that an increase in traffic volume without a corresponding change in the packet-size distrbution is *not* flagged as an attack. Thus, if the attack parameter is updated and signals an attack due to an increase in the background traffic, the packet-size SPRT, which would not have crossed the upper threshold, can ensure that only a warning is raised. We note that even if the packet-size SPRT had likewise been incorporated into the MAD to reduce false alarms, the static nature of the background parameter would still necessitate a minimum 10-12 second training period for the MAD, while the bPDM only requires 2-3 seconds.

The second detection scheme to be considered, the PAD, also requires a longer period of training data than the bPDM, which it uses to characterize the spectral-domain features of normal background-only traffic. In the testing phase, the presence of frequency-domain components that were not present in the background-only traffic spectrum are used to detect attacks. We note that the PAD is sensitive to significant changes in the background traffic, and thus the data used to train and test the algorithm must be statistically similar. This is not the case for the bPDM, which initializes using a limited amount of training data, but then automatically updated its parameters as the network traffic evolves.

The MNA-CUSUM is a non-parametric sequential algorithm developed by Tartakovsky et al [34], which requires a non-trivial amount of overhead when initially deployed: it filters incoming packets by size and uses individual channels and decision statistics (analogous to the log-likelihood ratio

in the SPRT) to detect an attack rapidly. Because the decision statistics are based on score functions that update periodically using a parameter update method, similar to that of the bPDM, for each of the channels, the initial deployment of the MNA-CUSUM involves hand-tuning of the thresholds of each channel to meet the false-alarm requirements. An alternative to the hand-tuning of thresholds could be an explicit search over the parameter space, which has not been implemented in [34] but would presumably be computationally intensive due to the multiple channels employed. Recall that the bPDM is initially deployed with no hand-tuning, since the initial parameter estimates are automatically computed using (12), given a limited amount of background-only training data.

Thus, our algorithm requires only up to 2-3 seconds of training data, as compared to the 10-12 seconds needed by the MAD and PAD, since the H_0 and H_1 update window sizes are 1 second long as described in Section IV. And unlike MNA-CUSUM, our algorithms requires only a few parameters and we have demonstrated automatic training. Thus, we find that the bPDM's use of limited training data and automatic updating of its model parameters in real time results in its being free from the drawbacks of other existing detection methods.

F. Robustness of bPDM to a smart attacker

As described in the previous sections, the bPDM uses the packet-size as a feature for detection, and to reduce false positives. We now consider the *smart adversary* scenario, wherein the attacker constructs an attack whose distribution of packet-sizes matches that of the background traffic. For this purpose, we create a set of smart adversary synthetic attacks wherein the attack stream uses a constant bitrate, but with a distribution of packet sizes that is drawn from the bimodal distribution described in [29]. We recall that the packet-size distribution of nominal Internet traffic has been characterized in [29] as mostly bimodal, and that an examination of our background trace data validates the bimodal distribution of packet-sizes. Attackers do not use this approach today, as in general they cannot perfectly guess the packet size distribution on the monitored link. We therefore present these smart attacks to consider one possible set of countermeasures against our detection mechanism.

It must be noted that, although the smart adversary actively manipulates only the packet-size distribution, the smart attacks affect both the packet-rate and the packet-size aspects of the bPDM:

1) The packet-sizes used by a smart adversary are drawn from a bimodal distribution that resembles normal Internet traffic. This results in the entropy of the packet-size distribution in the case of an attack being similar to that in the background-only case, reducing the effectiveness of the packet-size SPRT.
2) Recall that the synthetic TCP SYN attacks employ 68-byte packets. In drawing from a bimodal distribution, the smart adversary uses a range of packets including several that are larger than 68 bytes. This variety of packet sizes implies that for a fixed attack rate, say 60 Mbps, the smart attack has a smaller number of packets per second, as compared to a TCP SYN attack. This should challenge the packet-rate SPRT employed by the bPDM.

Despite these challenges, we will see that the bPDM can still detect attacks from a smart adversary.

Figure 9(a) shows the bPDM detection times for the smart, denoted "bPDM(smart)," and TCP SYN, denoted "bPDM(SYN)," simulated attacks as a function of the bitrate SNR. We recall that the (SYN) label refers to the set of synthetic TCP SYN traces wherein fixed 68-byte packets are employed; correspondingly, the (smart) label denotes packets drawn from the bimodal packet-size distribution. In particular, the bPDM algorithm was run on 8 synthetic TCP SYN and smart traces each, all of a specific bitrate SNR. The mean detection times are plotted in Figures 9(a), wherein the error bars represent the standard deviation of the detection times. We find that the bPDM detects the synthetic smart attacks, albeit with longer detection times than TCP SYN attacks. The longer detection times for the smart attacks are not surprising given that the smart adversary is designed as a countermeasure to reduce the effectiveness of the bPDM.

G. Bitrate SNR versus packet SNR

An alternative metric, the packet SNR, is used by He et al [16] to evaluate their methods. In this section, we compare the packet SNR to the bitrate SNR, and we find that the latter is a more effective metric for this application. The packet SNR is defined as [16]

$$\text{packet SNR} = \frac{\text{\# of attack packets}}{\text{\# of background packets}} = \frac{\sum_{S\in\mathcal{S}_A} M_S}{\sum_{S\in\mathcal{S}_B} M_S}, \tag{22}$$

where $\mathcal{S}_A$, $\mathcal{S}_B$ and M_S are as defined for (20). This metric is thus defined in terms of the packet-rates of both the attack and the background traffic, rather than the bitrate (in Mbps) as in the case of the bitrate SNR. Both the packet and bitrate SNRs are equivalent metrics for the TCP SYN attacks described in Section V-D, and in fact for any anomalies that employ fixed-size packets. To compare the metrics' efficacy in this case, we revisit Figure 7 in Section V-D as Figure 8, this time employing the packet SNR instead of the bitrate SNR. The time to detection for the three methods, averaged over 8 sets of synthetic TCP SYN attacks, are plotted as a function of packet SNR in Figure 8 with error bars representing the standard deviation of the detection times. In comparing Figures 7 and 8, we note that they are simply scaled and shifted versions of each other, which shows that the packet SNR is equivalent to the bitrate SNR in the case of attacks with *fixed-size packets*.

