INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 2010; **20:** 295–316
Published online in Wiley Online Library
(wileyonlinelibrary.com) DOI: 10.1002/nem.749

An evaluation of automatic parameter tuning of a statistics-based anomaly detection algorithm

Yosuke Himura[1,*,†], Kensuke Fukuda[2], Kenjiro Cho[3] and Hiroshi Esaki[1]

[1] *Graduate School of Information Science and Technology, University of Tokyo, Tokyo, Japan*
[2] *National Institute of Informatics/PRESTO, JST, Tokyo, Japan*
[3] *Internet Initiative Japan, Tokyo, Japan*

SUMMARY

We investigate an automatic and dynamic parameter tuning of a statistical method for detecting anomalies in network traffic (this tuning is referred to as *parameter learning*) towards real-time detection. The main idea behind the dynamic tuning is to predict an appropriate parameter for upcoming traffic by considering the detection results of past τ traces of traffic. The τ is referred to as the *learning period*, and we discuss in particular the appropriate value of τ. This automatic tuning scheme is applied to parameter setting of an anomaly detection method based on Sketch and the multi-scale gamma model, which is an unsupervised method and does not need predefined data. We analyze the tuning scheme with real traffic traces measured on a trans-Pacific link over 9 years (15 min from 14:00 Japan Standard Time every day, and 24 consecutive hours for some dates on the same link). The detection results with parameter prediction are compared to those with ideal parameters that maximize the detection performance for upcoming traffic. We also analyze predictability of the ideal parameter considering the past changes in it. The main findings of this work are as follows: (1) the ideal parameter fluctuates day by day; (2) parameter learning with a longer τ is affected by significant events included in the period, and the appropriate τ is about three traces (days) for everyday 15 min traces and around 1.5 h for 24 h traces; (3) the degradation in detection performance caused by introducing parameter learning is 17% with $\tau = 3$ for everyday 15 min traces; (4) the changes in the ideal parameter have no periodic correlation, and can be modeled as a random process followed by a normal distribution. We show that one cannot consistently use a fixed parameter in statistics-based algorithms to detect anomalies in practice. Copyright © 2010 John Wiley & Sons, Ltd.

1. INTRODUCTION

As the Internet has become an infrastructure essential to human life, all users are exposed to considerable threats such as those posed by viruses and worms, and distributed denial-of-service attacks. These menaces must be detected in real time so that secure networks can be maintained, and numerous methods of detecting anomalies in network have been proposed thus far. These anomaly detectors are generally classified into two categories; the first is a signature-based approach, which finds specific fingerprints of packet payloads predefined in their database [1], and the second is a statistics-based approach, which defines network anomalies as deviations from referential statistical behavior [2–11]. These statistical models are especially needed for detecting unknown anomalies, e.g. outbreaks of new worms (zero-day attacks), which are currently common in the Internet. Classical methods find

[*] Correspondence to: Yosuke Himura, Esaki laboratory, Room 102B, Building 2 of Engineering Department, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
[†] E-mail: him@hongo.wide.ad.jp
CGS: Strategic International Cooperative Program of CNRS and JST

Received 10 February 2010
Revised 19 March 2010
Accepted 12 April 2010

abrupt changes in traffic volume [2–6], and recent detectors have tried to uncover hidden anomalies adopting machine learning or statistical models [7,8,10].

