Statistical Traffic Anomaly Detection in Time-Varying Communication Networks *

Jing Wang† and Ioannis Ch. Paschalidis‡

***Abstract*—We propose two methods for traffic anomaly detection in communication networks where properties of normal traffic are time-varying. We formulate the anomaly detection problem as a *binary composite hypothesis testing problem* and develop a *model-free* and a *model-based* method, leveraging techniques from the theory of large deviations. Both methods operate by first extracting a family of *Probability Laws (PLs)* that represent normal traffic patterns during different time-periods, and then detect anomalies by assessing deviations of traffic from these laws. We establish the asymptotic Newman-Pearson optimality of both methods and develop an optimization-based approach for selecting the family of PLs from past traffic data. We validate our methods on networks with two representative time-varying traffic patterns and one common anomaly related to data exfiltration. Simulation results show that our methods perform better than their vanilla counterparts which assume that normal traffic is stationary.**

***Index Terms*—Statistical anomaly detection, large deviations theory, set covering, binary composite hypothesis testing, cyber-security.**

I. Introduction

A network traffic anomaly is, broadly speaking, an unusual traffic pattern that can not be explained by the typical variability observed in communication network traffic. Traffic anomalies may arise either due to operational malfunctions (e.g., router failures) or due to the presence of malicious traffic that threatens the security of the network. Automated online traffic anomaly detection has received a lot of attention, primarily motivated by security considerations.

Network traffic anomaly detection is a special case of more broadly defined system anomaly detection and relevant approaches can be roughly grouped into two classes: *signature-based anomaly detection*, where known patterns of past anomalies are used to identify ongoing anomalies [1, 2], and *change-based anomaly detection* which identifies patterns that substantially deviate from normal patterns of operations [3–6]. Signature-based methods do not suffer from false-alarms, yet, detection rates can be quite low, e.g., below 70% [7]. Furthermore, such methods cannot detect *zero-day attacks*, i.e., attacks not previously seen, and need constant (and expensive) updating to keep up with new attack signatures. In contrast, *change-based anomaly detection* methods are considered to be more economic and promising since they can identify novel attacks. In this work we focus on *change-based anomaly detection* methods, in particular methods that leverage statistical techniques.

Statistical anomaly detection consists of two steps. The first step is to characterize "normal behavior" by analyzing past system behavior. The second step is to identify time instances where system behavior does not appear to be normal by monitoring the system continuously. For anomaly detection in communication networks, [5] presents two methods to characterize normal behavior and to assess deviations from it based on the theory of *Large Deviations* (LD) [8]. Both methods consider the traffic, which is viewed as sequence of flows, as a sample path of an underlying stochastic process and "compare" empirical measures of current network traffic to some reference network traffic model. The first method – called *model-free* – assumes that traffic consists of an independent and identically distributed (i.i.d.) sequence of flows, while the second method – called *model-based* – models traffic as a *Markov Modulated Process*. Both methods make a *stationarity assumption*, postulating that the statistical properties of traffic do not change over time.

However, traffic in modern communication networks is hardly stationary [9]. Internet traffic is subject to weekly and diurnal variations [10, 11]. Internet traffic is also influenced by macroscopic factors such as important holidays and events [12], leading to spikes in the rate of flows that eventually subside after some period of time. Similar phenomena arise in local area networks as well. This motivates the work in this paper that aims at developing methods which can accommodate transient and periodic traffic behavior.

To that end, we present two methods that are robust to time-varying traffic behavior: a *robust model-free* and a *robust model-based* method that can be seen as generalizations of the corresponding methods from [5].

* Research partially supported by the NSF under grants CNS-1239021 and IIS-1237022, by the ARO under grants W911NF-11-1-0227 and W911NF-12-1-0390, and by the ONR under grant N00014-10-1-0952.

† Division of Systems Engineering, Boston University, 8 St. Mary's St., Boston, MA 02215, wangjing@bu.edu.

‡ Department of Electrical and Computer Engineering and Division of Systems Engineering, Boston University, 8 St. Mary's St., Boston, MA 02215, yannisp@bu.edu, http://ionia.bu.edu/.

Robustness, here, should be interpreted in the same vein as in [13], that is, anomaly detection decisions are made from a composite hypothesis test which allows for ambiguity in the probabilistic model that characterizes normal behavior. In particular, a *Probability Law (PL)* used in [5] to characterize normal behavior is now replaced by *family* of PLs where each member of this family captures different "modes" of the traffic at different time intervals (e.g., day, night, during a spike in traffic, etc.). We develop new robust hypothesis tests in this setting and establish their asymptotic optimality in a Neyman-Pearson sense. Second, we propose a two-stage method to estimate a family of PLs. Our two-stage method transforms a hard problem (i.e., estimating PLs for *multi-dimensional* data) into two well-studied problems: (i) estimating *one-dimensional* data parameters, and (ii) formulating the problem of selecting a representative set of PLs to form a PL family as a *set covering* problem. As we will see, this two-stage method is suitable for distributed computation.

The remainder of the paper is structured as follows. Sec. II formulates system anomaly detection as a binary composite hypothesis testing problem and develops our two robust methods. Sec. III discusses how we parametrize network traffic and presents our approach for estimating PL families. Sec. IV validates our overall approach in several realistic settings and compares the performance of our robust methods to their non-robust counterparts. Sec. V contains some concluding remarks.

Notation: Throughout the paper all vectors are assumed to be column vectors. We use lower case boldface letters to denote vectors and for economy of space we write $\mathbf{x} = (x_1, \ldots, x_n)$ for the column vector $\mathbf{x}$. We use upper case boldface letters to denote matrices. We use script letters to define sets, and denote by $|\mathcal{A}|$ the cardinality of set $\mathcal{A}$.

II. Binary composite hypothesis testing

In this section, we present a hypothesis testing framework for making anomaly detection decisions. As we mentioned earlier, the crux of our methodology is to model network traffic as a stochastic process. Historical traffic time-series can be used to estimate a set of parameters for this stochastic process, giving rise to what we called a PL. Then the problem of detecting an anomaly in real-time is equivalent to testing whether current observed traffic is indeed a "likely" sample path of the stochastic process we learned from past history.

The general problem we will consider is testing whether a sequence of observations $\mathcal{G} = \{g^1, \ldots, g^n\}$ is a sample path of a stochastic process $\mathscr{G}$ (hypothesis $\mathcal{H}_0$). The stochastic process $\mathscr{G}$ is assumed to be discrete-time, thus, a sample path of length n can be denoted by $\mathscr{G} = \{G^1, \ldots, G^n\}$. All random variables G^i are discrete and their sample space is a finite alphabet $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_{|\Sigma|}\}$, where $|\Sigma|$ denotes the cardinality of Σ. Realizations of $G^1, \ldots, G^n$ will be denoted by g^i and also take values in Σ. We assume that the joint probability distribution $p_\theta(G^1, \ldots, G^n)$ is parameterized by some parameter $\theta \in \Omega$, where Ω is the set where θ takes values. The parameter θ is considered unknown and $\{p_\theta(G^1, \ldots, G^n) : \forall\ \theta \in \Omega\}$ can be viewed as a family of PLs characterizing $\mathcal{H}_0$. As we will see later, θ will range over the various modes of the traffic at different time intervals and Ω will be a discrete set.

We will treat the problem of deciding whether a realization $\mathcal{G}$ of $\mathscr{G}$ is anomalous as the *binary composite hypothesis testing problem* between $\mathcal{H}_0$ and the complement of $\mathcal{H}_0$ denoted by $\bar{\mathcal{H}_0}$. We call such a test composite because θ is considered unknown. A decision rule $\mathcal{S}$ is a set such that $\mathcal{G} \in \mathcal{S} \triangleq \{\mathcal{G} | \mathcal{H}_0 \text{ is rejected}\}$, indicating an anomaly, and $\mathcal{G} \notin \mathcal{S} \triangleq \{\mathcal{G} \mid \mathcal{H}_0 \text{ is accepted}\}$, indicating no anomaly. For a decision rule $\mathcal{S}$, we define $\alpha^{\mathcal{S}}(\theta) = P_{\theta|\mathcal{H}_0}[\mathcal{G} \in S]$ to be the false alarm rate, and $\beta^{\mathcal{S}}(\theta) = P_{\theta|\bar{\mathcal{H}_0}}[\mathcal{G} \notin S]$ to be the miss detection rate, where $P_{\theta|\mathcal{H}_0}[\cdot]$ is the probability evaluated assuming $\mathcal{H}_0$ is true and $P_{\theta|\bar{\mathcal{H}_0}}[\cdot]$ is the probability evaluated assuming the alternative hypothesis is true.

We use the term *exponent* to refer to the quantity $-\lim_{n\to\infty} \frac{1}{n} \log P[\cdot]$ for some probability $P[\cdot]$, if the limit exists. If the exponent is d, then the probability approaches zero as e^{-nd}. We next present the definition of the *Generalized Neyman-Pearson Criterion* for decision rules.

Definition 1
(Generalized Neyman-Pearson (GNP) Criterion). A decision rule $\mathcal{S}$ is optimal if it satisfies

$$\lim_{n\to\infty} \sup \frac{1}{n} \log \alpha^{\mathcal{S}}(\theta) \leq -\lambda, \tag{1}$$

and maximizes $-\lim_{n\to\infty} \sup \frac{\log \beta^{\mathcal{S}}(\theta)}{n}$ uniformly for all $\theta \in \Omega$.

