On Tuning the Knobs of Distribution-based Methods for Detecting VoIP Covert Channels

Chrisil Arackaparambil
Dept. of Computer Science
Dartmouth College
cja@cs.dartmouth.edu

Guanhua Yan
Information Sciences
Los Alamos National Lab
ghyan@lanl.gov

Sergey Bratus
Dept. of Computer Science
Dartmouth College
sergey@cs.dartmouth.edu

Alper Caglayan
Milcord LLC
acaglayan@milcord.com

***Abstract*—We study the parameters (knobs) of distribution-based anomaly detection methods, and how their tuning affects the quality of detection. Specifically, we analyze the popular entropy-based anomaly detection in detecting covert channels in Voice over IP (VoIP) traffic.**

There has been little effort in prior research to rigorously analyze how the knobs of anomaly detection methodology should be tuned. Such analysis is, however, critical before such methods can be deployed by a practitioner. We develop a probabilistic model to explain the effects of the tuning of the knobs on the rate of false positives and false negatives. We then study the observations produced by our model analytically as well as empirically. We examine the knobs of window length and detection threshold. Our results show how the knobs should be set for achieving high rate of detection, while maintaining a low rate of false positives. We also show how the throughput of the covert channel (the magnitude of the anomaly) affects the rate of detection, thereby allowing a practitioner to be aware of the capabilities of the methodology.

I. Introduction

Distribution-based anomaly detection methods aim to detect anomalies in massive rate data streams like network traffic streams, update streams in social networks etc. They do so by monitoring the distributions (or histograms) of elements in the stream under consideration. The distributions are monitored for change by employing metrics of the distribution that track various aspects of the stream distribution. Volume, entropy, frequency moments, and KL-divergence are some metrics that have been used in this regard in prior work [17], [11], [7], [20]. A significant deviation in the value of a metric is flagged as an anomaly. Distribution-based anomaly detection has been shown to be useful in detecting port scanning, denial-of-service attacks, and flash crowds in network traffic [17], [5], and in detecting activity of bots, outages, and flash events in information networks [6].

Exfiltration of data via covert channels is an important threat faced by organizations like the military that deal with classified or sensitive information. Voice over IP (VoIP) is increasingly being adopted for communication in the networks of these organizations due to the benefits of the technology. These organizations have, however, expressed concern over the data confidentiality risks posed by covert channels piggybacking on VoIP sessions. These concerns are well-founded, as we discuss in Section III. Motivated by these concerns, we show the effectiveness of distribution-based anomaly detection methods for the detection of covert channels over VoIP. Intuitively, information exfiltrated over a covert channel *is* an anomaly for distributions associated with the abused carrier channel. While designed for VoIP, we believe our methodology is applicable to a much broader class of covert channels.

In our presentations and discussions with researchers on distribution-based anomaly detection methods, we have frequently had questions asked about the various parameters (we will refer to them as *knobs*) involved. We found that how those knobs should be tuned to achieve a good quality of detection is an open problem in general.

One of these knobs is the choice of the length of the window in which the distribution is measured. Distribution-based anomaly detection methods typically slice the observed stream into windows of equal length, either in terms of time or in terms of number of elements. The metric values of the distributions in successive windows is then monitored as a timeseries. Another knob is the detection threshold beyond which the metric value is flagged as an anomaly.

Many intrusion detection and prevention systems (IDS/IPS) offer such tunable knobs for user configuration. It is common knowledge among practitioners that large amounts of effort are required to tune these knobs for optimal performance. Despite the tuning of knobs being a complex and continuous task, a rigorous study of the effects involved has not as yet been undertaken. Such a study is also useful for estimating the effectiveness of effort invested in tuning.

In this work we construct a probabilistic model to observe the effects of the knobs, and then apply the model analytically and empirically by experiments on our VoIP testbed using the covert channels we developed. The primary contribution of this work is to show how the tuning of the knobs affect the quality of anomaly detection. To do the same, we analyze the results of a distribution-based method on a VoIP covert channel while varying the settings of the knobs. We design a simple VoIP covert channel to test the methodology. Our results serve to provide principles for tuning the knobs to practitioners deploying the methodology in real-world applications, and for systematically testing the limits of useful tunability. The latter is an important issue for many practitioners: whereas it is commonly assumed that tuning the parameters of an IDS/IPS can improve its performance on a given network, the limits of such improvements are not clear, and reaching them is something of an arcane art.

Covert channels give us a good handle to vary the magnitude of the anomaly by simply changing the throughput of the covert channel. This ability is perfect for studying how the magnitude of the anomaly affects detection. The Department

978-0-7695-4525-7/12 $26.00 © 2012 IEEE
DOI 10.1109/HICSS.2012.456

IEEE computer society

of Defense Orange Book [1] of security guidelines set the direction here by prescribing a threshold bandwidth for covert channels that should not be allowed on a secure system. It is widely accepted that it is not possible to eliminate all forms of covert channels, particularly those of a low throughput. Instead, the practitioner should aim at eliminating covert channels of throughputs that are sufficiently high so as to pose a real threat. In this work we study how the detection rate (and the false-alarm rate) vary as a function of the covert channel throughput, allowing the implementer to be aware of the capabilities and limitations of distribution-based methodology.

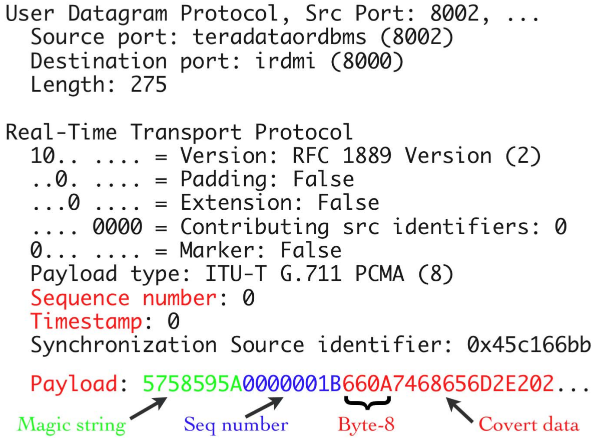
The contributions of this work are summarized thus:

1) A rigorous analysis of the effects of tuning the knobs of distribution-based anomaly detection on detection quality. Construction of a probabilistic model to observe such effects, and analytical and empirical applications of the model.
2) A recipe for a practictioner to follow, to tune the knobs optimally in his specific environment.
3) Our analysis yields an understanding of the subtle effects that come into play when knobs are tuned, and which have a resulting impact on detection quality.
4) Our work considers several different distributions and anomaly sizes (by varying the covert channel rate). It is thus generally applicable to all distribution-based anomaly detection methods.
5) Analysis of the tradeoff between the magnitude of the anomaly and the quality of detection, demonstrating anomalies of what sizes are feasible to detect robustly.
6) Design of a VoIP covert channel, and demonstration of the use of distribution-based anomaly detection to detect such a channel.