However, this is not always the case; we now consider an attack due to a *smart adversary*, as introduced in the previous Section V-F. We compare Figures 9(a) and 9(b), wherein the bPDM time to detection is plotted for the smart and TCP SYN synthetic attacks as a function of bitrate SNR and packet SNR, respectively. Note that in Figure 9(b), the smart adversary attacks are detected more quickly than the corresponding TCP SYN attacks. This is a very counterintuitive result, since, as described above, the smart adversary represents a set of

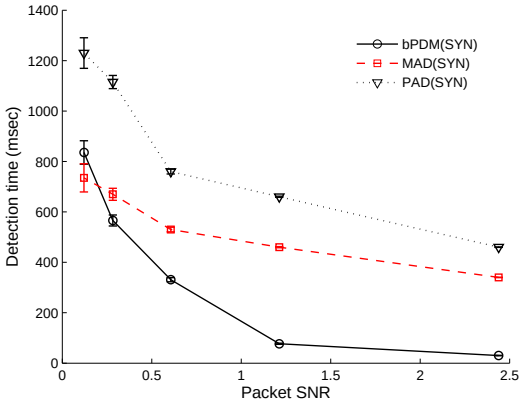

Fig. 8. Comparing the time to detection (in msec) for the bPDM, MAD and PAD detection algorithms as a function of the packet SNR metric as defined in [16].

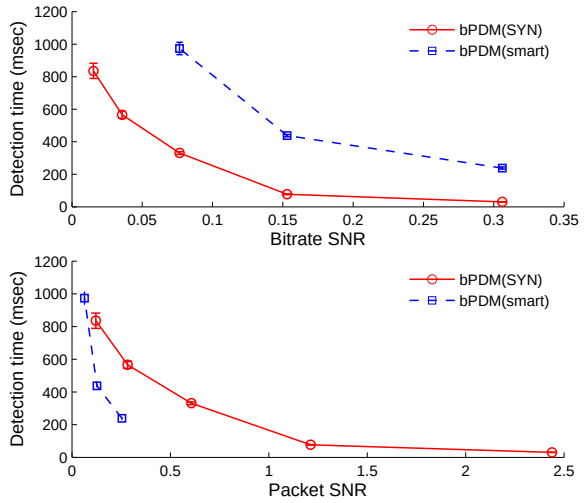

Fig. 9. Comparing the bPDM detection times for the set of synthetic TCP SYN and smart attacks as a function of (a) packet SNR and (b) bitrate SNR.

countermeasures against our detection mechanism. In contrast, in Figure 9(a), the TCP SYN attacks are shown to be detected markedly and uniformly faster than the smart adversary attacks for the entire range of bitrate SNR values, which is the result we expect. We thus conclude that the bitrate SNR is an effective metric for comparison and evaluation, and better than the packet SNR.

H. Validation of Synthetic Attacks

In order to confirm the conclusions drawn from the bPDM's performance on simulated attacks, we here test the bPDM using three real network attacks, which were captured in the wild and are available through PREDICT, and three proxy real attacks, constructed of real denial-of-service attacks (DoS) and real background traffic streams combined using Stream Merger [21]. We find that the detection times for the real and proxy real attacks closely resemble those of the synthetic attacks.

The three real network attacks considered were collected in varying network conditions, but all six attacks employ either 15-byte, 60-byte or 68-byte fixed-size attack packets. In this

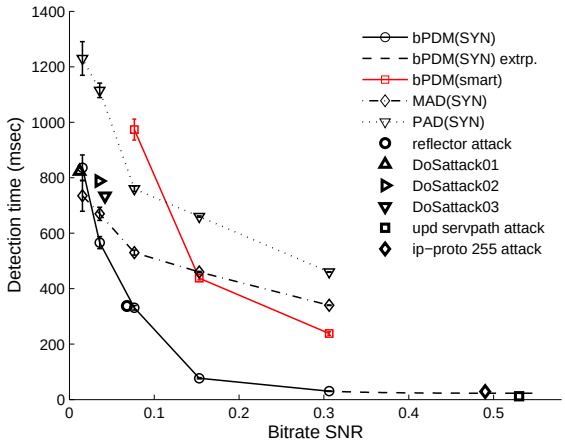

Fig. 10. Comparing the bPDM time to detection (in msec) of the real network attacks to the bPDM, MAD and PAD detection times for the simulated synthetic attacks, TCP SYN and smart adversary.

respect, they resemble the set of synthetic TCP SYN attacks. Thus, we expect the detection times of the real attacks to resemble those of the SYN attacks. Table V summarizes the attack details; the bitrate SNRs for the real attacks range from 0.012 to 0.53, with varying attack and background traffic levels (in Mbps).

The detection times of the individual real and proxy real attacks are plotted (as open, unconnected symbols) in Figure 10 for comparison to the detection times of the synthetic traces for the bPDM, MAD and PAD detection schemes. As for earlier plots, the points for the synthetic traces, *i.e.* bPDM(SYN), bPDM(smart), MAD(SYN) and PAD(SYN), represent the mean detection times, with the error bars providing the standard deviation. The bPDM detection times for the real and proxy real network attacks as shown in Figure 10 were obtained by running the algorithm on the attacks with 10-12 seconds of background traffic before the onset of the attack. Furthermore, the plot of the detection times for the synthetic TCP SYN attacks has been extrapolated (plotted using dashed black line) to compare the TCP SYN detection times to those of the real ip-proto 255 and UDP Servpath attacks. Of the models considered, the best fit for the synthetic TCP SYN detection times was found to be an exponential one, which is described as $f(\text{detection time}) = 22.6 + 1053e^{-16.08 \cdot \text{bitrate SNR}}$.

For all real and proxy real attacks, we see that a higher bitrate SNR corresponds to a lower time to detection, and further note that the actual (and extrapolated) detection times for the synthetic TCP SYN attacks are consistent with the times to detection of the real and proxy real network attacks.

I. Considering the Effect of Hop Count on Time to Detection

Recall that Figure 4 in Section V-A1 shows that the detection times for the synthetic TCP SYN attacks are consistently lower than the averaged Iperf detection times, described in Sections V-A1 and V-A2, respectively. We conjecture that this disparity might be explained by the different hop counts between the two sets of attacks: the synthetic TCP SYN attacks are constructed using Stream Merger, and thus the synthetic

TABLE V
BPDM DETECTION RESULTS FOR REAL NETWORK ATTACKS.