Because the joint distribution $p_\theta(G^1, \ldots, G^n)$ becomes complex when n is large, we focus on two types of simplification. One is to assume all random variables G^i are i.i.d., the other is to assume the stochastic process $\mathscr{G}$ is a Markov chain.

A. A model-free method

We first propose a method we call *model-free*, where the random variables G^i are i.i.d. Each G^i takes the value σ_j with probability $p_\theta^F(G^i = \sigma_j)$, $j = 1, \ldots, |\Sigma|$, which is parameterized by $\theta \in \Omega$. We refer to the vector $\mathbf{p}_\theta^F = (p_\theta^F(G^i = \sigma_1), \ldots, p_\theta^F(G^i = \sigma_{|\Sigma|}))$ as the *model-free* Probability Law (PL) corresponding to θ. Then the family of *model-free* PLs $\mathcal{P}^F = \left\{\mathbf{p}_\theta^F : \theta \in \Omega\right\}$

characterizes the stochastic process $\mathscr{G}$. To characterize the observation $\mathcal{G}$, let

$$\mathcal{E}_F^{\mathcal{G}}(\sigma_j) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(G^i = \sigma_j), \qquad j = 1, \ldots, |\Sigma|, \quad (2)$$

where $\mathbf{1}(\cdot)$ is an indicator function. Then, an estimate for the underlying *model-free* PL based on the observation $\mathcal{G}$ is $\boldsymbol{\mathcal{E}}_F^{\mathcal{G}} = \left\{\mathcal{E}_F^{\mathcal{G}}(\sigma_j) : \ j = 1, \ldots, |\Sigma|\right\}$, which is called the *model-free* empirical measure of $\mathcal{G}$.

Suppose $\boldsymbol{\mu} = (\mu(\sigma_1), \ldots, \mu(\sigma_{|\Sigma|}))$ is a *model-free* PL and $\boldsymbol{\nu} = (\nu(\sigma_1), \ldots, \nu(\sigma_{|\Sigma|}))$ is a *model-free* empirical measure. To quantify the difference between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$, we define the *model-free divergence* between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ as

$$D_F(\boldsymbol{\nu} \| \boldsymbol{\mu}) \triangleq \sum_{j=1}^{|\Sigma|} \nu(\sigma_j) \log \frac{\nu(\sigma_j)}{\mu(\sigma_j)}. \quad (3)$$

The anomaly detection test is given in the following definition. Notice that the minimization over θ is selecting the PL with the minimal exponent, or equivalently the largest probability, hence, the most likely PL.

Definition 2
(Model-Free Generalized Hoeffding Test). The model-free generalized Hoeffding test *[14] is to reject $\mathcal{H}_0$ when $\mathcal{G}$ is in the set:*

$$S_F^* = \{\mathcal{G} \mid \inf_{\theta \in \Omega} D_F(\boldsymbol{\mathcal{E}}_F^{\mathcal{G}} \| \mathbf{p}_\theta^F) \geq \lambda\},$$

where λ is a detection threshold.

We will be calling $\inf_{\theta\in\Omega} D_F(\boldsymbol{\mathcal{E}}_F^{\mathcal{G}} \| \mathbf{p}_\theta^F)$, the generalized model-free divergence between $\boldsymbol{\mathcal{E}}_F^{\mathcal{G}}$ and $\mathbf{p}_\theta^F$. A similar definition has been proposed for robust localization in sensor networks [15]. We introduce the following theorem.

Theorem II.1 *The model-free generalized Hoeffding test satisfies the GNP criterion.*

Proof: See Appendix A. ∎

We remark that when applying this detection rule in practice we substitute $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ in (3) with $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\nu}}$ where $\hat{\nu}(\sigma_j) = \max(\nu(\sigma_j), \varepsilon)$ and $\hat{\mu}(\sigma_j) = \max(\nu(\sigma_j), \varepsilon), \forall j$ and ε is a small positive constant introduced to avoid underflow and division by zero.

B. A model-based method

We now turn to the *model-based* method where the random process $\mathscr{G} = \{G^1, \ldots, G^n\}$ is assumed to be a Markov chain. Under this assumption, and for $\mathcal{G} = \{g^1, \ldots, g^n\}$, the joint distribution of $\mathscr{G}$ becomes $p_\theta(\mathscr{G} = \mathcal{G}) = p_\theta^B(g^1) \prod_{i=1}^{n-1} p_\theta^B(g^{i+1} \mid g^i)$, where $p_\theta^B(\cdot)$ is the initial distribution and $p_\theta^B(\cdot \mid \cdot)$ is the transition probability; all parametrized by $\theta \in \Omega$.

Let $p_\theta^B(\sigma_i, \sigma_j)$ be the probability of seeing two consecutive states (σ_i, σ_j). We refer to the matrix $\mathbf{P}_\theta^B = \{p_\theta^B(\sigma_i, \sigma_j)\}_{i,j=1}^{|\Sigma|}$ as the *model-based* PL associated with $\theta \in \Omega$. Then, we can interpret $\mathcal{P}^B = \left\{\mathbf{P}_\theta^B : \theta \in \Omega\right\}$ as a family of *model-based* PLs. To characterize the observation $\mathcal{G}$, let for all $i, j = 1, \ldots, |\Sigma|$

$$\mathcal{E}_B^{\mathcal{G}}(\sigma_i, \sigma_j) = \frac{1}{n}\sum_{l=2}^{n} \mathbf{1}(g^{l-1} = \sigma_i, g^l = \sigma_j). \quad (4)$$

We define the *model-based* empirical measure of $\mathcal{G}$ as the matrix $\boldsymbol{\mathcal{E}}_B^{\mathcal{G}} = \{\mathcal{E}_B^{\mathcal{G}}(\sigma_i, \sigma_j)\}_{i,j=1}^{|\Sigma|}$. The transition probability from σ_i to σ_j is simply $\mathcal{E}_B^{\mathcal{G}}(\sigma_j|\sigma_i) = \frac{\mathcal{E}_B^{\mathcal{G}}(\sigma_i,\sigma_j)}{\sum_{j=1}^{|\Sigma|} \mathcal{E}_B^{\mathcal{G}}(\sigma_i,\sigma_j)}$.

Suppose $\boldsymbol{\Pi} = \{\pi(\sigma_i, \sigma_j)\}_{i,j=1}^{|\Sigma|}$ is a *model-based* PL and $\mathbf{Q} = \{q(\sigma_i, \sigma_j)\}_{i,j=1}^{|\Sigma|}$ is a *model-based* empirical measure. Let $\hat{\pi}(\sigma_j|\sigma_i)$ and $\hat{q}(\sigma_j|\sigma_i)$ be the corresponding transition probabilities from σ_i to σ_j. Then, the *model-based divergence* between $\boldsymbol{\Pi}$ and $\mathbf{Q}$ is

$$D_B(\mathbf{Q} \parallel \boldsymbol{\Pi}) = \sum_{i=1}^{|\Sigma|}\sum_{j=1}^{|\Sigma|} q(\sigma_i, \sigma_j) \log \frac{q(\sigma_j|\sigma_i)}{\pi(\sigma_j|\sigma_i)}. \quad (5)$$

Similar to the *model-free* case, we present the following definition:

Definition 3
(Model-Based Generalized Hoeffding Test). The model-based generalized Hoeffding test *is to reject $\mathcal{H}_0$ when $\mathcal{G}$ is in the set:*

$$S_B^* = \{\mathcal{G} \mid \inf_{\theta \in \Omega} D_B(\boldsymbol{\mathcal{E}}_B^{\mathcal{G}} \| \mathbf{P}_\theta^B) \geq \lambda\},$$

where λ is a detection threshold.

As in the model-free case, we will be calling $\inf_{\theta\in\Omega} D_B(\boldsymbol{\mathcal{E}}_B^{\mathcal{G}} \| \mathbf{P}_\theta^B)$ the generalized model-based divergence between $\boldsymbol{\mathcal{E}}_B^{\mathcal{G}}$ and $\mathbf{P}_\theta^B$. GNP optimality can be established in this case, too (cf. Def. 1).

Theorem II.2 *The model-based generalized Hoeffding test satisfies the GNP criterion.*

Proof: See Appendix B. ∎

As in the model-free case, when applying this detection rule in practice we substitute $\hat{q}(\sigma_i, \sigma_j) = \max(q(\sigma_i, \sigma_j), \varepsilon)$ and $\hat{\pi}(\sigma_i, \sigma_j) = \max(\pi(\sigma_i, \sigma_j), \varepsilon)$ in the place of $q(\sigma_i, \sigma_j)$ and $\pi(\sigma_i, \sigma_j)$ in (5), respectively, where ε is some small positive constant introduced to avoid underflow and division by zero.

III. Network anomaly detection

In this section we present our anomaly detection methods whose structure is outlined in Fig. 1. As seen in the figure, first we use reference flows to estimate a family of PLs that characterize the normal operation of

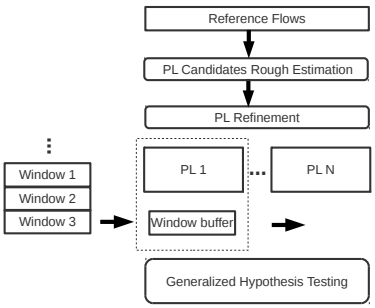

Fig. 1. Structure of the algorithms.

the network. This involves selecting a large candidate set of PLs and then refining it through optimization. Then we test windows of traffic against this family using the *generalized Hoeffding* tests in (2) or (3).

A. Data representation

1) Network traffic representation and flow aggregation: Till now, we deliberately formulated problems in a general way. In fact, the only requirement of our methods on data is that each data record should be comprised of a time stamp and some additional features. As a result, our methods can be used in different applications by choosing appropriate features.