The rest of the paper is organized as follows. We present related work next. Then, we discuss VoIP and our sample VoIP covert channel in Section III. Section V presents our probabilistic model, and analytical application of the same. In Section VI we present our empirical evaluations of the model.

II. Related Work

Work on selecting parameters for testing time series tends to be domain specific. In stock market forecasting [9], minor variations in parameter settings have a big effect on the performance, and it was noted that there is little explicit guidance from theory regarding the in-sample window used in the forecast model. In detecting shilling attacks on recommendation systems [21], an optimal window size is derived when the number of attack profiles is known, and a heuristic to estimate the number of attack profiles and adaptively adjust the window size is proposed. In medical diagnosis [14], entropy analysis has been shown to be useful in detecting Alzheimer's disease with window length determined based on spectral analysis. In network anomaly detection, it has been shown [19] that a variable-length window is a more effective strategy for sendmail anomalies based on monitoring conditional entropy. Here, we study the effects of parameter selection specifically for the detection of covert channels in VoIP traffic using distribution-based methods. Due to the nature of our analyses, however, our recommendations tend to be generally applicable to all distribution-based anomaly detection methods.



```
User Datagram Protocol, Src Port: 8002, ...
  Source port: teradataordbms (8002)
  Destination port: irdmi (8000)
  Length: 275

Real-Time Transport Protocol
  10.. .... = Version: RFC 1889 Version (2)
  ..0. .... = Padding: False
  ...0 .... = Extension: False
  .... 0000 = Contributing src identifiers: 0
  0... .... = Marker: False
  Payload type: ITU-T G.711 PCMA (8)
  Sequence number: 0
  Timestamp: 0
  Synchronization Source identifier: 0x45c166bb

  Payload: 5758595A0000001B660A7468656D2E202...
```

Fig. 1. Tshark output illustrating a CCSeq tampered packet with RTP timestamp and sequence number set to 0. [Best viewed in color.]

The survey of Giani et al. [12] provides a taxonomy of covert channel designs. We consider only storage covert channels that use packet header fields for detection in this work. This class of channels is known to be practical for use, because of the high bandwidth provided by the channels. However, Huang et al. [16] present a VoIP covert channel using steganography that claims to have a high capacity. Methods for detecting covert channels include those based on entropy [13] (but these are specifically only applicable to timing-based channels) and n-gram statistical analysis [10].

III. VoIP Covert Channels

In this section we discuss our choice of the sample covert channel design. This choice is motivated by our perception of the relative prominence of both the protocol abuse tricks and the related network technologies to be used. A better motivation would be grounded in statistics of actual illicit use of covert channels in enterprise networks; unfortunately, such statistics are not widely available. Hence, we must base our choice on general architectural and practical considerations of network monitoring and administration.

A. Why VoIP?

VoIP is an ideal target for establishing resilient covert channels across an organization's network boundary by a malicious insider. VoIP functionality is critical to modern enterprise. VoIP-based phone and voice mail systems offer numerous business advantages (maintenance costs not least) over traditional "Plain Old Telephone System" private phone exchanges in organizations, and are replacing PBXs across the board. Thus, it is reasonable to expect VoIP functionality to be deeply integrated into the organization's network fabric—with multiple security exceptions required to support it.

Accordingly, it is a safe bet that the organizations' network security policies and controls such as firewalls and intrusion prevention systems (IPS) must accommodate VoIP protocols, possibly in multiple implementation flavors. However, any single VoIP flavor itself involves multiple interrelated protocols with complex endpoint state. This complexity is hard to model on network appliances that are necessarily limited in the amount of context they can effectively mediate.

In particular, VoIP protocols are notoriously complex and demanding on firewall rules—for example, VoIP signaling and media use different protocols such as SIP and RTP, which require dedicated open port ranges to operate, and are hard to narrow down. Moreover, Network Address Translation (NAT), arguably the most effective way to isolate internal IP LANs, is well-known to cause problems for SIP-based media session setup, and therefore requires workarounds which in turn raise network configuration complexity and undermine firewall isolation of internal systems. Furthermore, VoIP media protocols such as RTP place considerable performance requirements on network elements and paths they cross. As a result, various forms of deep packet inspection and other in-line IPS functionality is impeded by *both* the latency and bandwidth requirements of these protocols to deliver good voice audio quality. The combination of these circumstances makes abusing VoIP protocols an effective way for a malicious insider to punch unnoticed through network boundary defences such as strict firewall rules and traffic auditing measures.

B. VoIP signaling and media protocol essentials

The basic design of VoIP involves separate protocols for signaling (e.g., SIP or Cisco's SCCP) to establish a voice medium session and for the actual transfer of encoded voice payload (e.g., using RTP). In particular, the signaling session facilitates the direct or proxied connection pathway between the VoIP endpoint devices such as handsets.

This separation is necessitated both by architectural concerns such as integration of the session creation with the organization's employee address directories and by performance considerations for established sessions (ideally, established directly between the communicating VoIP endpoints).

C. Basic VoIP convert channel design considerations

Having the firewalls' cooperation in passing payloads outside the organization is only one step towards successful data exfiltration. The exfiltrated data must also be reasonably well-hidden from network auditors, *and* exfiltrated at a reasonable rate to reduce the time of the illicit sessions' duration. Steganographic channels (e.g., SteganRTP [2]) must reduce bandwidth to satisfy classic concepts of steganographic stealth, which assume that any given session is likely to be scrutinized for illicit content; this is far from the everyday reality of network monitoring, in which even obviously anomalous sessions slip through daily, due to large volumes of traffic and the costs of its processing. Accordingly, the design of a covert channel need not be steganographic at the bandwidth's detriment. Indeed, most publicly available covert channel implementations do not pursue steganographic sophistication [4], [3].

However, when designing a VoIP covert channel, one must account for the possibility of a human auditor getting to listen in on an arbitrary segment of the call, and modern network monitoring appliances provide such a capability to operators. Failing such a test or otherwise attracting operator attention with loggable errors would be imprudent for an insider attempting to exfiltrate information.

Accordingly, we formulate the following observations regarding VoIP covert channel design: a covert channel must be resilient to network cross-site call paths common in production (thus favoring storage channels over timing-based channels [12]); a covert channel should not create network error conditions other than those commonly present; and a covert channel crossing a VoIP system should be inaudible to auditors, and to users making calls. Together, these considerations motivate our choice of a simple covert channel.

D. Sample VoIP covert channel

Our covert channel *CCseq* uses the RTP protocol used by VoIP for the voice medium. CCseq's design, while simple, is representative of popular non-steganographic covert channels.