Symbol	Trace	Attack (Mbps)	Background (Mbps)	bitrate SNR	TD (msec)	Description
◇	[3]	34.45	69.86	0.49	29	ip-proto 255 attack
○	[3]	2.11	31.12	0.678	338	reflector attack
□	[1]	21.6	40.75	0.53	12	udp servpath attack
△	[2]	3.84	320	0.012	823	
▷	[2]	11.2	320	0.035	788	proxy real DoS attacks
▽	[2]	13.44	320	0.042	734	

traces have traversed the equivalent of one router. In constrast, the emulated Iperf attacks from CSU to USC, are captured at Los Nettos (see Appendix A) after 8-10 hops. A greater number of hops corresponds to a longer network routing path, which mixes the attack packets to a greater degree. We believe that the greater hop count may be responsible for the fact that the TCP SYN detection times are consistently lower than most of the individual Iperf attack detection times.

VI. CONCLUSIONS

We have developed the bivariate Parametric Detection Mechanism (bPDM), which can detect anomalies and low-rate attacks in a few seconds. This approach allows the real-time estimation of model parameters, and only requires 2-3 seconds of background-only traffic for training. Incorporating the packet rate and packet size features enables us to detect anomalies in encrypted traffic and avoid state-intensive flow tracking, since our method does not use flow-separated traffic, and combining these same two features also eliminates most false positives. We have evaluated our methods using synthetic traces and emulated Iperf attacks, and find that the bPDM can detect attacks in a few seconds. The detection times for the synthetic attacks are validated using real and proxy real network attacks, and the bitrate SNR is shown to be not only an effective metric for evaluating anomaly detection methods, but also a better one than the previously proposed packet SNR metric. For all the datasets considered, as well as the underlying theoretical model, we find that the time to detection decreases as the bitrate SNR increases. Furthermore, we examine the effect of the individual components of the bitrate SNR on the time to detection: as the attack rate increases, the detection time decreases; as background traffic level increases, the time to detection decreases.

REFERENCES

[1] attack-servpath-udp22-20061106, available through PREDICT.
[2] DoS_80_timeseries-20020629, available through PREDICT.
[3] DoS_traces_20020629, available through PREDICT.
[4] UniformAttack_Traces_Generated20070821-20041202, PREDICT.
[5] P. Barford, J. Kline, D. Plonka, and A. Ron. A signal analysis of network traffic anomalies. In *Proceedings of the SIGCOMM Internet Measurement Workshop*, France, November 2002.
[6] M. Basseville and I. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice-Hall, Englewood Cliffs, NJ, 1993.
[7] M. Brunner. *Service Provision: Technologies for Next Generation Communications*, chapter Basic Internet Technology in Support of Communication Services. Wiley Series on Communications Networking and Distributed Systems. Wiley, 2004.
[8] Y. Chen and K. Hwang. Spectral Analysis of TCP Flows for Defense against Reduction-of-Quality Attacks. In *Proc. of the IEEE Intl. Conf. on Communications*, Glasgow, Scotland, June 2007.
[9] P. Consul. *Generalized Poisson Distributions: Applications and Properties*. Marcel Dekker Inc., New York, NY, 1989.
[10] P. Consul and F. Famoye. *Lagrangian Probability Distributions*. Birkhauser, Boston, MA, 2006.
[11] H. Cramér. *Mathematical Methods of Statistics*. Princeton University Press, 1946.
[12] N. Duffield, P. Haffner, B. Krishnamurthy, and H. Ringberg. Rule-based anomaly detection on IP flows. In *Proceedings of IEEE INFOCOM*, Rio de Janeiro, Brazil, April 2009.
[13] J. Ellis and T. Speed. *The Internet Security Guidebook: From Planning to Deployment*. Academic Press, 2001.
[14] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred. Statistical approaches to DDoS attack detection and response. In *Proc. of DARPA Information Survivability Conf. and Exposition*, pages 303–314, 2003.
[15] Z. Govindarajulu. *Sequential Statistics*. World Scientific Publishing, Singapore, 2004.
[16] X. He, C. Papadopoulos, J. Heidemann, U. Mitra, and U. Riaz. Remote detection of bottleneck links using spectral and statistical methods. *Computer Networks*, 53:279–298, 2009.
[17] A. Hussain, J. Heidemann, and C. Papadopoulos. Identification of repeated denial of service attacks. In *Proceedings of the Conference on Computer Communications (INFOCOM)*, Barcelona, Spain, April 2006.
[18] C. Jin, H. Wang, and K. Shin. Hop-count filtering: An effective defense against spoofed DoS traffic. In *Proceedings of Conference on Computer and Communications Security*, Washington DC, October 2003.
[19] J. Jung, B. Krishnamurthy, and M. Rabinovich. Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites. In *11th International WWW Conference*, Honolulu, HI, May 2002.
[20] J. Jung, V. Paxson, A. Berger, and H. Balakrishnan. Fast portscan detection using sequential hypothesis testing. In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 2004.
[21] P. Kamath, K.-C. Lan, J. Heidemann, J. Bannister, and J. Touch. Generation of high bandwidth network traffic traces. In *Proceedings of the 10th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, Fort Worth, TX, 2002.
[22] S. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall PTR, 1993.
[23] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. In *Proceedings of ACM SIGCOMM*, Philadelphia, PA, August 2005.
[24] A. Nandi. Higher order statistics for digital signal processing. In *IEE Colloquium on Mathematical Aspects of Digital Signal Processing*, pages 6/1–6/4, London, UK, February 1994.
[25] G. Nychis, V. Sekar, D. G. Andersen, et al. An empirical evaluation of entropy-based traffic anomaly detection. In *Proceedings of the 8th ACM SIGCOMM Internet Measurement Conference*, pages 151–156, Vouliagmeni, Greece, October 2008.
[26] S. Ramanujan. *The Lost Notebook and Other Unpublished Papers*. Narosa, New Delhi, India, 1988.
[27] J. Rodriguez, A. Briones, and J. Nolazco. Dynamic DDoS mitigation based on TTL field using fuzzy logic. In *Proceedings of 17th International Conference on Electronics, Communications and Computers*, Cholula, Mexico, February 2007.
[28] M. Shoukri. *Estimation Problems for Some Generalized Discrete Distributions*. PhD thesis, University of Calgary, Calgary, Canada, 1980.
[29] R. Sinha, C. Papadopoulos, and J. Heidemann. Internet packet size distributions: Some observations. Technical Report ISI-TR-2007-643, University of Southern California, Los Angeles, CA, USA, May 2007.
[30] V. Siris and F. Papagalou. Application of anomaly detection algorithms for detecting SYN flooding attacks. In *Proceedings of IEEE GLOBECOM*, Dallas, TX, November 2004.
[31] A. Soule, K. Salamatian, and N. Taft. Combining filtering and statistical