In this paper, however, we focus on *host-based anomaly detection*, a specific application in which we monitor the incoming and outgoing packets of a server. We assume that the server provides only one service (e.g., web server) and other ports are either closed or outside our interests. As a result, we only monitor traffic on a certain port (e.g., port 80 for web service). For servers with multiple ports in need of monitoring, we can run our method on each port.

The features we propose for this particular application relate to a flow representation slightly different from that of commercial vendors like the Cisco NetFlow [16]. Hereafter, we will use the terms "flows", "traffic", and "data" interchangeably. Let $\mathcal{S} = \{\mathbf{s}^1, \ldots, \mathbf{s}^{|\mathcal{S}|}\}$ denote the collection of all packets collected on the monitored port of the host. Since the server IP is always fixed it will be ignored and we will only track the user IP address, denoted by $\mathbf{x}^i$ for packet $\mathbf{s}^i$. The size of $\mathbf{s}^i$ is $b^i \in [0, \infty)$ in bytes and the start time of transmission is $t^i \in [0, \infty)$ in seconds. Using this convention, packet $\mathbf{s}^i$ can be represented as $(\mathbf{x}^i, b^i, t^i_s)$ for all $i = 1, \ldots, |\mathcal{S}|$.

We compile a sequence of packets $\mathbf{s}^1 = (\mathbf{x}^1, b^1, t^1_s), \ldots, \mathbf{s}^m = (\mathbf{x}^m, b^m, t^m_s)$ with $t^1_s < \cdots < t^m_s$ into a *flow* $\mathbf{f} = (\mathbf{x}, b, d_t, t)$ if $\mathbf{x} = \mathbf{x}^1 = \cdots = \mathbf{x}^m$ and $t^i_s - t^{i-1}_s < \delta_F$ for $i = 2, \ldots, m$ and some prescribed $\delta_F \in (0, \infty)$. Here, the *flow size* b is the sum of the sizes of the packets that comprise the flow. The *flow duration* is $d_t = t^m_s - t^1_s$. The *flow transmission time* t equals the start time of the first packet of the flow, i.e., t^1_s. In this way, we can translate the large collection of packets $\mathcal{S}$ into a relatively small collection of flows $\mathcal{F}$.

We first distill the "user space" into something more manageable while enabling us to characterize network behavior of user groups instead of just individual users. For simplicity of notation, we only consider IPv4 address. If $\mathbf{x}^i = (x^i_1, x^i_2, x^i_3, x^i_4) \in \{0, \ldots, 255\}^4$ and $\mathbf{x}^j = (x^j_1, x^j_2, x^j_3, x^j_4) \in \{0, \ldots, 255\}^4$ are two IPv4 addresses, the *distance* between them is defined as $d(\mathbf{x}^i, \mathbf{x}^j) = |x^i_1 - x^j_1|256^3 + |x^i_2 - x^j_2|256^2 + |x^i_3 - x^j_3|256 + |x^i_4 - x^j_4|$. This metric can be easily extended to IPv6 addresses. Suppose $\mathcal{X}$ is the set of unique IP addresses in $\mathcal{F}$. We apply typical K-means clustering on $\mathcal{X}$ with the distance metric defined above. For each $\mathbf{x} \in \mathcal{X}$, we thus obtain a cluster label $k(\mathbf{x})$. Suppose the cluster center for cluster k is $\bar{\mathbf{x}}^k$; then the distance of $\mathbf{x}$ to the corresponding cluster center is $d_a(\mathbf{x}) = d(\mathbf{x}, \bar{\mathbf{x}}^{k(\mathbf{x})})$. The cluster label $k(\mathbf{x})$ and distance to cluster center $d_a(\mathbf{x})$ are used to identify a user IP address $\mathbf{x}$, leading to our final representation of a flow as:

$$\mathbf{f} = (k(\mathbf{x}), d_a(\mathbf{x}), b, d_t, t). \tag{6}$$

2) Quantization: For each $\mathbf{f}^i$, we first quantize $d_a(\mathbf{x}^i)$, b^i, and d^i_t to discrete values. Let $[d^{min}_a, d^{max}_a]$ be the range of $d_a(\mathbf{x}^i)$. We define a discrete alphabet $\Sigma_{d_a} \triangleq \left\{d^{min}_a + (m - \frac{1}{2})(d^{max}_a - d^{min}_a)/|\Sigma_{d_a}|\right\}_{m=1,\ldots,|\Sigma_{d_a}|}$ for $d_a(\mathbf{x}^i)$, where $|\Sigma_{d_a}|$ is the total number of quantization levels for $d_a(\mathbf{x}^i)$. For features b^i and d^i_t, Σ_b and Σ_{d_t} are defined similarly. We then quantize $d_a(\mathbf{x}^i)$, b^i, and d^i_t in $\mathbf{f}^i$ to the closest symbol in the discrete alphabets Σ_{d_a}, Σ_b, and Σ_{d_t}, respectively. Suppose also the total number of user clusters is K. With the proposed quantization, each tuple $(k(\mathbf{x}), d_a(\mathbf{x}), b, d_t)$ corresponds to a symbol in a composite alphabet $\Sigma = \{1, \ldots, K\} \times \Sigma_{d_a} \times \Sigma_b \times \Sigma_{d_t}$. We will denote by $\mathbf{g}^i$ the symbol corresponding to flow $\mathbf{f}^i$. For every (discretized) flow $\mathbf{g}^i$, we will refer to the symbols in $\mathbf{g}^i$ corresponding to $k(\mathbf{x}^i)$, $d_a(\mathbf{x}^i)$, b^i, and d^i_t as features 1, 2, 3, and 4, respectively.

3) Window Aggregation: In our methods, flows in some reference set $\mathcal{F}$ are further aggregated into windows based on their *flow transmission times*. A window is a detection unit that consists of flows in a continuous time range that are to be processed together. Let h be the interval between the start points of two consecutive time windows and w_s be an appropriate window size. Flow $\mathbf{f}^i$ belongs to window j if its transmission time t^i satisfies $t^1 + (j-1)h \le t^i < t^1 + (j-1)h + w_s$. $h < w_s$. $\mathcal{F}_j$ will denote the collection of flows in a window j and we will use $\mathcal{G}_j$ to denote the discretized version of $\mathcal{F}_j$.

B. Estimating PLs from reference traffic

The purpose now is to estimate families of *model-free* and *model-based* PLs $\{\mathbf{p}^F_\theta : \theta \in \Omega\}$ and $\{\mathbf{P}^B_\theta : \theta \in \Omega\}$,

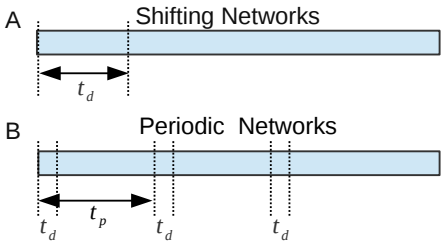

Fig. 2. The relationship between *flow transmission time* t and indices of flows $\mathcal{M}(j)$ governed by PL j for *shifting networks* (A) and *periodic networks* (B).

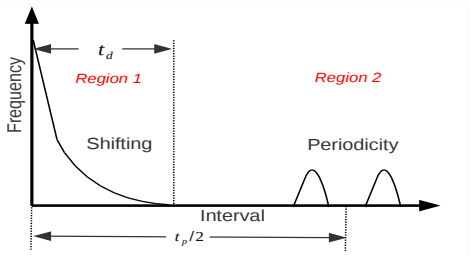

Fig. 3. Histogram of intervals between two consecutive flows with a specific feature quantized to the same discrete value.

respectively, from a discretized collection of flows $\mathcal{G}_{ref}$ with flow transmission times $\mathbf{t} = \{t^1, \ldots, t^n\}$. We will partition $\mathcal{G}_{ref}$ into segments and denote by $\mathcal{M}(j)$ the indices of all flows in segment j. Depending on the time-varying traffic pattern, the partitioning of $\mathcal{G}_{ref}$ will be done in a way such that flows in each $\mathcal{M}(j)$ can be assumed to be generated from the same PL. Thus, each set of flows $\mathcal{M}(j)$ can be seen as a window from which a model-free or model-based empirical measure can be derived using either (2) or (4); this empirical measure is the PL derived from $\mathcal{M}(j)$. Motivated by two representative types of time-varying networks, we consider two approaches of partitioning $\mathcal{G}_{ref}$ into segments.

1) Shifting networks: The first type of networks we consider will be called shifting networks, loosely speaking these are networks where properties of normal traffic slowly "shift" with respect to time. This suggests that the flows close in time are more likely to be governed by the same PL. We divide $\mathcal{G}_{ref}$ into segments, each with a duration of a prescribed value t_d. The flows in each segment are used to estimate a single PL (Fig. 2A). The flow indices in segment j are

$$\mathcal{M}(j) = \{i : t^1 + (j-1)t_d \leq t^i < t^1 + jt_d\}, \quad (7)$$

where $j = 1, \ldots, \lfloor (t^n - t^1)/t_d \rfloor$. t_d characterizes how quickly we expect the statistical properties of the traffic to shift. Larger t_d indicates a slower shifting of traffic properties in the network. One can choose a variety of t_d's, and for each t_d generate the corresponding $\mathcal{M}(j)$ and the resulting PLs.