The CCseq "protocol" abuses the real-time nature of RTP voice streams to inject data chunks comparable in size to actual packet payloads, in place of the actual payloads. To keep these payloads from being interpreted as voice data (and creating loud audible noises and/or packet parsing errors), the RTP *sequence number* and/or *timestamp* in such packets is changed to appear outside of the current stream's window (see Figure 1). A number of softphones and handsets we tested silently discard such packets, assuming them to be delayed by the network (and delivered late, out of order).

The channel, despite its design naivete, is transparent to human listeners and commands much higher bandwidth than, say, SteganRTP, which is limited to just using the lower bit of each voice payload byte (under the simplest G.711 codec).

For the protocol's intended recipient on the remote end of the VoIP path (from the organization's border outward), the first 4 bytes of RTP payload contain a fixed magic string for easy identification, the next 4 bytes contain a sequence number to detect loss of injected packets, and the rest of the bytes are covert data (example in Figure 1).

IV. Distribution-based anomaly detection

We now introduce distribution-based anomaly detection. Distribution-based anomaly detection methods typically take in as input a stream of data. For instance, in network traffic the stream of packets might be the input. Usually, if this stream consists of complex multidimensional values, then one dimension may be selected for consideration as a *feature*. In network traffic, the source IP address is an example of a feature. We can assume that the feature under consideration may take values from a universe $[n] = \{1, 2, \ldots, n\}$, for certain value of n. Then we can consider the distribution (histogram) of the feature values in the stream. Distribution-based methods are a class of methods that monitor several aspects of such distributions for significant changes that indicate anomalous activity.

Since the distribution itself is a multidimensional value, a metric is usually used to capture some aspect of the distribution. Volume, entropy, frequency moments, and KL-divergence

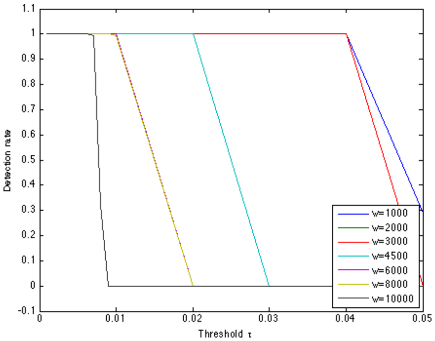


Fig. 4. Detection rate for various windows. Feature: byte-0; t=10. [Best viewed in color.]

have all been used previously as metrics for comparing distributions in this manner [17], [11], [7], [20]. Usually the stream is monitored by slicing it into consecutive windows of equal length, either in terms of time or number of elements, and then computing and comparing the metric values in each window. Since each metric captures only certain characteristics of the distribution, several metric and feature combinations may be monitored together. The results may then be combined using a classifier for a more complete strategy.

Distribution-based anomaly detection is useful to network operators because they are challenged by the high data rates observed in their networks, and are unable to manually comb through the stream for anomalies. Distribution-based methods offer a summarized view of the stream for easy comprehension of network state. Computation of the metric over high rate streams is a challenge. Data stream algorithms offer computational efficiency in this regard, and have been applied previously to enable efficient monitoring [18], [5]. We further discuss them and their use in our setting in Section VII.

V. Probabilistic Modeling of Detection Quality

In this section we present our probabilistic model to predict detection quality terms of window length, covert channel rate, and detection threshold. We apply the model analytically.

Notation: We call a packet (and the corresponding stream element) that is part of the covert channel as a *tainted packet* (*tainted element*), and a packet (element) that belongs to the normal VoIP stream as a *regular packet* (*regular element*). Let ℓ and t denote the number of tainted packets, and the interval (in terms of number of regular packets) between successive tainted packets, respectively. Further, let w denote the length of the window over which the detection method is applied, and τ denote the threshold used in the method to determine when the metric value has deviated far enough for the window to be flagged as an anomaly. We assume that the normal VoIP stream elements are produced by sampling from some underlying distribution $\mathcal{D}$ over the universe $[n]$ of possible values. For instance, when the RTP sequence numbers are considered as the stream for monitoring $n = 2^{32}$. We will denote a window W of elements sampled from the distribution $\mathcal{D}$ as $W \sim \mathcal{D}$. Let M denote the metric under consideration in the detection method, so that $M(W)$ denotes its value for the distribution induced by elements in window W. We will denote a window W^C containing tainted elements among regular elements that are sampled from underlying distribution $\mathcal{D}^C$ as $W^C \sim \mathcal{D}^C$.

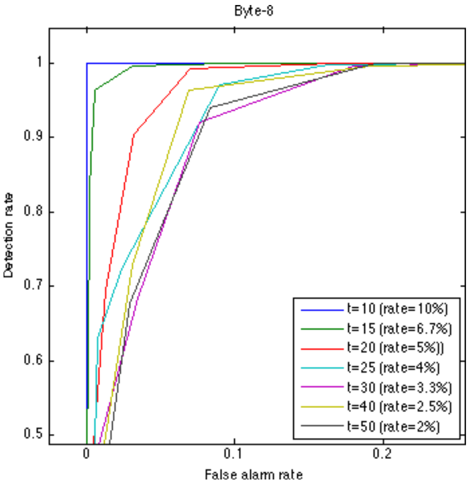


Fig. 5. ROC curves for different throughput rates. Feature: byte-8; w=1000. [Best viewed in color.]

A. General Model

We first present our model in full generality, leaving $\mathcal{D}$ and M unspecified. Later on we demonstrate how the model can be applied analytically in this section, and experimentally in Section VI with specific values of $\mathcal{D}$ and M.

Our goal is to model the effect of the knobs (the window length w, and the threshold τ) and the covert channel throughput (controlled by the variable t) on the rate of false positives and the rate of false negatives (which in turn determines the detection rate). First, we need to establish a baseline normal value for the metric against which the deviation is measured at the time of monitoring the stream. Our detection algorithm raises an alarm if it finds the deviation in any window is significant. The baseline is simply given by the expected value $\mathbb{E}[M] = \sum_W \Pr[W] \cdot M(W)$ of the metric in the absence of any covert channel, where the summation is over all possible windows W of length w. If $r = \ell t / w$ is the number of windows required to accommodate the covert channel at the given rate, then our detection algorithm detects and anomaly if, in any window $W_i, 1 \leq i \leq r$, we have $|M(W_i) - \mathbb{E}[M]| > \tau \cdot \mathbb{E}[M]$. We have the false-positive rate $x_w(\tau)$ and the false-negative rate $y_w(\tau)$ given by

$$x_w(\tau) = \Pr_{W \sim \mathcal{D}} \left[|M(W) - \mathbb{E}[M]| > \tau \cdot \mathbb{E}[M] \right], \text{ and} \quad (1)$$

$$y_w(\tau) = \Pr_{W_i^C \sim \mathcal{D}^C \forall i} \left[\left| M(W_i^C) - \mathbb{E}[M] \right| \leq \tau \cdot \mathbb{E}[M] \ \forall i \right]. \quad (2)$$

Note that, we consider tainted packets only while computing the false-negative rate, and not when computing the false-positive rate. In general, we expect the false-positive rate to decrease as window length is increased, given the reduced variance of the sampled distribution, as sample size increases.