methods for anomaly detection. In *Proceedings of the 2005 Internet Measurement Conference*, Berkeley, CA, October 2005.
[32] M. Stoecklin, J.-Y. L. Boudec, and A. Kind. A two-layered anomaly detection technique based on multi-model flow behavior models. *Lecture Notes in Computer Science (PAM)*, 4979:212–221, 2008.
[33] B. Tabachnick and L. Fidell. *Using Multivariate Statistics (5th Edition)*. Allyn and Bacon, 2006.
[34] A. Tartakovsky, B. Rozovskii, R. Blazek, and H. Kim. A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods. *IEEE Transactions on Signal Processing*, 54(9):3372–3382, 2006.
[35] G. Thatte, U. Mitra, and J. Heidemann. Detection of low-rate attacks in computer networks. In *Proceedings of IEEE 11th Global Internet Symposium*, Phoenix, AZ, April 2008.
[36] G. Thatte, U. Mitra, and J. Heidemann. Parametric methods for anomaly detection in aggregate traffic. *IEEE/ACM Transactions on Networking*, In Preparation, 2008.
[37] M. Thottan and C. Ji. Anomaly detection in IP networks. *IEEE Transactions on Signal Processing*, 51(8):2191–2204, August 2003.
[38] H. V. Trees. *Detection, Estimation, and Modulation Theory, Part I*. John Wiley, New York, 1968.
[39] A. Wagner and B. Plattner. Entropy based worm and anomaly detection in fast IP networks. In *Proceedings of the SIG SIDAR Graduierten-Workshop uber Reaktive Sicherheit*, Berlin, Germany, July 2006.
[40] A. Wald. *Sequential Analysis*. John Wiley, New York, 1947.
[41] H. Wang, D. Zhang, and K. Shin. Detecting SYN flooding attacks. In *Proceedings of the Conference on Computer Communications (INFOCOM)*, New York, NY, June 2002.

APPENDIX A
GENERATION OF IPERF ATTACKS

Our evaluation of the bPDM uses controlled Iperf attacks in varying Internet traffic mixes. These 80-second UDP attacks are generated at a fixed attack rate (in Mbps) and employ 345-byte fixed-size attacks packets. The Iperf attacks originate at Colorado State University (USC) and are destined for the University of Southern California, with 10 routers traversed between the source and destination as determined by traceroute. The network packet traces consisting of these attacks are captured via port-mirroring, and with capture machines that use DAG cards and that see both incoming and outgoing university traffic. In particular, we use one link (out of five) at Los Nettos, a regional ISP in the Los Angeles area serving both commercial and academic institutions. The traces are collected at Los Nettos with a timing precision of 0.1 microsecond, and is due to the accuracy of the Endace DAG network card. The link we use captures bidirectional traffic, but since the bPDM operates on a unidirectional traffic stream, the incoming traffic is filtered from the bidirectional traffic using a complete list destination IP subnets for the University of Southern California. Once the incoming traffic has been isolated, the bPDM exploits only the aggregate traffic fields, the timestamp and the packet-size, which yield the packet-rate and entropy of packet-size distribution statistics.

We collected four datasets, each 3 hours long, and consisting of an average of 15 Iperf attacks with attack rates of 20, 25, 30 and 40 Mbps. The experiments were conducted at different times of day; during weekend non-peak hours, and during busier weekday hours, to investigate the effect of different background traffic levels. We see qualitatively similar results across all the datasets, in the fact that the time to detection is uncorrelated with when the datasets were collected, and thus we do not analyze the dataset-partitioned data.

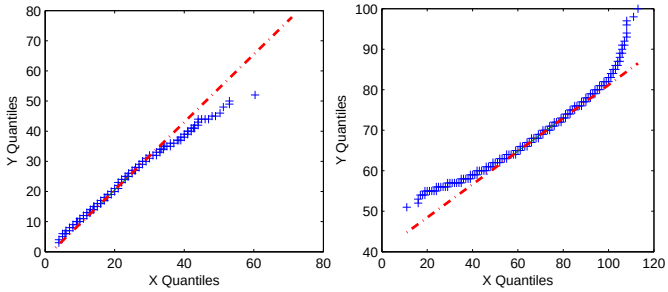

Fig. 11. Q-Q plots for real background traffic (left) and attack traffic (right) matched to the GPD and sGPD, respectively.

APPENDIX B
QUANTIFYING THE MODEL MISMATCH

The parametric models employed in the bPDM do *not* represent general Internet traffic. Despite this mismatch, we are able to successfully detect anomalies as seen in Section V. In this Appendix, we provide a brief analysis of the model mismatch using one of the real network attacks [3].

Since the start of the attack is known, the background traffic and attack traffic are individually compared to the GPD and sGPD, respectively. In particular, we quantify the model mismatch between our parametric model and the real data for this particular attack, using quantile-quantile (Q-Q) plots and the two-sample Kolmogorov-Smirnov test [11]. Recall that the Q-Q plot will be linear if and only if the two sets of samples being compared are drawn from the same distribution. Similarly, the null hypothesis of a two-sample Kolmogorov-Smirnov test declares that the two sets of samples are drawn from the identical distribution.

For the analysis of the H_0 hypothesis, the first sample set is the background traffic from the real network trace. The second sample set is synthetic data drawn from the GPD, and generated using the Inversion Method by Consul and Famoye [10]. The GPD parameters used to generate the synthetic data are estimated from the background traffic via the estimators in (12). Similarly, the H_1 hypothesis is analyzed using the attack traffic and synthetic data drawn from the sGPD with parameters estimated using (14) and (15).