2) Periodic networks: The second approach is motivated by periodic networks where properties of the normal traffic change periodically. In these networks, two flows can be governed by the same PL either if their transmission times are close or if the difference of their transmission times equals the period (Fig. 2B). Let t_d characterize shifts within the period and let t_p be the period. For $j = 1, \ldots, \lfloor t_p/t_d \rfloor$, let

$$\mathcal{M}(j) = \bigcup_{k \in \mathcal{K}_j} \left\{ i : kt_p + (j-1)t_d \leq t^i < kt_p + jt_d \right\}, \quad (8)$$

where

$$\mathcal{K}_j = \left\{ k : kt_p + (j-1)t_d > t^1 \text{ and } kt_p + jt_d < t^n \right\}.$$

Again, we can choose a variety of t_p's and t_d's with each combination resulting into $\lfloor t_p/t_d \rfloor$ PLs.

Practical networks can exhibit both types of non-stationary behavior described above. Moreover, the periodicity and the degree of shift may change over time, too. To increase the robustness of the set of estimated PLs to these non-stationarities, we first propose a large collection of candidates and then refine it using integer programming.

C. Generating a large collection of PLs

To generate a collection of PLs – one for each window $\mathcal{M}(j)$ – we need to estimate t_d and t_p described in the previous section. We have a reference sequence of quantized flows $\mathcal{G}_{ref} = \{\mathbf{g}^1, \ldots, \mathbf{g}^n\}$ and the corresponding *flow transmission times* $\mathbf{t} = \{t^1, \ldots, t^n\}$. Recall that each quantized flow $\mathbf{g}^i$ consists of quantized values of a cluster label $k(\mathbf{x}^i)$, a distance to cluster center $d_a(\mathbf{x}^i)$, a flow size b^i and a flow duration d_t^i, which are called features $1, \ldots, 4$, respectively. For all $a = 1, 2, 3, 4$, let $\mathcal{G}_a = \{g_a^1, \ldots, g_a^n\}$ be the sequence of quantized feature a for each flow in $\mathcal{G}_{ref}$. For $a = 1, 2, 3, 4$ and $b = 1, \ldots, |\Sigma_a|$, we say a flow $\mathbf{g}^i$ belongs to *channel* a–b if g_a^i equals σ_b^a. We first analyze each channel separately to get a rough estimate of t_d and t_p. Then, channels corresponding to the same feature are aggregated to generate a combined estimate for this feature.

1) Estimation in one channel: We define $\mathcal{I}_{ab} = \{t^i : g_a^i = \sigma_b^a\}$ as the sorted sequence of *flow transmission times* for flows in *channel* a–b. The interval between two consecutive flows in *channel* a–b is $\tau_{ab}^k = t_{ab}^k - t_{ab}^{k-1}, k = 2, \ldots, |\mathcal{I}_{ab}|$, where t_{ab}^k is the kth element in $\mathcal{I}_{ab}$.

For shifting networks, since the majority of flows in each channel belong to a continuous time range, the intervals between two consecutive flows are small. The histogram of the intervals $\{\tau_{ab}^k : k = 2, \ldots, |\mathcal{I}_{ab}|\}$ will have a heavy head but will contain most of the mass within a certain time close to zero (Region 1 in Fig. 3). The end of that time interval that contains most of the mass can be used as an upper bound on the interval between two consecutive flows and is a good option for t_d.

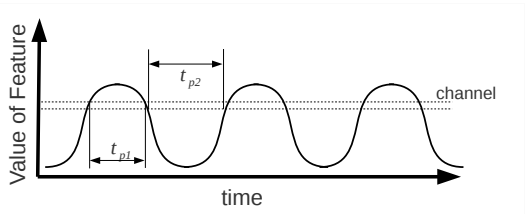

Fig. 4. Illustration of the peaks in Region 2 of Fig. 3.

For periodic networks, the histogram for intervals in $\{\tau_{ab}^k : k = 2, \ldots, |\mathcal{I}_{ab}|\}$ is also heavily skewed to small intervals, thus the t_d can be estimated in the same way as with shifting networks. However, the intervals between two consecutive flows can be large. Fig. 4 shows an example of a feature that exhibits periodicity. There will be two peaks around t_{p1} and t_{p2} in the histogram of intervals for flows whose values are between the two dashed lines. We can select t_p such that $(t_{p1} + t_{p2})/2 \approx t_p/2$. There can be a single or more than two peaks because of the randomness in the network; in either case, we can choose the mean of all peaks as an approximation of $t_p/2$ (cf. Fig. 3).

2) Aggregation for channels of same feature: Denote the estimate of t_d and t_p based on channel a–b as t_d^{ab} and t_p^{ab}, respectively. We use the subscript $\{d, p\}$ to unify the notations for both estimates. $t_p^{ab} = 0$ if no periodicity is found in channel a–b. For $a = 1, 2, 3, 4$, let $\mathcal{T}_{\{d,p\}}^a = \{t_{\{d,p\}}^{ab} : b = 1, \ldots, |\Sigma_a|, \text{ and } t_{\{d,p\}}^{ab} > 0\}$ be the collection of estimates for all channels of feature a. We define the combined estimate of t_d and t_p for feature a as $t_{\{d,p\}}^a = \textsc{Mean}\left(\mathcal{T}_{\{d,p\}}^a\right)$, where $\textsc{Mean}(\cdot)$ calculates the sample mean of a set.

If $\mathcal{T}_p^a$ is empty, the network is non-periodic according to feature a, thus, a family of candidate PLs can be generated using t_d^a and (7). Otherwise, the network is periodic according to feature a, and a family of candidate PLs can be generated using t_d^a, t_p^a, and (8). In addition, in case that some prior knowledge of t_d and t_p is available, the family of candidate PLs can include the PLs calculated based on this prior knowledge.

D. PL refinement with integer programming

From Sec. III-C, we now have a large family of candidates for PLs. The larger this family of PLs is, the more likely it is to overfit $\mathcal{G}_{ref}$. Furthermore, a smaller family of PLs is desirable since it reduces the computational cost of anomaly detection. In this light, this section introduces a method to refine the family of candidate PLs.

For simplicity, we will only describe the procedure for the *model-free* method. The procedure for the *model-based* method is similar. To make the exposition more concise, when we refer to the divergence between the empirical measure of a set of flows and a PL we will simply say the divergence between the set of flows and the PL.

Suppose the family (namely the set) of candidate PLs is $\mathcal{P} = \{\mathbf{p}_1^F, \ldots, \mathbf{p}_N^F\}$ of cardinality N. Because no alarm should be reported for $\mathcal{G}_{ref}$, or segments of $\mathcal{G}_{ref}$, our *primary objective* is to choose the smallest set $\mathcal{P}^F \subseteq \mathcal{P}$ such that there is no alarm for $\mathcal{G}_{ref}$. We aggregate $\mathcal{G}_{ref}$ into M windows as outlined in Sec. III-A and denote the data in window i as $\mathcal{G}_{ref}^i$. Let $D_{ij} = D_F(\boldsymbol{\mathcal{E}}_F^{\mathcal{G}_{ref}^i} \parallel \mathbf{p}_j^F)$ be the divergence between flows in window i and PL j for $i = 1, \ldots, M$ and $j = 1, \ldots, N$. We say window i is covered (namely, reported as normal) by PL j if $D_{ij} \leq \lambda$. Here, λ is the same threshold we used in Def. 2. With this definition, the primary objective becomes to select the minimum number of PLs to cover all the windows.

There may be more than one subsets of $\mathcal{P}$ having the same cardinality and covering all windows. We propose a *secondary objective* characterizing the variation of a set of PLs. Let $\mathcal{N}_j = \{i : \ D_{ij} \leq \lambda\}$ be the index set of windows covered by PL j and denote by $N_j^{(1)}, \ldots, N_j^{(|\mathcal{N}_j|)}$ the ordered elements of $\mathcal{N}_j$. Define $\mathcal{D}_j = \{N_j^{(i)} - N_j^{(i-1)} : \ i = 2, \ldots, |N_j|\}$ the set of differences between consecutive window indices covered by PL j. The *coefficient of variation* for PL j is defined as $c_v^j = \textsc{Std}(\mathcal{D}_j)/\textsc{Mean}(\mathcal{D}_j)$, where $\textsc{Std}(\mathcal{D}_j)$ and $\textsc{Mean}(\mathcal{D}_j)$ are the sample standard deviation and mean of set $\mathcal{D}_j$, respectively. A smaller *coefficient of variation* means that the PL is more "regular." The *secondary objective* is to minimize the sum of *coefficients of variation* for selected PLs. We formulate PL selection as a *weighted set cover problem* in which the weight of PL j is $1 + \gamma c_v^j$, where γ is a small weight for the secondary objective. Let x_i be the 0–1 variable indicating whether PL i is selected or not; let $\mathbf{x} = (x_1, \ldots, x_N)$. Let $\mathbf{A} = \{a_{ij}\}$ be an $M \times N$ matrix whose (i, j)th element a_{ij} is set to 1 if $D_{ij} \leq \lambda$ and to 0 otherwise. Let $\mathbf{c}_v = (c_v^1, \ldots, c_v^N)$. Selecting a set $\mathcal{P}^F (\subseteq \mathcal{P})$ of PLs can be formulated as the following integer programming problem:

$$\begin{aligned} \min \quad & \mathbf{1}'\mathbf{x} + \gamma \mathbf{c}_v'\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{1}, \\ & x_j \in \{0, 1\}, \ j = 1, \ldots, N, \end{aligned} \tag{9}$$

where $\mathbf{1}$ is a vector of ones. The cost function equals a weighted sum of the *primary cost* $\mathbf{1}'\mathbf{x}$ and the *secondary cost* $\mathbf{c}_v'\mathbf{x}$. The first constraint enforces there is no alarm for $\mathcal{G}_{ref}$.