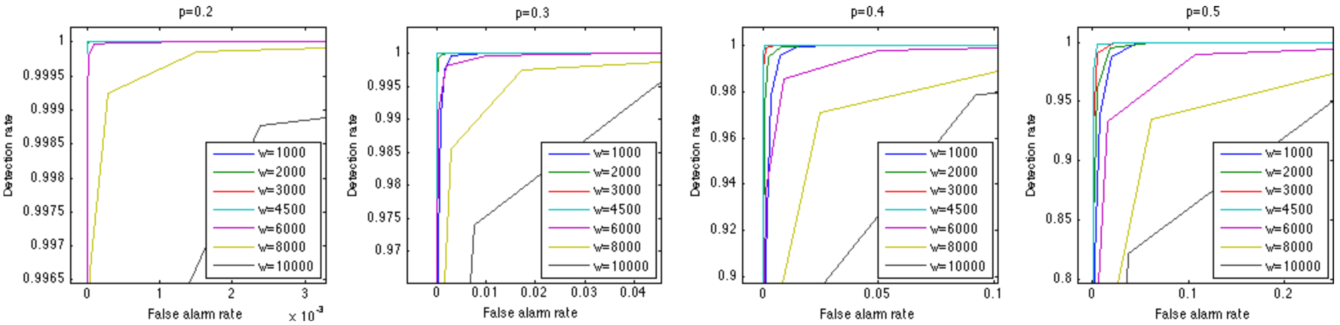


Fig. 2. ROC curves computed analytically for the distribution in Case 1. [Best viewed in color.]

Given the above model, the effect of w and τ on the false positive and false-negative rate can be observed by evaluating the probabilities $x_w(\tau)$ and $y_w(\tau)$. The effects can be visualized by plotting the ROC (Receiver Operating Characteristic) curves $R_w = [x_w(\tau), 1 - y_w(\tau)]$, for different fixed values of w, while varying τ.

The effect of the covert channel throughput (anomaly magnitude) can be observed by plotting the same ROC curves for different values of the throughput parameter t.

B. Applying the Model

We now demonstrate applications of our model to specific scenarios. For the metric M we consider the entropy H since it is the most widely studied. Two effects on distributions frequently occur due to anomalies. First, the anomaly introduces elements, all of the same value in the stream. Second, the anomaly introduces elements each of which has a distinct value. For instance, in network traffic, a port scan results in the effect of the second case on the destination port distribution, and the effect of the first case on the source IP distribution. For more examples and justification, see [17] (network traffic), and [6] (information networks). For brevity we model only the first case in the first analytical application below.

We apply analytical models for the following distributions.

1) All identical elements, with noise: In this case, we assume that the normal distribution $\mathcal{D}$ consists of all regular elements in the window being identical. But we allow for the possibility of noise, with the elements introduced by noise being all distinct. We model this distribution as follows. Without loss of generality, each element $x_i, 1 \leq i \leq w$ in the window is either 0 with probability $1-p$, or is a distinct non-zero random element with probability p, for $p \in [0,1]$. The parameter p controls the noise in the stream. The number of noise elements k in the window follows the Binomial Distribution $\mathcal{B}(w,p)$. If we let H_k denote the value of entropy when there are k noisy elements in the window, then we have

$$\begin{aligned} H_k &= H(\underbrace{1,1,\ldots,1}_{k \text{ times}}, w-k) \\ &= k \cdot (1/w)\log(w) + (w-k)/w \log(w/(w-k)), \\ \mathbb{E}[M] &= \mathbb{E}[H] = \sum_{k=0}^{w} \binom{w}{k} p^k (1-p)^{w-k} H_k. \end{aligned}$$

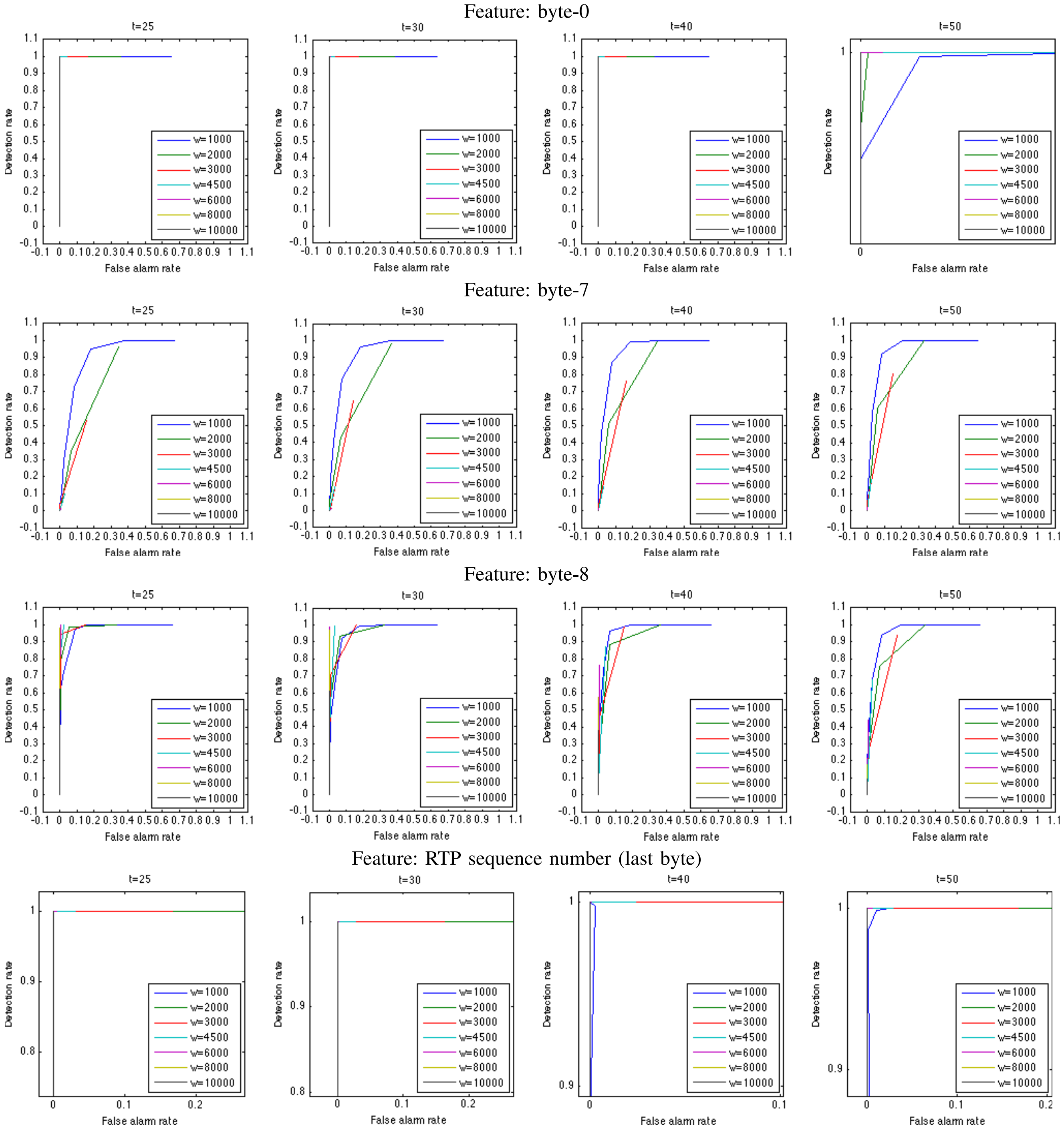
We can now compute $x_w(\tau)$ by simply finding all values of k for which $|H_k - \mathbb{E}[H]| > \tau \cdot \mathbb{E}[H]$, and summing up the corresponding probabilities $\binom{w}{k} p^k (1-p)^{w-k}$.