The two plots in Figure 11, for the background and attack traffic, show that the Q-Q plots are not linear in either case. This nonlinearity implies that the background traffic is not GPD, and the attack traffic is not sGPD. Furthermore, the null hypothesis of the two-sample Kolmogorov-Smirnov test was rejected at the 5% significance level for both the background traffic and the attack traffic, further supporting our claim that the bPDM parametric models do not model general Internet traffic.

APPENDIX C
AVERAGE SAMPLE NUMBER ANALYSIS

The average sample number (ASN) function is used to evaluate the efficacy of the sequential test reviewed in Section III. The ASN function is simply the average number of samples required to make a decision by a particular test. For

the binary hypothesis test considered in our work, the ASN function is denoted $\mathbb{E}_i(N)$ for hypothesis H_i. We present an analysis of the ASN function for the GPD/sGPD hypothesis test since it is used to determine an alternative window size for parameter estimation. In order to compute the ASN function, we first define

$$z = \log \frac{p(x|H_1)}{p(x|H_0)}, \tag{23}$$

and denote $\mathbb{E}_\theta(z)$ to be the expected value $\mathbb{E}(z)$ of z when $\theta \in \{0, 1\}$ is the true hypothesis. In the determination of $\mathbb{E}_\theta(z)$, we avoid underflow and overflow errors by computing the probability mass function of the GPD (7) as

$$p(x|H_0) = \frac{\theta e^{-\theta}}{\theta + \lambda x} \prod_{n=1}^{x} \frac{(\theta + \lambda x)e^{-\lambda}}{n!}, \tag{24}$$

where each of the product terms are first computed, and then multiplied to yield the required probability. From the expressions in (3), we can solve for

$$\alpha = \frac{A-1}{A-B} \quad \text{and} \quad \beta = -\frac{A(B-1)}{A-B}\ , \tag{25}$$

and now obtain expressions for the ASN functions for each of the hypotheses as [15]:

$$\mathbb{E}_0(N) \;=\; \frac{\alpha \log B + (1-\alpha)\log A}{\mathbb{E}_0(z)}, \tag{26}$$

$$\text{and} \quad \mathbb{E}_1(N) \;=\; \frac{(1-\beta)\log B + \beta \log A}{\mathbb{E}_1(z)}. \tag{27}$$

Thus, for the GPD/sGPD hypothesis test, given the probability mass functions in (7) and (9), we derive

$$\begin{aligned} z \;=\; & (x-r-1)\log[\theta + \lambda(x-r)] + \lambda r + \log[x!] \\ & -(x-1)\log[\theta + \lambda x] - \log[(x-r)!], \end{aligned} \tag{28}$$

and then compute $\mathbb{E}_\theta(z)$ numerically. The ASN function for hypothesis H_1 is computed using (27) and is plotted in Figure 12 for typical values of θ and r for varying λ. Given that relatively high packet counts may exist for the packet-rate, we cannot compute the $\log(x!)$ term in (28) directly due to overflow and precision limitations. To that end, we employ the following approximation by Ramanujan [26]

$$\log n! \approx n\log n - n + \frac{\log(n(1+4n(1+2n)))}{6} + \frac{\log(\pi)}{2}. \tag{29}$$

The sizes of the update windows are chosen to be on the order of the ASN function for the respective hypotheses to ensure that the parameter estimates, computed using the observations in those windows, correspond to a decision being made by the SPRT. When an anomaly is detected, the ASN functions and update window sizes are recomputed using the current estimate $\hat{r}$. This process is outlined in Algorithm 1 in Section IV.

APPENDIX D
PERFORMANCE OF THE GPD/SGPD ESTIMATORS

The asymptotic variances and biases of the two estimators $\widehat{\theta}_0$ and $\widehat{\lambda}_0$ of the GPD are given in [9]. The estimators for the three parameters of the sGPD are given in Section IV, and their performance is analyzed here. The fact that the sGPD is discrete and involves the factorial function makes the analytic computation of the Cramer-Rao Lower Bound[5] (CRLB) not practical. Thus, following the analytic technique in [28], we derive the asymptotic variances and biases of the estimators of the sGPD parameters.

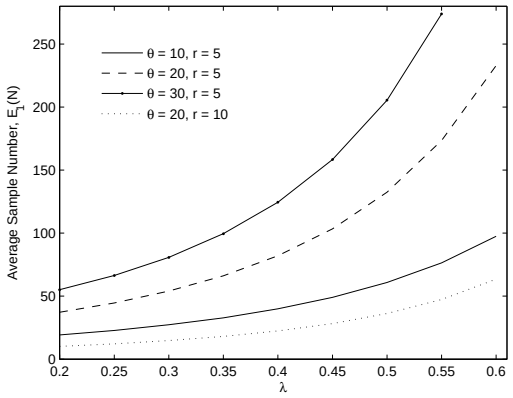

Fig. 12. For $\alpha = \beta = 10^{-5}$, ASN function $\mathbb{E}_1(N)$ computed for increasing λ and typical values of θ and r.

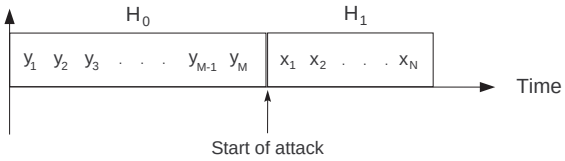

Fig. 13. Samples used to compute H_0 and H_1 parameter estimates.

Given the framework shown in Figure 13, we now derive the estimators for both the GPD and sGPD. For the GPD (under hypothesis H_0), the moment estimators are function of the sample mean $\overline{y}$ and second sample central moment s_y^2 which are computed using the M samples $\{y_1, \ldots, y_M\}$. Recall that the GPD parameter estimates, which satisfy the equations obtained by equating the population mean and population variance with the sample mean and sample variance, are given as [9]

$$\widehat{\theta}_y = \sqrt{\frac{\overline{y}^3}{s_y^2}} \quad \text{and} \quad \widehat{\lambda}_y = 1 - \sqrt{\frac{\overline{y}}{s_y^2}}\ . \tag{30}$$

For the sGPD (under hypothesis H_1), the moment estimators of the three model parameters (θ_1, λ_1, r) require using a higher-order moment, in addition to the mean and variance. Computing good estimates of $\hat{\mu}_3$ or $\hat{\mu}_4$ require a much larger number of samples compared to estimating $\hat{\sigma}^2$ [24], and thus employing these higher-order moments to implement a SPRT is infeasible; the time taken to compute good estimates of the third- or fourth-order moments is more than an order of magnitude greater than the average time to detection in our

[5]The CRLB [38] expresses a lower bound on the variance of estimators of a deterministic parameter. In its simplest form, as in the case of $\hat{r}$, the bound states that the variance of any unbiased estimator is at least as high as the inverse of the Fisher information, which is defined using the second derivative of the underlying probability density/mass function. An unbiased estimator which achieves this lower bound is said to be *efficient*.

hypothesis test in [36]. On the other hand, the ML estimates of the sGPD parameters may be obtained by numerically solving three non-linear equations for each observation, which is extremely computationally expensive. Thus we present an alternative estimation procedure for the model parameters under the H_1 hypothesis that is computationally lightweight.