Because (9) is NP-hard, we propose a *heuristic algorithm* to solve it (see the display Algorithm 1). HEURISTICREFINEPL is the main procedure whose parameters are $\mathbf{A}$, $\mathbf{c}_v$, a discount ratio $r < 1$, and a termination threshold γ_{th}. In each iteration, the algorithm de-

function HEURISTICREFINEPL($\mathbf{A}$, $\mathbf{c}_v$, r, γ_{th})
  Init: bestCost := ∞, $\gamma := 1$, $\mathbf{x}^* := 0$
  while $\gamma > \gamma_{th}$ **do**
    $\mathbf{x}$:= GREEDYSOLVE($\mathbf{A}$, γ, $\mathbf{c}_v$), $\gamma := r\gamma$
    if $\mathbf{1}'\mathbf{x} + \gamma_{th}\mathbf{c}_v'\mathbf{x} <$ bestCost **then**
      bestCost := $\mathbf{1}'\mathbf{x} + \gamma_{th}\mathbf{c}_v'\mathbf{x}$
      $\mathbf{x}^* := \mathbf{x}$
    end if
  end while
  return $\mathbf{x}^*$
end function
function GREEDYSETCOVER($\mathbf{A}$, γ, $\mathbf{c}_v$)
  Init: $\mathbf{x}^0 := \mathbf{0}$, $C := \varnothing$
  while $|C| < M$ **do**
    $j^+ := \arg\max_{j:\mathbf{x}[j]=0} \frac{\sum_{i \notin C} a_{ij}}{1+\gamma \mathbf{c}_v[j]}$
    $\mathbf{x}[j^+] := 1$, $C := C \cup \{i : a_{ij^+} = 1\}$
  end while
  return $\mathbf{x}$
end function

Algorithm 1: Greedy algorithm for PL refinement.

creases γ by a ratio r and calls the GREEDYSETCOVER procedure to solve (9). The algorithm terminates when $\gamma < \gamma_{th}$. In the initial iterations, the weight γ for the secondary cost is large so that the algorithm explores solutions which select PLs with less variation. Later, the weight γ decreases to insure that the primary objective plays the main role. Parameters γ_{th} and r determine the algorithm's degree of exploration, which helps avoid local minima. In practice, we can choose small γ_{th} and large r if we have enough computation power.

GREEDYSETCOVER uses the ratio of the number of uncovered windows a PL can cover and the cost $1+\gamma c_v$ as heuristics. GREEDYSETCOVER will add the PL with the maximum heuristic value to $\mathcal{P}^F$ until all windows are covered by the PLs in $\mathcal{P}^F$. Suppose the return value of HEURISTICREFINEPL is $\mathbf{x}^*$. Then, the refined family of PLs is $\mathcal{P}^F = \{\mathbf{p}_j^F : x_j^* > 0, j = 1, \ldots, N\}$.

Once we have a family of PLs, then anomaly detection for either the model-free or the model-based case can be done by using either the test of Definition 2 or Definition 3 and comparing the family of PLs against the empirical measure of windows of current activity (see also Fig. 1). Thus, the proposed method processes windows of current activity one after the other and indicates which windows are found anomalous.

IV. SIMULATION RESULTS

Lacking data with annotated anomalies is a common problem for validation of network anomaly methods. To that end, we have developed an open source software package SADIT [17] to provide flow-level datasets with annotated anomalies. Based on the *fs*-simulator [18], SADIT simulates normal and abnormal flows in networks efficiently.

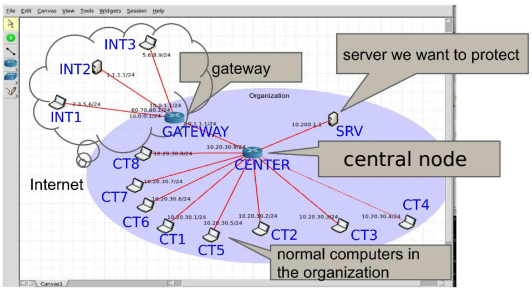

Fig. 5. Simulation setting.

Our simulated network is consists of an internal (to an organization) network and several Internet nodes (Fig. 5). The internal network consists of 8 normal nodes *CT1-CT8* and 1 server *SRV* containing some sensitive information. There are also three Internet nodes *INT1-INT3* that access the internal network through a gateway (*GATEWAY*). For all links, the link capacity is 10 Mb/s and the delay is 0.01 s.

All internal and Internet nodes communicate with the *SRV* and there is no communication between other nodes. The normal flows from all nodes to *SRV* have the same characteristics. The size of the normal flows follows a Gaussian distribution $N(m(t), \sigma^2)$. The arrival process of flows is a Poisson process with arrival rate $\lambda(t)$. Both $m(t)$ and $\lambda(t)$ change with time t.

We consider two representative types of changing patterns for normal traffic: *shifting pattern*, a common pattern for traffic to booming web services [1] (e.g., `www.snapchat.com`), and *day-night pattern*, a common pattern for traffic to services with geographically concentrated users (e.g., `www.boston.com`). For both patterns, we monitor the traffic on the server and evaluate the performance of the robust *model-free* and *model-based* methods for an anomaly associated with attacks in which some hackers exfiltrate sensitive information through SQL injection [19].

A. Shifting pattern

When a web service is booming, users tend to generate and download more content from its servers. From the flow perspective, this means that the average flow size is shifting to higher values. As a simple model, we assume all users exhibit the same shift pattern that is linear with respect to time. We also assume flow arrival rate is stationary. As a result, $m(t)$ is a linear function of time as $m(t) = at + b$, where a and b are two

[1] meaning, web services where interesting content is suddenly posted and many users flock to the site to access, download it, and post more content in response.

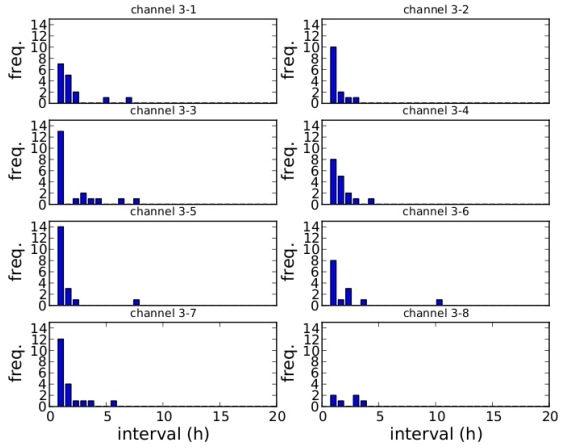

Fig. 6. Histogram of intervals for a network in which flow size exhibits a *shifting pattern*. Each plot corresponds to a channel. There are 30 bins and all plots share the x-axis. For all plots, the first bin is not plotted because it is significantly higher than the rest of the bins.

parameters characterizing the shift of the traffic and $\lambda(t)$ is a constant. In our simulation, we set $b = 4$ Mb, $a = 3.6$ Kb/h, and $\sigma^2 = 0.01$ for all users. The flow arrival rate is constant and $\lambda(t) = 0.1$ fps (flows per second). Using this *shifting pattern*, we generate reference traffic $\mathcal{G}_{ref}$ for one week (168 hours).

To obtain the traffic patterns in $\mathcal{G}_{ref}$, the procedure of Sec. III-C is applied to identify t_d and t_p. For window aggregation, both the window size w_s and the interval h between two consecutive windows is 2000 s. The number of user clusters is $K = 2$. For the quantization, the number of quantization levels for the distance to cluster center, the flow size, and the flow duration (features 2, 3, 4) are 2, 2, and 8, respectively.

The values of t_p and t_d can be estimated by inspecting the histograms of the 8 channels of feature 3 (cf. Fig. 6). Most channels have light tails and no peak caused by periodicity, which clearly indicates that the normal pattern is shifting and non-periodic, thus, t_p is unnecessary. The combined estimate of t_d based on flow size is $t_d^3 = 3.89h$. Our PL generation approach leads to 43 candidate *model-free* and *model-based* PLs, each PL being calculated using a segment of $\mathcal{G}_{ref}$.

For the *model-free* method, because $m(t)$ and $\lambda(t)$ shift with time, the PL calculated based on the flows in a certain window has small divergence with near-by windows, but the divergence becomes larger for windows further away (cf. Fig. 7A). There are 4 PLs selected by the PL refinement procedure when the detection threshold is set to $\lambda = 2$ (Fig. 7B).

We say PL j^* is *active* during window i if its divergence with traffic in this window is the smallest among all selected PLs, namely $j^* = \arg\min_j D_{ij}$. Each selected PL is active for a continuous range of time, which is consistent with the fact that the traffic pattern is

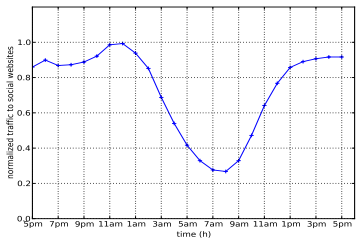

Fig. 11. Traffic pattern of social network websites.

shifting (Fig. 7C). The active PL oscillates between the former active and a new PL before it switches to the new PL. Although the number of PLs is drastically reduced after refinement, the *generalized model-free divergence* between each $\mathcal{G}_{ref}^i$ and the set of selected PLs is very close to the corresponding divergence between $\mathcal{G}_{ref}^i$ and the set of all candidate PLs (Fig. 7D) for each i. This implies that the reduced set represents $\mathcal{G}_{ref}$ well.