To compute $y_w(\tau)$, we need to incorporate the effect of the covert channel. We can write $y_w(\tau) = (y'_w(\tau))^r$, where $y'_w(\tau) = \Pr_{W^C \sim \mathcal{D}^C}\left[|M(W^C) - \mathbb{E}[M]| \leq \tau \cdot \mathbb{E}[M]\right]$ and r we recall is the number of windows with tainted packets (since windows are sampled independently). We consider the case when the covert channel introduces tainted elements that are all distinct and non-zero, and further that they are also distinct from noise elements, if any. If $w \leq \ell t/2$, we are guaranteed to have a complete window overlapping the covert channel. To see this consider the case when $w = \ell t$; it may happen that a window overlaps (and ends at) the first half of the covert channel (of length $\ell t/2$), and the following window overlaps (and begins at) the latter half of the covert channel. But when $w \leq \ell t/2$ this cannot happen and we will have $\lfloor w/t \rfloor$ tainted elements in some window. On the other hand, if $w > \ell t/2$, then we will have a window with at least $\ell/2$ tainted elements. So if we let $c = \lfloor \min(w, \ell t/2)/t \rfloor$, then we get a histogram with k distinct noise elements and $\geq c$ distinct tainted elements (we assume there are exactly c tainted elements to get a lower bound on the detection rate). The number of noise elements now follows $\mathcal{B}(w-c,p)$. If we now let H_k^C denote the value of entropy in the presence of the covert channel, we have

$$\begin{aligned} H_k^C &= H(\underbrace{1,1,\ldots,1}_{k+c \text{ times}}, w-k-c) \\ &= (k+c) \cdot \frac{1}{w}\log(w) + \frac{w-k-c}{w}\log\left(\frac{w}{w-k-c}\right), \end{aligned}$$

We can now compute $y'_w(\tau)$ by simply finding all values of k for which $\left|H_k^C - \mathbb{E}[H]\right| > \tau \cdot \mathbb{E}[H]$, and summing up the corresponding probabilities $\binom{w-c}{k} p^k (1-p)^{w-k-c}$.

Observations: The ROC curves of the false positives and detection rates obtained in this manner are shown in Figure 2. The covert channel in the plots are introduced with parameters $\ell = 600$, and $t = 15$. There are four plots, one for each value of p that we set. We observe that, other factors being equal, the ROC curves drift towards the 45-degree line (classifier quality degrades) as the noise probability increases. This result is primarily due to the increase in variance and hence the rate of false alarms as the noise levels increase. With a fixed value



Feature: byte-0

Feature: byte-7

Feature: byte-8

Feature: RTP sequence number (last byte)

Fig. 3. ROC curves computed analytically for Case 2 with various features. [Best viewed in color.]

of p within a plot we see, however, that as window length is increased, the ROC curves first approach the ideal classifier, until the window length is $w = \ell t/2 = 4500$. Thereafter, as the window length is increased further, the classifier quality deteriorates towards the 45-degree diagonal line. This suggests that the ideal window length for detection is $w = \ell t/2$. This may be clear as a consequence of the maximum value $\ell t/2$ possible for the parameter c. However, the reason is not completely clear merely by looking at the expressions for $x_w(\tau)$ and $y_w(\tau)$. A direct reason for the suggested setting of w is that as the window length is increased, beyond the length $\ell t/2$, the proportion of tainted packets in each window containing the covert channel keeps declining, resulting in a lower detection rate.

2) Multinomial distribution: In this general case we assume the normal distribution $\mathcal{D}$ consists of each element $i \in [n]$ being produced independently with some probability p_i in the stream ($\sum_{i=0}^{n-1} p_i = 1$). That is, the number of occurrences (frequency) m_i of any element i in a window of length w is distributed according to the multinomial distribution $\mathcal{M}(p_0, p_1, \ldots, p_{n-1}, w)$. Then for each possible frequency vector $\vec{m} = (m_0, m_1, \ldots, m_{n-1})$ possible in the window (i.e., for each such vector with $\sum_{i=0}^{n-1} m_i = w$), we have that $\Pr[\vec{m}] = w!/(m_0!m_1! \ldots m_{n-1}!)p_0^{m_0} p_1^{m_1} \ldots p_{n-1}^{m_{n-1}}$. The probabilities $p_i, 0 \leq i \leq n-1$ may be determined by observing a "training set" stream of regular elements. If we denote the entropy with frequency vector $\vec{m}$ as $H(\vec{m})$, then we have

$$\begin{aligned} H(\vec{m}) &= \sum_{i=0}^{n-1} m_i/w \log(w/m_i), \text{ and} \\ \mathbb{E}[H] &= \sum_{\vec{m} \text{ s.t. } \sum_i m_i = w} \Pr[\vec{m}] \cdot H(\vec{m}). \end{aligned}$$

We are now in a position to compute $x_w(\tau)$ from Equation 1 as before by summing up the relevant probabilities. We compute $y_w(\tau)$ similarly as before from Equation 2, only this time we use the Multinomial Distribution $\mathcal{M}(p_0^C, p_1^C, \ldots, p_{n-1}^C, w)$ instead that incorporates the covert channel. The probabilities p_i^C now may be obtained from a training set stream having the covert channel.

Computation: The difficulty of computing $x_w(\tau)$ and $y_w(\tau)$ is that it requires looping over the set of all possible vectors $\vec{m}$ such that $\sum_{i=0}^{n-1} m_i = w$. Generating such vectors is a non-trivial process, but more importantly as w increases, the number of such vectors grows at a rate that is exponential or greater. So, instead of computing $x_w(\tau)$ and $y_w(\tau)$ directly from the formulas, we instead perform a Monte-Carlo simulation to sample values from distribution $\mathcal{D}$, and use those values to compute first $\mathbb{E}[H]$ and then $x_w(\tau)$ and $y_w(\tau)$. With a large number of simulations we expect to get results that are close to the actual values of $x_w(\tau)$ and $y_w(\tau)$. We ran at least 1000 simulations in each case, and more as needed to ensure that the change in the values computed using the samples was not more than 0.1%.