Given the framework in Figure 13, we compute the estimates of the sGPD model parameters using both M samples from the GPD distribution and N samples from the sGPD distribution. The sGPD parameter estimates are given as

$$\begin{aligned}\widehat{r} &= \max\left\{\left\lfloor -\frac{\widehat{\theta}_y}{1-\widehat{\lambda}_y}+\overline{x}\right\rfloor, \min\{x_1,\ldots,x_N\}\right\} \\ &= \max\left\{\lfloor -\overline{y}+\overline{x}\rfloor, \min\{x_1,\ldots,x_N\}\right\} \quad (31)\\ \widehat{\theta}_x &= \sqrt{\frac{(\overline{x}-\widehat{r})^3}{s_x^2}} \text{ and } \widehat{\lambda}_x = 1-\sqrt{\frac{\overline{x}-\widehat{r}}{s_y^2}}\,. \quad (32)\end{aligned}$$

where the $\min\{\cdot\}$ function is a constraint on $\widehat{r}$ since the support of $P_x(\theta,\lambda,r)$ is $x \in \{r, r+1, r+2, \ldots\}$; for the purposes of our analysis, this constraint is ignored. Thus, the parameter estimates for the sGPD can be rewritten as

$$\widehat{\theta}_x = \sqrt{\frac{\overline{y}^3}{s_x^2}}\,,\ \widehat{\lambda}_x = 1-\sqrt{\frac{\overline{y}}{s_x^2}}\,, \quad (33)$$

$$\text{and} \quad \widehat{r} = -\overline{y}+\overline{x}\,. \quad (34)$$

A. Bias and Variance Computations

The GPD parameter estimates, specified in (12), are biased estimators. The fact that the r.v. is discrete and involves the factorial function makes the analytic computation of the CRLB intractable [9]. Thus, the asymptotic biases and the asymptotic variances of these moment estimators, derived in [28], are given as

$$\begin{aligned} b\left(\widehat{\theta}_y\right) &\simeq \frac{1}{4M}\left[5\theta+\frac{3\lambda(2+3\lambda)}{1-\lambda}\right], \quad (35)\\ b\left(\widehat{\lambda}_y\right) &\simeq -\frac{1}{4M\theta}\left[5\theta(1-\lambda)+\lambda(10+9\lambda^2)\right], \quad (36)\\ V\left(\widehat{\theta}_y\right) &\simeq \frac{\theta}{2M}\left[\theta+\frac{2-2\lambda+3\lambda^2}{1-\lambda}\right], \quad (37)\\ V\left(\widehat{\lambda}_y\right) &\simeq \frac{1-\lambda}{2M\theta}\left[\theta-\theta\lambda+2\lambda+3\theta^2\right], \quad (38)\end{aligned}$$

wherein they have been derived correct to terms of order M^{-1} and M^{-2}. Note that only samples corresponding to the GPD, namely $\{y_1,\ldots,y_M\}$, were used compute the estimates.

Following the analysis in [28], we now derive the biases and variances of the estimators for the sGPD. The parameter estimates of the three sGPD parameters, $\{\theta_x, \lambda_x, r\}$, are computed using both M and N samples, from the GPD and sGPD, respectively. We first compute

$$\begin{aligned}\mathbb{E}\{\widehat{r}\} &= \mathbb{E}\{-\overline{y}+\overline{x}\} \\ &= -\frac{\theta}{1-\lambda}+r+\frac{\theta}{1-\lambda} = r\,, \quad (39)\end{aligned}$$

which implies that the estimator in (34) in unbiased. We now compute the variance of $\widehat{r}$ as

$$\begin{aligned}V(\widehat{r}) &= \mathbb{E}\left\{(\widehat{r}-r)^2\right\} \quad (40)\\ &= \mathbb{E}\left\{\widehat{r}^2\right\}-r^2 = \mathbb{E}\left\{(\overline{x}-\overline{y})^2\right\}-r^2 \\ &= \ldots \\ &= \left(\frac{1}{M}+\frac{1}{N}\right)\frac{\theta}{(1-\lambda)^3}\,. \quad (41)\end{aligned}$$

The estimators for the other two parameters of the sGPD, specified in (15), are biased, and we now compute their corresponding biases and variances. Given the estimator forms in (15), we reconsider the estimators in (33) as functions $f(\overline{y}, s_x)$ of the sample central moments $\overline{y}$ and s_x[6]. The bivariate Taylor expansion of $f(\overline{y}, s_x)$ becomes

$$\begin{aligned}f(\overline{y},s_x) &= f(\mu_y,\sigma_x^2)+f_{\overline{y}}(\mu_y,\sigma_x^2)(\overline{y}-\mu_y) \\ &\quad +f_{s_x}(\mu_y,\sigma_x^2)(s_x-\sigma_x^2) \\ &\quad +\frac{1}{2}\left[f_{\overline{y}\overline{y}}(\mu_y,\sigma_x^2)(\overline{y}-\mu_y)^2\right. \quad (42)\\ &\quad +f_{\overline{y}s_x}(\mu_y,\sigma_x^2)(\overline{y}-\mu_y)(s_x-\sigma_x^2) \\ &\quad \left.+f_{s_xs_x}(\mu_y,\sigma_x^2)(s_x-\sigma_x^2)^2\right] \\ &\quad +\text{ higher-order terms}\,,\end{aligned}$$

wherein the partial derivatives are to be evaluated at $\overline{y}=\mu_y$ and $s_x=\sigma_x^2$. We only need to compute the second derivatives and can ignore the higher-order terms in (42) since we want to derive expressions that are accurate to order N^{-1} and M^{-1}.