For the *model-based* method, 6 PLs are selected by the PL refinement procedure when $\lambda = 2$ (Fig. 8A,B). Each PL is active for a continuous range of time with similar oscillations as in the *model-free* method during the transition between two active PLs (Fig. 8C). Again, the *model-based generalized divergence* between each $\mathcal{G}_{ref}^i$ and either the set of selected PLs or the set of all candidate PLs is very similar (Fig. 8D) for all i.

B. Day-night pattern

The traffic of local web services usually exhibits *diurnal variations* because people browse websites more frequently during the day than during the night. Fig. 11 shows the normalized average traffic to American social websites over a day [20]. We assume that the flow arrival rate and the mean flow size have the same *day-night pattern*. Let $p(t)$ be the function shown in Fig. 11, and assume $\lambda(t) = \Lambda p(t)$ and $m(t) = M_p p(t)$, where Λ and M_p are the peak arrival rate and the peak mean flow size. In our simulation, we set $M_p = 4$ Mb, $\sigma^2 = 0.01$, and $\Lambda = 0.1$ fps for all users. Using this *day-night pattern*, we generate a reference traffic trace $\mathcal{G}_{ref}$ for one week (168 hours) whose start time is 5 pm. Again, an estimation procedure is applied to estimate t_d and t_p. The parameters for window aggregation and quantization are the same as in Sec. IV-A.

The period can be estimated by inspecting the histograms of the 8 channels of feature 3, namely the flow size feature (Fig. 12). Peaks can be observed in all channels except for channel 3-4 and 3-5. The combined estimate of the period based on flow size is $t_p^3 = 24.56$ h, which has only 2.3% error with the real value of 24 h.

For the *model-free* method, there are 64 candidate *model-free* PLs proposed in the estimation stage. The *model-free divergence* between each window and each candidate PL is a periodic function of time, too. Some

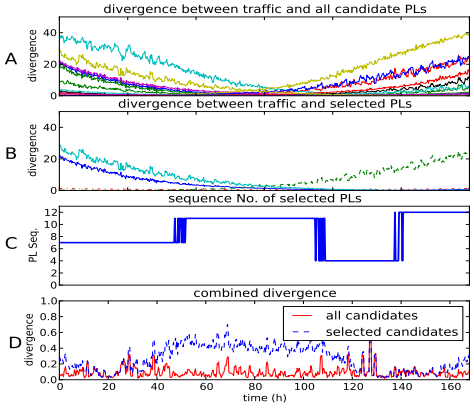

Fig. 7. Results of PL refinement for *model-free* PLs in a network with *shifting pattern*. (A) and (B) plot the *model-free divergence* between $\mathcal{G}^i_{ref}$ and all candidate PLs or just selected PLs, respectively. (C) depicts the active PL at each time. (D) plots the *model-free generalized divergence* between $\mathcal{G}^i_{ref}$ and all candidate PLs or selected PLs. The x-axis corresponds to the start time of the various windows i.

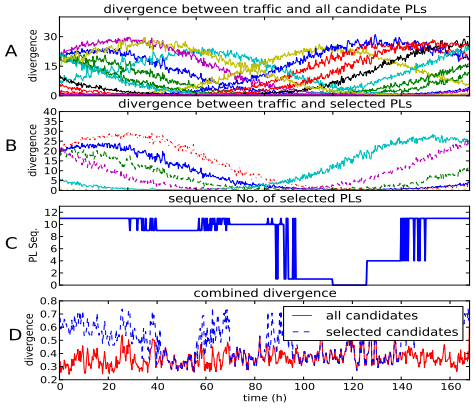

Fig. 8. Results of PL refinement for *model-based* PLs in a network with *shifting pattern*. (A) and (B) plot the *model-based* divergence between $\mathcal{G}^i_{ref}$ and all candidate PLs or just selected PLs, respectively. (C) plots the *active* PL for each window. (D) plots the *model-based generalized divergence* between $\mathcal{G}^i_{ref}$ for all i and all candidate PLs/selected PLs.

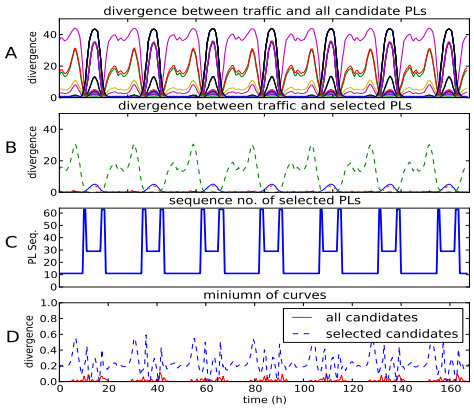

Fig. 9. Results of PL refinement for *the model-free* method in a network with *day-night pattern*. All figures share the x-axis. (A) and (B) plot the divergence of traffic in each window with all candidate PLs and with selected PLs, respectively. (C) shows the *active* PL for each window. (D) plots the *generalized divergence* of traffic in each window with all candidate PLs and selected PLs.

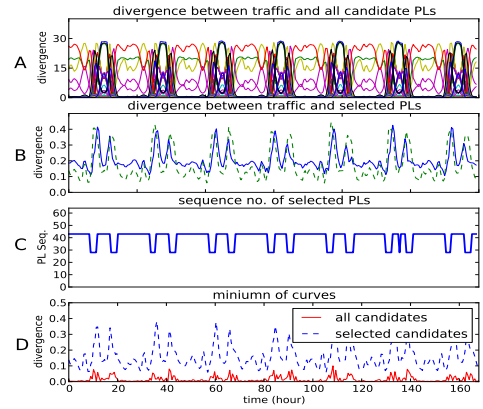

Fig. 10. Results of PL refinement for *the model-based* method in a network with *day-night pattern*. All figures share the x-axis. (A) and (B) plot the divergence of traffic in each window with all candidate PLs and with selected PLs, respectively. (C) shows the *active* PL for each window. (D) plots the *generalized divergence* of traffic in each window with all candidate PLs and selected PLs.

PLs have smaller divergence during the day and some others have smaller divergence during the night (cf. Fig. 9A). However, no PL has small divergence for all windows. 3 PLs out of the 64 candidates are selected when the detection threshold is $\lambda = 0.6$ (cf. Fig. 9B). The 3 selected PLs are active during day, night, and the *transition time* between day and night, respectively (cf. Fig. 9C for the active PLs of all windows). For all windows, the *model-free generalized divergence* between $\mathcal{G}_{ref}$ and all candidate PLs is very close to the divergence between $\mathcal{G}_{ref}$ and only the selected PLs (Fig. 9D). The difference is relatively larger during the *transition time* between day and night. This is because the network behavior is changing fast during this time, thus, more PLs are required to represent the network accurately.

For the *model-based* method, there are 64 candidate *model-based* PLs, too. Similar to the *model-free* method, the *model-based* divergence between all candidate PLs and flows in each window in $\mathcal{G}_{ref}$ is periodic (Fig. 10A) and there is no PL that can represent all the reference data $\mathcal{G}_{ref}$. 2 PLs are selected when $\lambda = 0.4$ (Fig. 10B). One PL is active during the *transition time* and the other is active during the *stationary time*, which consists of both day and night (Fig. 10C). As before, the divergence between each $\mathcal{G}^i_{ref}$ and all candidate PLs is similar to the divergence between $\mathcal{G}^i_{ref}$ and just the selected PLs (Fig. 10D).

The results show that the PL refinement procedure is

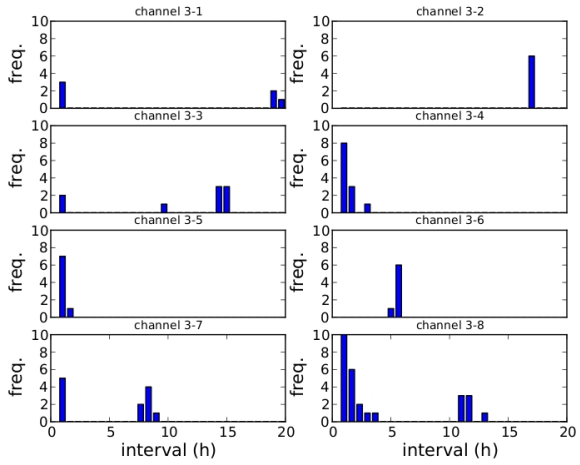

Fig. 12. Histogram of intervals for the *day-night pattern*. Each subfigure corresponds to one channel. All subfigures share the x-axis and the total number of bins is $q = 30$. The first bin is not plotted in the histogram because it is significantly higher than the rest of the bins.

effective and the refined family of PLs is meaningful. Each PL in the refined family of the *model-free* method corresponds to a "pattern of normal behavior," whereas, each PL in the refined family of the *model-based* method describes the transition among the "patterns." This information is useful not only for anomaly detection but also for understanding the normal traffic in dynamic networks.

C. Comparison with vanilla stochastic methods

For both types of normal patterns in Sec. IV-A and Sec. IV-B, we compared the performance of our robust *model-free* and *model-based* method with their vanilla counterparts in detecting anomalies ([5, 21]). In the vanilla methods, all reference traffic $\mathcal{G}_{ref}$ is used to estimate a single PL. We used all methods to monitor the server *SRV* for one week (168 hours) under the two network traffic patterns.

We considered an anomaly in which node $CT2$ increases the mean flow size by 30% at 59h and the increase lasts for 80 minutes before the mean returns to its normal value. This type of anomaly could be associated with a situation when attackers try to exfiltrate sensitive information (e.g., user accounts and passwords) through SQL injection [19].