Parameter ranges: We compute $x_w(\tau)$ and $y_w(\tau)$ in this manner for t = 10, 15, 20, 25, 30, 40, 50, and $\ell = 600$, as is the case in our experiments in the next section. We use window lengths from $w = 1000, 2000, 3000, 4500, 6000, 8000, 10000, 20000, 30000$. These values straddle the length of the covert channel ℓt for the values of t. We observed in previous models that the classification quality shows an inflection point at $w = \ell t/2$.

Features: We consider the following features that are affected by our covert channel described in Section III:

- Byte 0 of RTP payload. In tainted packets this byte contains the fixed magic number, whereas in regular packets the byte is distributed according to bytes generated by the G.711a voice encoding codec over our input voice stream. For the covert distribution, we chose probabilities p_i^C with $p_k^C = 1$, and $p_i^C = 0$ for $i \neq k$, where k is the magic number. On the other hand for the regular feature probability distribution, we computed probabilities p_i using a training set generated from a real VoIP call with about 350,000 RTP packets. This training set is described in Section VI. We found the probability distribution p_i to be close to a uniform distribution.
- Byte 7 of RTP payload. This byte contains a sequence number used internally by the covert channel. The sequence number is incremented sequentially for every tainted packet. In regular packets, this feature contains G.711a voice encoded bytes. For the regular packet we chose the probabilities p_i from the training set (again we found it is close to normal). We set the covert channel probabilities $p_i^C = 1/n$, i.e., uniformly distributed (the best we can do given the multinomial distribution model).
- Byte 8 of RTP payload. In regular packets, this feature contains G.711a voice encoded bytes. For the regular packets we choose the probabilities p_i from the training set (again we find it is close to uniform). In the tainted packets this byte is distributed according to ASCII text. We used our entire text input, the text of *Alice's Adventures in Wonderland*, to compute the covert channel probabilities p_i. Note, however, that in sending the text via the covert channel, only specific bytes from the file at periodic offsets will appear in this byte position in the covert channel. The other bytes will occupy other bytes in the covert payload.
- RTP sequence number. This is the last byte of the RTP sequence number that is incremented sequentially for each RTP packet. For regular packets, we set the probabilities $p_i = 1/n$, i.e., uniformly distributed (the best we can do given the multinomial distribution model). Our covert channel relies on setting the RTP sequence number to zero. So, for tainted packets, we set the probabilities p_i^C such that $p_0^C = 1$, and $p_i^C = 0$ for $i \neq 0$.

The ROC curves computed by this analysis are in Figure 3. For brevity, we only include the plots for $t = 25, 30, 40, 50$.

Observations: As t increases (i.e., covert channel throughput decreases) we observe that the ROC curves tend to move toward the 45-degree diagonal. This indicates that performance degrades as a result of decrease in detection rate due to the dilution of tainted packets by regular packets in covert channel windows. The effect of t on the false-alarm rate is a bit harder to observe from the figure, but careful observation (e.g., by looking at the end-points of the ROC curves) shows that the false-alarm rate remains in the same interval range for all values of t, indicating that the false-alarm rate was not affected by t. This is simply because the false-alarm rate is computed in the absence of anomalies.

Now, as w is increased, we observe consistently in the plots that the false-alarm rate decreases. The reason for this observation may be that as the number of samples in the window from the underlying distribution is increased, the variance decreases (analogous to the Law of Large Numbers). However, the effect of w on the detection rate is not consistent. As w increases beyond $\ell t/2$ we expect the detection rate *per window* to decrease since the proportion of tainted elements decreases in a window (as also observed in Case 1). On the other hand, when $w < \ell t/2$ the detection rate per channel may decrease because of the smaller number of windows, when the window length is decreased. But although the proportion of tainted elements does not change in this case, different metrics react in different ways to the change in the window length and the detection rate may be determined by such metric behavior.

The combined effect of the detection rate and the false-alarm rate makes it hard to see the change in the detection rate alone by increasing w. To see the effect of detection rate separately, in Figure 4 we plot it against various values of the threshold τ (for the byte-0 feature with $t = 10$). We observe that for each window length, the detection rate fell from 1 to 0 as the threshold is increased. However, the fall was at a much later threshold for window lengths 1000, 2000 and 3000, and as the window length is increased we find that the fall occurs earlier. This observation indicates that when the window length was increased beyond 3000, the detection rate decreased.

When $w < \ell t/2$, with the byte-7 and byte-8 features, we see the detection rate mostly decreasing with increase in w. But with byte-0 and the RTP sequence number the detection rate increased as w is increased (at least for $t = 50$). We attribute this inconsistent behavior to the way the metric behaves under different distribution combinations (i.e., tainted element distribution and regular element distribution).

For the byte-8 feature we find that at higher throughputs ($t = 25, 30$) the ROC curve tends away from the 45-degree diagonal as w is increased. When looking at the ROC curve as a whole we observe that for our features, as the window length w is increased, the curve may drift away from the 45-degree diagonal or towards it depending on the feature and throughput.

Effect of Throughput: Our framework of analysis allows us also to see what effect the covert channel throughput would have on the detection quality. To see this effect, we must now fix a value of w and plot the ROC curves corresponding to different values of the parameter t, and then repeat the process for different values of w. For brevity, here, we present only the results for $w = 1000$, with the byte-8 feature. The practitioner should apply the same process for different values of w while modeling his setting. The results are shown in Figure 5. We observe, as expected, that as the rate (as a fraction of the total VoIP traffic) decreased from 10% to 2%, the ROC curve moved towards the 45-degree diagonal, indicating that the detection quality is getting poorer. These curves enable the practitioner to understand what rates of covert channel he has a hope of detecting within his acceptable false alarm rate.

VI. Experimental Analysis

In the previous section, our analytical models let us study some of the ways in which the knobs affect detection via our analytical models. In this section we look at the effects experimentally to understand the effects directly with actual calls in our testbed.