To obtain terms for the expected value which give an accuracy of (N^{-1}, M^{-1}), we calculate the following expectations:

$$\begin{aligned}\mathbb{E}\{(\overline{y}-\mu_y)\} &= 0, \quad (43)\\ \mathbb{E}\left\{(s_x-\sigma_x^2)\right\} &= -\frac{\sigma_x^2}{N}, \quad (44)\\ \mathbb{E}\left\{(\overline{y}-\mu_y)^2\right\} &= \frac{\sigma_y^2}{M}, \quad (45)\\ \mathbb{E}\left\{(s_x-\sigma_x^2)^2\right\} &= \frac{\mu_{4(X)}-(\sigma_x^2)^2}{N}+\mathcal{O}\left(N^{-2}\right), \quad (46)\end{aligned}$$

and

$$\mathbb{E}\left\{(\overline{y}-\mu_y)(s_x-\sigma_x^2)\right\}=0 \quad (47)$$

since the samples from the GPD and sGPD are independent. Substituting these values into (42) yields

$$\begin{aligned}\mathbb{E}\{f(\overline{y},s_x)\} &= f(\mu_y,\sigma_x^2)-\frac{\sigma_x^2}{N}f_{s_x}(\mu_y,\sigma_x^2) \\ &\quad +\frac{1}{2}\frac{\mu_{4(X)}-(\sigma_x^2)^2}{N}f_{s_xs_x}(\mu_y,\sigma_x^2) \quad (48)\\ &\quad +\frac{1}{2}\frac{\sigma_y^2}{M}f_{\overline{y}\overline{y}}(\mu_y,\sigma_x^2)+\mathcal{O}(N^{-2}).\end{aligned}$$

B. Biases of Moment Estimators

Recall that the bias of an estimator $\widehat{X}$ is defined as

$$b\left(\widehat{X}\right) = \mathbb{E}\left\{\widehat{X}-X\right\} = \mathbb{E}\left\{\widehat{X}\right\}-X\,. \quad (49)$$

[6]For the purposes of this derivation, we use s_x to represent the second sample central moment s_x^2, and not its square root as the notation may suggest.

For both estimators, note that we evaluate the partial derivatives at

$$\overline{y} = \mu_y = \theta(1-\lambda)^{-1} \quad (50)$$

$$\text{and} \quad s_x = \sigma_x^2 = \theta(1-\lambda)^{-3}\ . \quad (51)$$

We first consider the moment estimator $\widehat{\theta}_x$ of the parameter θ which is given by

$$\widehat{\theta}_x = \sqrt{\overline{y}^3/s_x} \quad (52)$$

as the function $f(\overline{y}, s_x)$. For this particular function, we can compute the derivatives:

$$\frac{\partial}{\partial s_x} f(\mu_y, \sigma_x^2) = -\frac{1}{2}(1-\lambda)^3, \quad (53)$$

$$\frac{\partial^2}{\partial {s_x}^2} f(\mu_y, \sigma_x^2) = \frac{3}{4}\frac{(1-\lambda)^6}{\theta}, \quad (54)$$

$$\text{and} \quad \frac{\partial^2}{\partial \overline{y}^2} f(\mu_y, \sigma_x^2) = \frac{3}{4}\frac{(1-\lambda)^2}{\theta}. \quad (55)$$

Furthermore, the fourth central moment for the sGPD is [9]

$$\mu_{4(X)} = 3\theta^2(1-\lambda)^{-6} + \theta(1+8\lambda+6\lambda^2)(1-\lambda)^{-7}, \quad (56)$$

and thus the bias of the estimator of the parameter θ of the sGPD is computed as

$$b\left(\widehat{\theta}_x\right) \simeq \frac{1}{8N}\left[10\theta + \frac{18\lambda^2+24\lambda+3}{1-\lambda}\right] + \frac{1}{8M}\frac{3}{1-\lambda}, \quad (57)$$

which has error terms of the order N^{-2}.

Similarly, we consider the moment estimator $\widehat{\lambda}_x$ which is given by

$$\widehat{\lambda}_x = 1 - \sqrt{\overline{y}/s_x} \quad (58)$$

as the function $g(\overline{y}, s_x)$, and compute the derivatives:

$$\frac{\partial}{\partial s_x} g(\mu_y, \sigma_x^2) = \frac{1}{2}\frac{(1-\lambda)^4}{\theta}, \quad (59)$$

$$\frac{\partial^2}{\partial {s_x}^2} g(\mu_y, \sigma_x^2) = -\frac{3}{4}\frac{(1-\lambda)^7}{\theta^2}, \quad (60)$$

$$\text{and} \quad \frac{\partial^2}{\partial \overline{y}^2} g(\mu_y, \sigma_x^2) = \frac{1}{4}\frac{(1-\lambda)^3}{\theta^2}. \quad (61)$$

We can now derive the bias of the estimator of the parameter λ of the sGPD as

$$b\left(\widehat{\lambda}_x\right) \simeq -\frac{10\theta(1-\lambda)+18\lambda^2+24\lambda+3}{8N\theta} + \frac{1}{8M\theta}. \quad (62)$$

C. Asymptotic Variances of Moment Estimators

Continuing with the function approach, we can compute the variances of the moment estimators via

$$V\{f(\overline{y}, s_x)\} \quad \text{and} \quad V\{f(\overline{y}, s_x)\}\ . \quad (63)$$

To this end, we first rewrite

$$\begin{aligned} V\{f(\overline{y}, s_x)\} &= V\left\{f(\overline{y}, s_x) - f(\mu_y, \sigma_x^2)\right\} \\ &= \mathbb{E}\left\{\left[f(\overline{y}, s_x) - f(\mu_y, \sigma_x^2)\right]^2\right\} \\ &\quad - \left[\mathbb{E}\left\{f(\overline{y}, s_x) - f(\mu_y, \sigma_x^2)\right\}\right]^2, \end{aligned} \quad (64)$$