For all methods, the window size is $w_s = 2,000$ s and the interval $h = 2,000$ s. The quantization parameters are equal to those in the procedure for analyzing the reference traffic $\mathcal{G}_{ref}$. The simulation results show that the robust *model-free* and *model-based* methods perform better than their vanilla counterparts for both types of normal traffic patterns (Fig. 13).

For the case when normal traffic exhibits a *shifting pattern*, the detection threshold λ equals 2.0 for all methods. The vanilla *model-free* method misses the anomaly when the normal traffic shows a shifting pattern (Fig. 13A). Even worse, it generates false alarms for the first 30 hours. In contrast, the robust *model-free* method detects the anomaly successfully without false alarms (Fig. 13B). False alarms also appear in the detection results of the vanilla *model-based* method, though it detects the anomaly successfully (Fig. 13C). Again, the robust *model-based* method reports no false alarm (Fig. 13D).

Compared with the *shifting pattern*, the *day-night pattern* has more influence on the results from the vanilla methods. For both the vanilla and the robust *model-free* methods, the detection threshold λ equals 0.6. The vanilla *model-free* method reports all night traffic (between 3 am to 11 am) as anomalies (Fig. 13E). The reason is that the night traffic is lighter than the day traffic, so the PL calculated using all of $\mathcal{G}_{ref}$ is dominated by the *day pattern*, whereas the *night pattern* is underrepresented. In contrast, because both the *day* and the *night patterns* are represented in the refined family of PLs (Fig. 9B), the robust *model-free* method is not influenced by the fluctuation of normal traffic and successfully detects the anomaly (Fig. 13F).

The *day-night pattern* has similar effects on the *model-based* methods. When the detection threshold λ equals 0.4, the anomaly is barely detectable using the vanilla *model-based* method (Fig. 13G). Similar to the vanilla *model-free* method, the divergence is higher during the *transition time* between day and night because the *transition* pattern is underrepresented in the PL calculated using all of $\mathcal{G}_{ref}$. Again, the robust *model-based* method is superior because both the *transition pattern* and the *stationary pattern* are well represented in the refined family of PLs (Fig. 13H).

V. Conclusions

The statistical properties of normal traffic are time-varying for most actual communication networks. To address limitations of earlier methods that relied on stationarity, we propose a robust *model-free* and a robust *model-based* method suitable for host-based anomaly detection in time-varying networks. Our methods can generate a more complete representation of the normal traffic and are robust to the non-stationarity in networks.

Appendix A
Proof of Theorem II.1

Proof: Denote by $\mathcal{L}_n \triangleq \{\boldsymbol{\nu} \mid \boldsymbol{\nu} = \mathcal{E}^{\mathcal{G}} \text{ for some } \mathcal{G}\}$ the set of all possible *model-free* empirical measures, i.e., types (Def. 2.1.1 of [8]) of sequences with length

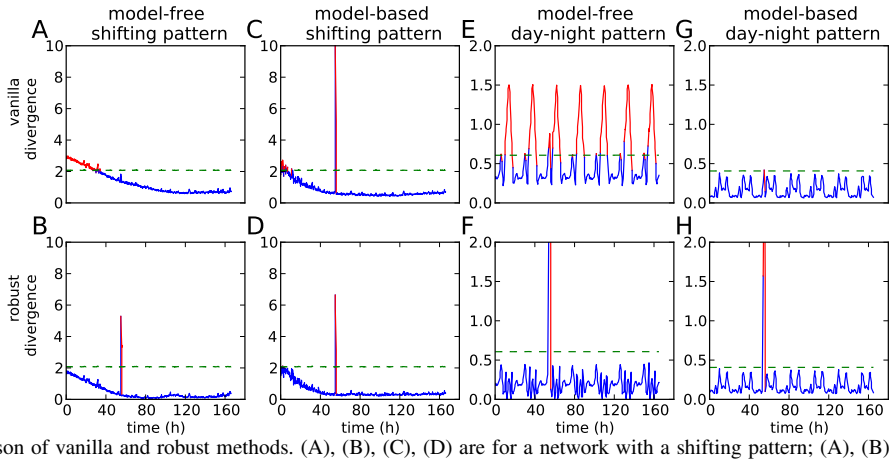

Fig. 13. Comparison of vanilla and robust methods. (A), (B), (C), (D) are for a network with a shifting pattern; (A), (B) show detection results of vanilla and robust *model-free* methods and (C), (D) show detection results of vanilla and robust *model-based* methods. (E), (F), (G), (H) correspond to a network with a day-night pattern; (E) (F) are detection results of vanilla and robust *model-free* methods and (G), (H) show detection results of vanilla and robust *model-based* methods. The horizontal lines indicate the detection threshold.

n. The *type class* of a *model-free* empirical measure $\boldsymbol{\nu}$ is $T_n(\boldsymbol{\nu}) = \{\mathcal{Y} \in \Sigma^n \mid \boldsymbol{\mathcal{E}}_F^{\mathcal{Y}} = \boldsymbol{\nu}\}$, where Σ^n denotes the Cartesian product of Σ with itself n times. Note that a *type class* consists of all permutations of a given observation sequence $\mathcal{Y}$ in this set.

Suppose $\mathrm{P}(\boldsymbol{\nu}|\mathbf{p}_\theta^F)$ is the probability for empirical measure $\boldsymbol{\nu}$ under some PL $\mathbf{p}_\theta^F$ ($\mathcal{H}_0$ is correct). According to Lem. 2.1.9 in [8],

$$(n+1)^{-|\Sigma|}e^{-nD_F(\boldsymbol{\nu}\|\mathbf{p}_\theta^F)} \leq \mathrm{P}(\boldsymbol{\nu}|\mathbf{p}_\theta^F) \leq e^{-nD_F(\boldsymbol{\nu}\|\mathbf{p}_\theta^F)}. \tag{10}$$

For all $\theta \in \Omega$, the false alarm rate of the model-free generalized Hoeffding test is

$$\begin{aligned}
\alpha^{\mathcal{S}_F^*}(\theta) &= P_{\theta|\mathcal{H}_0}\left[\mathcal{G} \in \mathcal{S}_F^*\right] \\
&= \sum_{\{\boldsymbol{\nu}|T_n(\boldsymbol{\nu})\subseteq\mathcal{S}_F^*\}} \mathrm{P}(\boldsymbol{\nu}|\mathbf{p}_\theta^F) \\
&\leq \sum_{\{\boldsymbol{\nu}|T_n(\boldsymbol{\nu})\subseteq\mathcal{S}_F^*\}} e^{-nD_F(\boldsymbol{\nu}\|\mathbf{p}_\theta^F)} \\
&\leq (n+1)^{|\Sigma|}e^{-n\lambda}.
\end{aligned}$$

The first inequality comes from (10). For the second inequality, we use the definition of $\mathcal{S}_F^*$ and the fact that $|\mathcal{L}_n| \leq (n+1)^{|\Sigma|}$ (cf. Lem. 2.1.2 in [8]) . Furthermore,

$$\begin{aligned}
&\lim_{n\to\infty}\sup\frac{1}{n}\log\alpha^{\mathcal{S}_F^*}(\theta) \\
&\quad\leq \lim_{n\to\infty}\sup\frac{1}{n}\log((n+1)^{|\Sigma|}e^{-n\lambda}) \\
&\quad= \lim_{n\to\infty}\sup\frac{1}{n}[|\Sigma|\log(n+1) - n\lambda] \\
&\quad= -\lambda,
\end{aligned}$$

which proves that $\mathcal{S}_F^*$ satisfies (1). Let now $\mathcal{S}$ be some other decision rule satisfying (1). For all $\epsilon > 0$ and large enough n

$$\frac{1}{n}\log\alpha^{\mathcal{S}}(\theta) \leq -\lambda - \epsilon,$$

which is equivalent to

$$\alpha^{\mathcal{S}}(\theta) \leq e^{-n(\lambda+\epsilon)}. \tag{11}$$

In addition, we have

$$\begin{aligned}
\alpha^{\mathcal{S}}(\theta) &= \sum_{\{\boldsymbol{\nu}|T_n(\boldsymbol{\nu})\subseteq\mathcal{S}\}} \mathrm{P}(\boldsymbol{\nu}|\mathbf{p}_\theta^F) \\
&\geq \sum_{\{\boldsymbol{\nu}|T_n(\boldsymbol{\nu})\subseteq\mathcal{S}\}} (n+1)^{-|\Sigma|}e^{-nD_F(\boldsymbol{\nu}\|\mathbf{p}_\theta^F)}.
\end{aligned}$$

The inequality comes from (10). If n is large enough, $(n+1)^{-|\Sigma|} \geq e^{-n\epsilon}$. Moreover, $(n+1)^{-|\Sigma|}e^{-nD_F\left(\boldsymbol{\nu}\|\mathbf{p}_\theta^F\right)} > 0$, $\forall\ \boldsymbol{\nu} \in \mathcal{L}_n$, so

$$\alpha^{\mathcal{S}}(\theta) \geq e^{-n\left(D_F\left(\boldsymbol{\nu}\|\mathbf{p}_\theta^F\right)+\epsilon\right)}.$$