Dataset: First we describe our call setup and the datasets we use in our experiments. Our testbed consists of two client machines registered with an Asterisk SIP server. The clients are able to make calls to each other through their registration with the server, but the server is configured so that the RTP packets are exchanged directly between the clients without being routed through the server. Our dataset is generated by making calls in this manner. For the audio input in the call, we play a French language training audio program on both ends of the call. We save packet capture traces of calls, and our dataset consists of these traces. We save one trace consisting of approximately 350,000 RTP packets in each direction. This trace does not contain any tainted packets and serves to compute the expected normal value of entropy, and also to compute the false-alarm rate. We also capture traces while running our covert channel at different throughputs. The payload for our covert channel is the entire text of *Alice's Adventures in Wonderland* sent in 256 byte chunks per tainted packet. This results in a total of 602 tainted packets. For each value $t = 10, 15, 20, 25, 30, 40, 50$, we run a covert channel and save the call packet trace separately. These traces contain approximately just enough packets to include all tainted packets at the given covert channel throughput.

Experiments: We compute the entropy with window lengths $w = 1000, 2000, 3000, 4500, 6000, 8000, 10000, 20000, 30000,$ in each of our data sets. In the data sets containing the covert channel we ensure that we do not consider any windows that do not contain any tainted packets at all. To increase our sample set size, in each trace, we slice the stream into windows, by first starting at different offsets $(1, 2, \ldots, 1000)$ from the start of the trace. We then compute the entropy in each window. Using these entropy values we are now able to compute the expected value $\mathbb{E}[H]$ and the false-alarm rate $x_w(\tau)$ from the trace not containing the covert channel, and then compute the detection rate $y_w(\tau)$ from the traces containing the covert channel.

Observations: The ROC curves from our experiments are shown in Figure 6. We have not included the plots for the RTP sequence feature, because we find that the false-alarm rate is always zero, leading to an empty figure. The reason for the zero false-alarm rate is that the distribution in each window is always the same—it consists of sequence

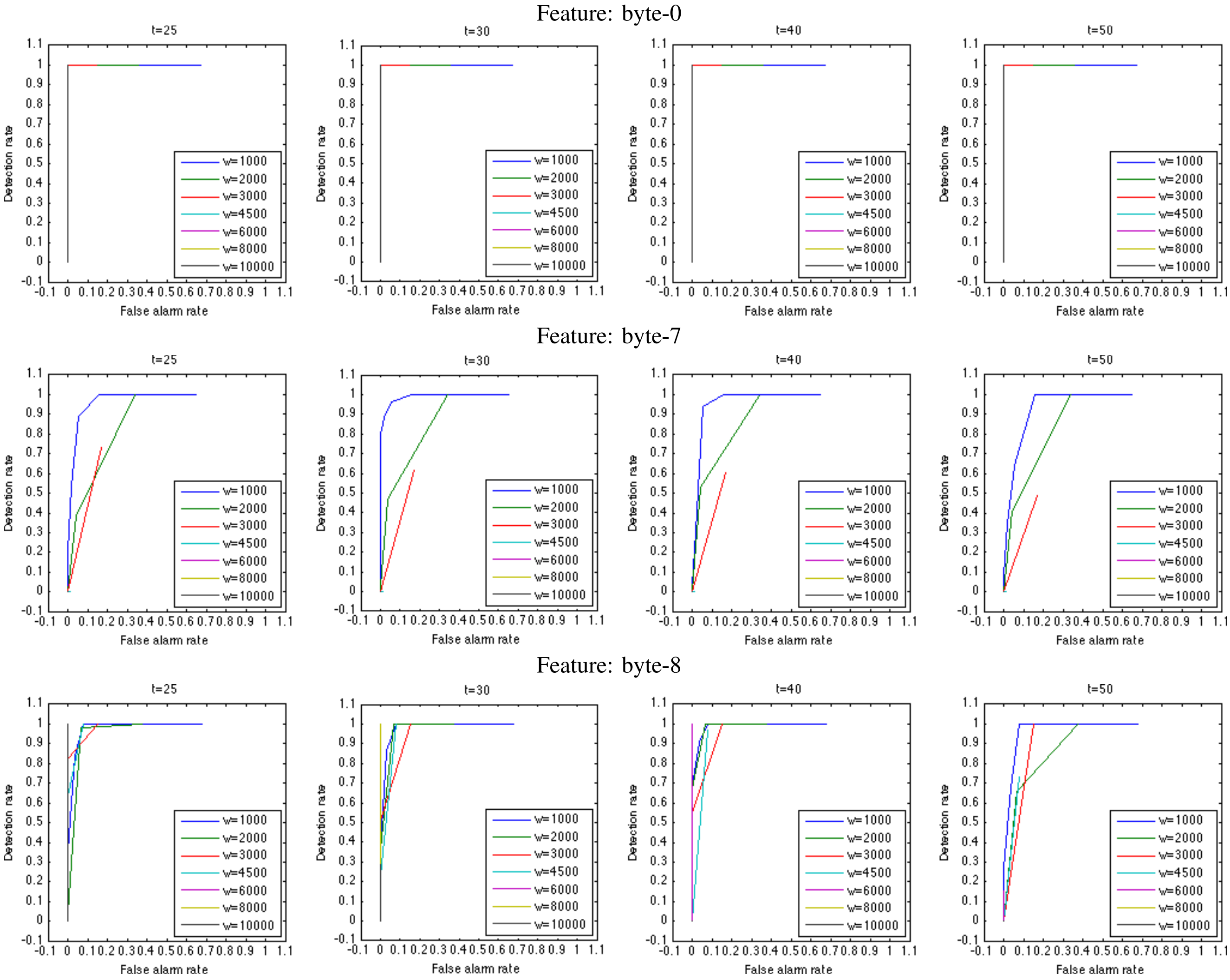


Fig. 6. ROC curves computed experimentally with various features. [Best viewed in color.]

numbers increasing consecutively. Note that in the analytical model in the previous section, the assumptions of the model—the underlying multinomial distribution—results in a non-zero false-positive rate in some cases. This demonstrates that the assumptions made by our analytical model may not always hold in practice. For other features our observations are similar to the analytical case. We found that increasing t tended to a poorer detection rate, resulting in a poorer overall detection quality even though the false-alarm rate was unaffected. For the effect of increase in w, we generally found that an increase in w led to a decrease in the detection rate, and a decrease in the false-alarm rate.

As in the analytical case, at higher throughputs for byte-8 we see the ROC curve move away from the 45-degree diagonal as w increases. But overall otherwise, we find, as in the analytical case, that as the window length is increased, the ROC curves move toward the 45-degree diagonal.

VII. Recommendations

In this section we provide recommendations to a practitioner based on our results on how to tune the knobs for distribution-based anomaly detection methods in his specific setting. We also provide recommendations on how to efficiently apply the methods using sketching algorithms.

Our results show that,

- All other factors being equal, as the window length w is increased, the detection rate may increase or decrease. The behavior may be different for $w < \ell t/2$ and for $w > \ell t/2$. The false-alarm rate, on the other hand, decreases monotonically as the window length is increased. This effect is attributed to the decrease in the variance as the number of samples in the window is increased. If the distribution of regular elements is fixed in each window (e.g., the RTP sequence number), the false positive rate is always zero (even as window length is increased).