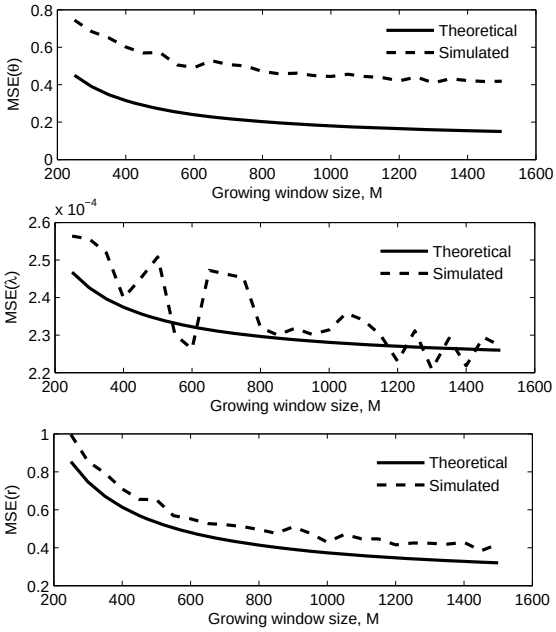

Fig. 14. Comparison of the theoretical and simulated mean-squared errors for the sGPD parameters for the case when $\{\theta = 20, \lambda = 0.5, r = 7\}$.

and then substituting the expression from (42), and simplifying since the second term in the above equation is of order N^{-2} and M^{-2} or lower, we obtain

$$\begin{aligned} V\{f(\overline{y}, s_x)\} &= \frac{\sigma_y^2}{M} f_{\overline{y}}^2(\mu_y, \sigma_x^2) \\ &\quad + \frac{\mu_{4(X)} - (\sigma_x^2)^2}{N} f_{s_x}^2(\mu_y, \sigma_x^2) \end{aligned} \quad (65)$$

which is accurate on the order of N^{-1} and M^{-1}. We first compute the partial derivatives

$$f_{\overline{y}}^2(\mu_y, \sigma_x^2) = \frac{9}{4}(1-\lambda)^2, f_{s_x}^2(\mu_y, \sigma_x^2) = \frac{1}{4}(1-\lambda)^8 \quad (66)$$

$$g_{\overline{y}}^2(\mu_y, \sigma_x^2) = \frac{1}{4}\frac{(1-\lambda)^4}{\theta^2}, g_{s_x}^2(\mu_y, \sigma_x^2) = \frac{1}{4}\frac{(1-\lambda)^8}{\theta^2} \quad (67)$$

and now obtain expressions for the variances of the estimators which are given by

$$V\left(\widehat{\theta}_x\right) \simeq \frac{\theta(1-\lambda)Q(\theta,\lambda)}{4N} + \frac{9\theta}{4M(1-\lambda)}, \quad (68)$$

and

$$V\left(\widehat{\lambda}_x\right) \simeq \frac{(1-\lambda)Q(\theta,\lambda)}{4N\theta} + \frac{1-\lambda}{4M\theta}, \quad (69)$$

where

$$Q(\theta,\lambda) = 2\theta(1-\lambda) + 6\lambda^2 + 8\lambda + 1. \quad (70)$$

D. Analysis of the MSE

Given these expressions for the bias and variance of an estimator, we can compute the MSE of the estimator via

$$\text{MSE} = b^2\left(\widehat{\theta}_x\right) + V\left(\widehat{\theta}_x\right) \quad (71)$$

and we notice that the MSE goes to zero for the estimators of all three sGPD parameters as the number of samples M

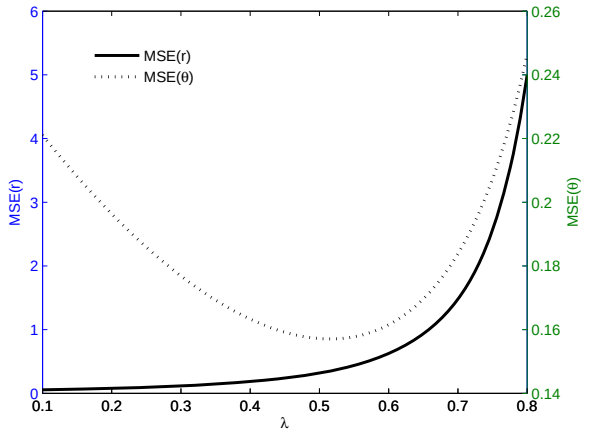

Fig. 15. Effect of λ on the performance of the $\hat{\theta}$ and $\hat{r}$ estimators; parameters for the sGPD are $\{\theta = 20, r = 7\}$ with varying $\lambda \in [0.1, 0.8]$.

and N go to infinity. Thus we conclude that $\widehat{r}$, $\widehat{\theta}_x$ and $\widehat{\lambda}_x$ are *consistent*[7] estimators.

Figure 14 provides a numerical comparison between the mean-squared error (MSE) in the theoretical and simulated cases. We consider the GPD and sGPD distributions with parameters $\{\theta = 20, \lambda = 0.5, r = 7\}$, and an update window of length $N = 750$ for hypothesis H_1. The length of the growing window, for hypothesis H_0, is varied from $M = 250, \ldots, 1500$. The simulated mean-squared error, plotted in Figure 14 for each of the three sGPD parameters, was averaged over 2000 runs. We see that the theoretical expression derived for the estimator mean-squared errors (71), which is accurate on the order of N^{-1}, seems to be a lower bound on the error. The estimators have a low MSE when 1 second of observations are used, and the estimation of λ is very accurate.

The value of θ does not significantly affect the performance of the estimators, but the value of λ does, as evidenced in Figure 15. The estimator $\hat{\theta}$ exhibits a slightly better performance when $\lambda \approx 0.55$, and the error in the estimation of r becomes significant when $\lambda > 0.6$. In this latter case, we can conclude that the joint estimation of the GPD and sGPD parameters will be very poor. For the traces we have considered, we find that $\lambda < 0.5$, and thus the error in estimating r is less than unity.

ACKNOWLEDGMENT

Research has been funded by DHS NBCHC040137 and NSF CNS-0626696. Traces used in this work were provided by the USC/LANDER project.

[7]A sequence of estimators for a parameter θ is said to be *consistent* (or asymptotically consistent) if this sequence converges in probability to θ. In our case, the estimator is a function of the sample sizes M and N. Thus, as M and N tend to infinity, the estimator converges in probability to the true value of the parameter, and the mean-squared error tends to zero [11].