Combined with (11), we obtain

$$e^{-n\left(D_F\left(\boldsymbol{\nu}\|\mathbf{p}_\theta^F\right)+\epsilon\right)} \leq e^{-n(\lambda+\epsilon)},$$

which implies $D_F\left(\boldsymbol{\nu} \parallel \mathbf{p}_\theta^F\right) \geq \lambda$ for all $\boldsymbol{\nu}$ such that $T_n(\boldsymbol{\nu}) \subseteq \mathcal{S}$ and $\theta \in \Omega$. Consequently, $\mathcal{S} \subseteq \mathcal{S}_F^*$ and $\beta^{\mathcal{S}}(\theta) \geq \beta^{\mathcal{S}_F^*}(\theta)$ for all $\theta \in \Omega$, so the *model-free* generalized Hoeffding test satisfies the GNP criterion. ■

Appendix B
Proof of Theorem II.2

Proof: Let $\mathcal{L}_n \triangleq \{\mathbf{Q} : \mathbf{Q} = \boldsymbol{\mathcal{E}}_B^{\mathcal{G}}$, for some $\mathcal{G} \in \Sigma^n\}$ be the set of possible model-based *empirical measures*, i.e., types of the sample paths of Markov chains with length n. Note that every component of $\mathbf{Q} \in \mathcal{L}_n$ belongs to the set $\{\frac{0}{n}, \frac{1}{n}, \ldots, \frac{n}{n}\}$, whose cardinality is $n+1$. $\mathbf{Q}$ is specified by at most $|\Sigma|^2$ such quantities, which means $|\mathcal{L}_n| \leq (n+1)^{|\Sigma|^2}$. Suppose $\mathrm{P}(\mathbf{Q}|\mathbf{P}_\theta^B)$ is

the probability for observation $\mathbf{Q}$ when under PL $\mathbf{P}_\theta^B$ ($\mathcal{H}_0$ is correct). According to Lemma 3 in [22],

$$(n+1)^{-|\Sigma|^2-|\Sigma|}e^{-nD_B(\mathbf{Q}\|\mathbf{P}_\theta^B)} \le \mathrm{P}(\mathbf{Q}|\mathbf{P}_\theta^B) \le e^{-nD_B(\mathbf{Q}\|\mathbf{P}_\theta^B)}. \quad (12)$$

Similar to the *model-free* case, define the *type class* of a *model-based* empirical measure $\mathbf{Q}$ to be $T_n(\mathbf{Q}) = \{\mathcal{Y} \in \Sigma^n \mid \boldsymbol{\mathcal{E}}_B^{\mathcal{Y}} = \mathbf{Q}\}$. For the false alarm rate of the *model-based* generalized Hoeffding test we have

$$\begin{aligned}\alpha^{\mathcal{S}_B^*}(\theta) &= P_\theta[\mathcal{G} \in \mathcal{S}_B^*] \\ &= \sum_{\{\mathbf{Q}|T_n(\mathbf{Q})\subseteq\mathcal{S}_B^*\}} \mathrm{P}(\mathbf{Q}|\mathbf{P}_\theta^B) \\ &\le \sum_{\{\mathbf{Q}|T_n(\mathbf{Q})\subseteq\mathcal{S}_B^*\}} e^{-nD_B(\mathbf{Q}\|\mathbf{P}_\theta^B)} \\ &\le (n+1)^{|\Sigma|^2}e^{-n\lambda},\end{aligned}$$

where the first inequality comes from (12), and the second one is because $|\mathcal{L}_n|$ is bounded by $(n+1)^{|\Sigma|^2}$. Then,

$$\begin{aligned}&\lim_{n\to\infty}\sup\frac{1}{n}\log\alpha^{\mathcal{S}_B^*}(\theta) \\ &\le \lim_{n\to\infty}\sup\frac{1}{n}\log((n+1)^{|\Sigma|^2}e^{-n\lambda}) \\ &= \lim_{n\to\infty}\sup\frac{1}{n}[|\Sigma|^2\log(n+1)-n\lambda] \\ &= -\lambda.\end{aligned}$$

So the *model-based* generalized Hoeffding test satisfies (1). Let now $\mathcal{S}$ be some other decision rule which satisfies (1), for all $\epsilon > 0$ and large enough n. Then,

$$\alpha^{\mathcal{S}}(\boldsymbol{\theta}) \le e^{-n(\lambda+\epsilon)}. \quad (13)$$

Also,

$$\begin{aligned}\alpha^{\mathcal{S}}(\theta) &= \sum_{\{\mathbf{Q}|T_n(\mathbf{Q})\subseteq S\}} \mathrm{P}(\mathbf{Q}|\mathbf{P}_\theta^B) \\ &\ge \sum_{\{\mathbf{Q}|T_n(\mathbf{Q})\subseteq S\}} (n+1)^{-(|\Sigma|^2+|\Sigma|)}e^{-nD_B(\mathbf{Q}\|\mathbf{P}_\theta^B)}.\end{aligned}$$

The inequality comes from (12). If n is large enough, $(n+1)^{-(|\Sigma|^2+|\Sigma|)} \ge e^{-n\epsilon}$, which implies

$$\alpha^{\mathcal{S}}(\theta) \quad \ge \quad e^{-n(D_B(\mathbf{Q}\|\mathbf{P}_\theta^B)+\epsilon)}$$

for all $\mathbf{Q}$ such that $T_n(\mathbf{Q}) \in \mathcal{S}$ and for $\theta \in \Omega$. Combining the above with (13), we obtain

$$e^{-n(D_B(\mathbf{Q}\|\mathbf{P}_\theta^B)+\epsilon)} \le e^{-n(\lambda+\epsilon)},$$

and $D_B(\mathbf{Q} \parallel \mathbf{P}_\theta^B) \ge \lambda$ for all $\mathbf{Q}$ s.t. $T_n(\mathbf{Q}) \in \mathcal{S}$ and $\theta \in \Omega$. Consequently, $\mathcal{S} \subseteq \mathcal{S}_B^*$ and $\beta^{\mathcal{S}}(\theta) \ge \beta^{\mathcal{S}_B^*}(\theta)$, so the *model-based* generalized Hoeffding test satisfies the GNP criterion. ■

REFERENCES

[1] M. Roesch *et al.*, "Snort-lightweight intrusion detection for networks," in *Proceedings of the 13th USENIX conference on System administration*. Seattle, Washington, 1999, pp. 229–238.

[2] V. Paxson, "Bro: a system for detecting network intruders in real-time," *Computer networks*, vol. 31, no. 23, pp. 2435–2463, 1999.

[3] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. ACM, 2002, pp. 71–82.

[4] W. Lu and A. A. Ghorbani, "Network Anomaly Detection Based on Wavelet Analysis," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, no. 1, p. 837601, 2009.

[5] I. C. Paschalidis and G. Smaragdakis, "Spatio-temporal network anomaly detection by assessing deviations of empirical measures," *IEEE/ACM Trans. Networking*, vol. 17, no. 3, pp. 685–697, 2009.

[6] I. C. Paschalidis and Y. Chen, "Statistical anomaly detection with sensor networks," *ACM Trans. Sensor Networks*, vol. 7, no. 3, pp. 17:1–17:23, 2010.

[7] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham *et al.*, "Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation," in *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings*, vol. 2. IEEE, 2000, pp. 12–26.

[8] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*, 2nd ed. NY: Springer-Verlag, 1998.

[9] N. Leavitt, "Network-usage changes push internet traffic to the edge," *Computer*, pp. 13–15, 2010.

[10] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics," *Network, IEEE*, vol. 11, no. 6, pp. 10–23, 1997.

[11] A. King, B. Huffaker, A. Dainotti, and K. C. Claffy, "A coordinated view of the temporal evolution of large-scale Internet events," *Computing*, pp. 53–65, Jan. 2013.

[12] I. Sandvine, "Global Internet phenomena report," https://www.sandvine.com/downloads/general/global-internet-phenomena/2013/sandvine-global-internet-phenomena-report-1h-2013.pdf, 2013.

[13] P. J. Huber, "A robust version of the probability ratio test," *The Annals of Mathematicsl Statistics*, vol. 36, no. 6, pp. 1753–1758, 1965.

[14] W. Hoeffding, "Asymptotically optimal tests for multinomial distributions," *Ann. Math. Statist.*, vol. 36, pp. 369–401, 1965.

[15] I. C. Paschalidis and D. Guo, "Robust and distributed stochastic localization in sensor networks: Theory and experimental results," *ACM Transactions on Sensor Networks*, vol. 5, no. 4, 2009.

[16] Cisco System, "Cisco netflow," http://en.wikipedia.org/wiki/NetFlow, 2012.

[17] J. Wang, "SADIT: Systematic Anomaly Detection of Internet Traffic," http://people.bu.edu/wangjing/open-source/sadit/html/index.html, 2012.

[18] J. Sommers, R. Bowden, B. Eriksson, P. Barford, M. Roughan, and N. Duffield, "Efficient network-wide flow record generation," pp. 2363–2371, 2011.

[19] M. Stampar, "Data Retrieval over DNS in SQL Injection Attacks," *arXiv preprint arXiv:1303.3047*, 2013. [Online]. Available: http://arxiv.org/abs/1303.3047

[20] A. Technologies, "The Net Usage Index by Industry," http://www.akamai.com/html/technology/nui/industry/index.html, 2013.

[21] R. Locke, J. Wang, and I. Paschalidis, "Anomaly detection techniques for data exfiltration attempts," Center for Information & Systems Engineering, Boston University, 8 Saint Mary's Street, Brookline, MA, Tech. Rep. 2012-JA-0001, June 2012.

[22] I. Csiszar, T. M. Cover, and B.-S. Choi, "Conditional limit theorems under Markov conditioning," *IEEE Transactions on Information Theory*, vol. 33, no. 6, pp. 788–801, 1987.