- For a fixed window length, all other factors being equal, as the threshold is increased, both the false-alarm rate and the detection rate decrease. This effect is in agreement with our basic intuition.
- For a fixed window length, all other factors being equal, the detection rate increases with the increase in covert channel throughput. The covert channel rate has no bearing on the false-alarm rate, since the false-alarm rate is computed in the absence of anomalies.

Since the rate of change of the detection rate and false-alarm rate might be independent, an optimal tuning of the window length might depend on the covert channel throughput. Thus, it may be necessary to run the detection methods with several different window lengths, depending on the throughputs of the covert channels that one expects. Below we describe how detection methods with multiple window lengths can be employed efficiently. To tune the window length w and the threshold τ the practitioner must repeat our factorial-design-like analysis to compute the false-alarm rate and detection rate for several settings of the knobs and the covert channel throughput and compare them via ROC curves.

Efficient Monitoring with Sketching Algorithms: We now discuss how we can run distribution-based methods efficiently when analysis like ours suggests it is necessary to do so at different window lengths. As we mentioned in Section IV, data stream algorithms enable efficient online estimation of metrics of distributions when the input arrives in a stream-like fashion. Suppose the stream σ consists of m elements $a_1, a_2, \ldots, a_m$ from the universe $[n]$. A data-stream algorithm estimates metric $M(\sigma)$ efficiently with respect to memory usage and computation. In our case, the stream σ is the window W for each window in network traffic. When the traffic rate is massive, it is not feasible to compute $M(W)$ by first storing the entire window in memory. The problem is compounded when many features and metrics are to be monitored. Further, as we observe, monitoring metrics at different window lengths may be required. Certain data-stream algorithms, called sketching algorithms, address these problems. These algorithms maintain a data structure $\mathcal{D}(W)$ called a *sketch*, from which the metric value $M(W)$ can be estimated. Sketches have the property that they can be combined efficiently: for two disjoint windows W_1 and W_2, given only their sketches $\mathcal{D}(W_1)$ and $\mathcal{D}(W_2)$, it is possible to combine them to obtain $D(W_1 \cup W_2)$. So in addition to estimating $M(W_1)$ and $M(W_2)$, sketches allow us to additionally estimate $M(W_1 \cup W_2)$ at almost no additional computational cost. The property extends to the case when sketches for multiple disjoint windows are provided. Sketching algorithms for computing entropy [15] and the frequency moments [8] are available.

Now, when our analysis suggests that we run our distribution-based method with window lengths $w_1, w_2, \ldots, w_t$, we can run one instance of the distribution-based method with a window length of $w = \text{GCD}(w_1, w_2, \ldots, w_t)$ sketching algorithms. Then, to estimate the metric value $M(W)$ in a window W of length w_i, we only need to combine the sketches from the w_i/w sub-windows, and estimate $M(W)$ from the combined sketch. Note, however, that the error bounds on the estimates are usually a function of the length of the stream, so that applying this method to combine a large number of windows may result in larger estimation errors. We refer the reader to the relevant algorithms [15], [8] for specific details on the error bounds.

VIII. Conclusions

We showed the complex effects the knobs of distribution-based methods can have on detection quality. We provide a analysis framework that practitioners case use to tune these knobs when implementing such methods in their own environments. We observe that in many scenarios it will be necessary to employ detection methods using several window lengths. In such cases, we provide an efficient approach using sketches to minimize memory and computation overhead.

References

[1] Department of Defense. 1985. Trusted Computer System Evaluation Criteria. DoD 5200.28-STD.

[2] http://steganrtp.sourceforge.net/.

[3] http://www.jjtc.com/Security/stegtools.htm.

[4] http://www.sans.org/security-resources/idfaq/covert_chan.php.

[5] C Arackaparambil, S Bratus, J Brody, and A Shubina. Distributed monitoring of conditional entropy for anomaly detection in streams. In *Proc. of IEEE Workshop on Scalable Stream Processing Systems*, 2010.

[6] C Arackaparambil and G Yan. Wiki-watchdog: Anomaly detection in Wikipedia through a distributional lens. In *Proc. of IEEE/ACM Web Intelligence*, 2011.

[7] P Barford, J Kline, D Plonka, and A Ron. A signal analysis of network traffic anomalies. In *Proc. of SIGCOMM Workshop on Internet Measurment*, 2002.

[8] L Bhuvanagiri, S Ganguly, D Kesh, and C Saha. Simpler algorithm for estimating frequency moments of data streams. In *Proc. of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 708–713, 2006.

[9] M J Cooper and H Gulen. Is Time-Series Based Predictability Evident in Real Time? *SSRN eLibrary*, 2004.

[10] V Crespi, G Cybenko, and A Giani. Attacking and defending covert channels and behavioral models. *CoRR*, abs/1104.5071, 2011.

[11] A D'Alconzo, A Coluccia, and P Romirer-Maierhofer. Distribution-based anomaly detection in 3g mobile networks: from theory to practice. *Int. J. Netw. Manag.*, 20:245–269, September 2010.

[12] A Giani, V Berk, and G Cybenko. Data exfiltration and covert channels. In *Proc. of Sensors, and Command, Control, Comm., and Intelligence Technologies for Homeland Security and Homeland Defense IV*, 2006.

[13] S Gianvecchio and H Wang. Detecting covert timing channels: an entropy-based approach. In *Proc. of ACM CCS*, pages 307–316, 2007.

[14] C Gomez and R Hornero. Entropy and complexity analyses in alzheimer's disease: An MEG study. *Open Biomed Eng J*, 4:223–35, 2010.

[15] N Harvey, J Nelson, and K Onak. Sketching and streaming entropy via approximation theory. In *Proc. of 49th IEEE FOCS*, 2008.

[16] Y F Huang, S Tang, and J Yuan. Steganography in inactive frames of VoIP streams encoded by source codec. *Information Forensics and Security, IEEE Transactions on*, 6(2):296 –306, june 2011.

[17] A Lakhina, M Crovella, and C Diot. Mining anomalies using traffic feature distributions. In *Proc. of SIGCOMM*, 2005.

[18] A Lall, V Sekar, M Ogihara, J Xu, and H Zhang. Data streaming algorithms for estimating entropy of network traffic. *SIGMETRICS Perform. Eval. Rev.*, 34(1):145–156, 2006.

[19] W Lee and D Xiang. Information-theoretic measures for anomaly detection. In *Proc. of IEEE Symposium on Security and Privacy*, 2001.

[20] G Yan, S Eidenbenz, and E Galli. SMS-Watchdog: Profiling social behaviors of SMS users for anomaly detection. In *Proc. of RAID*, 2009.

[21] S Zhang, A Chakrabarti, J Ford, and F Makedon. Attack detection in time series for recommender systems. In *Proc. of ACM SIGKDD KDD'06*, pages 809–814, 2006.