Even though a great deal of attention has been paid to statistical anomaly detection methods, one crucial problem not fully discussed is how parameters should be set with these methods [12,13]. Arbitrarily determined parameters may detect significant (e.g. high-volume) anomalous traffic, but they will miss minor (hidden) anomalies. Thus the parameter setting of statistical models directly affects the performance (i.e. accuracy) of anomaly detection. In addition, it is generally difficult to choose appropriate parameters a priori, because the macroscopic dynamics of anomalies in Internet traffic vary greatly depending on the temporal and spatial features to be measured. The changes in anomalous behavior pose difficulties in learning-based detectors [7,8,10], which defines anomalies by analyzing past traffic behavior to extract low-intensity anomalies. Even though some earlier studies have addressed how to set parameters [5,14] on the fly, this parameter setting is just a conversion from a confidence value (e.g. 99% or 95%) to a threshold. Hence this tuning does not consider the dynamics of anomalies or the efficiency of anomaly detection, and even if we used such a parameter setting based on the confidence level, we would need to find a better confidence level leading to better detection, which is quantified by ground truth. Consequently, there has been demand for an automatic tuning scheme that dynamically follows macroscopic trends in anomalous traffic, in order to realize accurate and real-time anomaly detection. Furthermore, automatic and dynamic parameter tuning is essential not only for detecting anomalies in the real world but also for comparing anomaly detectors. To ensure a fair comparison of detection performance, the parameters of all methods must be optimized in addition to using the same dataset. One reasonable approach toward this demand is to predict the behavior of anomalies in upcoming traffic by considering their trends in the past, in order to automatically tune anomaly detectors. The tuning scheme will train anomaly detectors to uncover high-abnormality traffic (e.g. scanning or flooding attack) and provide network operators with detection reports including high-priority anomalies so that they can efficiently decide how to deal with the detected traffic according to their management policy. This tuning is different from learning-based anomaly detectors, because they still have to set thresholds to find anomalies after defining referential behavior computed from learning with traffic traces. Also, automatic parameter tuning scheme can be applied to many kinds of anomaly detectors.

This paper discusses automatic and dynamic parameter tuning, referred to as *parameter learning*. To the best of our knowledge, this is the first intensive work on dynamic parameter tuning for anomaly detectors. The main idea behind our learning framework is that we predict an appropriate parameter for upcoming traffic by considering the detection result of the past several traffic traces. We particularly focus on the *learning period* τ, which is the number of traces needed for parameter prediction. Intuitively, τ involves a trade-off between (a) a lower τ, which leads to lower accuracy because there are fewer data points to calculate statistics appropriately, and (b) a higher τ, which will miss abrupt changes in the macroscopic dynamics of anomalies in network traffic. We investigate this trade-off by evaluating an anomaly detector based on the multi-scale gamma model [9], which is an unsupervised learning tool and does not need predefined data, so this tool is promising for security and suitable for discussing evaluation results. We analyze this tuning scheme with real traffic traces measured at a trans-Pacific link over 9 years (15 min from 14:00 Japan Standard Time (JST) every day, and 24 h for some dates) with pseudo ground truth generator validated by BLINC [15]. Thus far, some of these issues have been addressed in our earlier work [13], and we have additionally (a) analyzed how predictable the ideal parameter is with respect to periodicity and a random process and (b) extended the dataset including 9-year-long traffic traces and three sets of 24 h consecutive traces. The four main results obtained from this work are that (1) the ideal parameter fluctuates daily and that (2) parameter learning with a longer τ is affected by significant events included in the period, and the appropriate τ is about three traces (days) for everyday 15 min traces and about 1.5 h for 24 h traces. We have also found that (3) the performance degradation caused by introducing parameter learning is 17% with $\tau = 3$, and that (4) the changes in the ideal parameter have no periodical correlation, and can be modeled as a random process followed by a normal distribution. The contribution of this paper is to clarify and quantify the importance of setting dynamic parameters with statistics-based anomaly detectors in the real world, i.e. it is ineffective to continuously use fixed values for parameters.

2. PRELIMINARIES

This section introduces three preliminaries involving an anomaly detection method (Section 2.1), traffic traces collected over the long term and a traffic labeling scheme (or pseudo ground truth generator) to evaluate detection performance (Section 2.2), and an automatic parameter tuning method (Section 2.3).

2.1 Anomaly detection algorithm based on Sketch and multi-scale gamma model

We use an algorithm based on Sketch and the multi-scale gamma model [9]. The reasons for adopting this method are (a) the technical advantage of detecting low-intensity (hidden) anomalies on multiple timescales, (b) the ability to detect unknown events deviating from dynamically calculated reference data (i.e. no need for learning data or predefined anomalies), and (c) availability of an implementation tuned practically. The appropriateness of using a multi-scale gamma model has been discussed elsewhere [16]. The main idea underlying detection is to find outliers in the normalized values of statistics computed from longitudinal behavior. The method is currently used in detecting source hosts that generate anomalous traffic, and the overview of the detection procedure includes four steps:

1. Sketch: traffic is divided approximately into a set of sub-traffic from a source host. This is done with the hash function of a quasi-huge hash table called Sketch [4] and using a source IP address for hashing keys. This quasi-huge hash table is created with N hash functions of table-size M.
2. Multi-scale gamma model: for each piece of sub-traffic, the histogram for the number of packets arriving during a certain timescale Δ is approximated as a gamma distribution. The gamma distribution has two parameters: α determines the shape of the histogram, and β the scale. α is helpful for detecting hidden anomalies. This approximation is conducted on multiple timescales ($\Delta = \Delta_0 \times 2^j$ with $j = 0, \dots, 7$), and the computed parameters are aggregated as one measure for each α and β.
3. Anomaly identification: the α of the set of sub-traffic are compared each other, and the sub-traffic having outlier α is an anomaly, that is, if $|\alpha - \mu_\alpha| > \theta_\alpha \times \sigma_\alpha$, the sub-traffic is judged to be an anomaly, where μ_α and σ_α stand for the average and standard deviation of α among the set of sub-traffic in the data, and θ_α is the threshold for α. The same technique is applied to β.
4. The above procedures are carried out for each time window T. A source host detected over at least one window is regarded as one anomaly. The detector focuses on host that sends more than P packets in a trace, which is referred to as an *event*, to calculate reliable statistics (α and β). Also, a large amount of packets allows us to identify whether the detected traffic is truly an anomaly.

Since $\frac{|\alpha - \mu_\alpha|}{\sigma_\alpha}$ is a kind of Mahalanobis distance (i.e. a normalized metric), the threshold θ_α represents the normalized degree of deviation from referential behavior. Hence lower θ_α leads to a higher number of detected events, including all events detected by any higher θ_α.

For all traces, we empirically set $N = 8$, $M = 32$, $P = 1000$, and $\Delta_0 = 5$ ms, following Dewaele *et al.* [9], and $T = 15$ min. Although the algorithm requires several parameters, we concentrate on a main parameter θ_α and do not use θ_β, because the changes in the shape parameter clarify low-intensity anomalies (i.e. θ_α can detect hidden anomalies better than θ_β), and the use of both thresholds leads to complicated discussion on evaluation results. Also, detection by θ_α and by θ_β are independent, so this choice for evaluation of learning is compatible with other studies using this anomaly detector. Also, this detection tool empirically focuses on host traffic composed of more than 1000 packets per trace (which we call an *event*), because larger numbers of packets allow us to identify abnormalities in detected traffic. After this, the main threshold θ_α is referred to as θ.

DOI: 10.1002/nem

2.2 MAWI dataset and predefined pseudo ground truth for computing detection performance

Rather than using synthetic traffic, we conduct our evaluation by using a measurement and analysis of wide-area Internet (MAWI) traffic repository [17]. The traffic traces have been captured at a trans-Pacific link between the USA and Japan (from 14:00 to 14:15 JST) since 2001, and they consist of 15 min pcap traces. The payloads of all packets were removed for all traces, and both source and destination IP addresses were anonymized, preserving the prefix structure. The observed link had been upgraded twice up to the end of 2009; the first was from 18 Mbps to 100 Mbps in July 2006, and the second was from 100 Mbps to 150 Mbps in July 2007. The link was congested before the first upgrade. In addition to the 15 min traces every day, we used some consecutive 24 h traces known as the Day in the Life of the Internet (DITL) dataset [18] (these data are also collected at the same link and stored as 15 min traces). The long-term everyday traces and 24 h all-day traces will subsequently be referred to as *15 min traces* and *24 h traces*.

We analyze 15 min traces from 1 January 2001 to 31 December 2009, and 24 h traces on 3 March 2006, 10 January 2010, and 19 March 2008. The long-term measurements make this dataset suitable for investigating changes in optimal parameters for statistical anomaly detectors. In addition, the dataset provides us with generality of results: time generality thanks to the long measurements, and link state generality due to the two upgrades. Moreover, as these traffic traces are available to the public, using this dataset adds reproducibility and comparability to our study. (Also, this dataset has been well studied [9,19,20]). This dataset has 3098 traces from 1 January 2001 to 31 December 2009 (some traces could not be provided because of measurement failures), i.e. 900 GB in size in the gzip file format. Even though these 15 min traces are not consecutive (every 15 min throughout the day), the measurement location and the start time (14:00 JST) for all traces are the same. Hence these traces are sufficient to evaluate the algorithm's ability to follow macroscopic (e.g. several days level) changes and microscopic (e.g. hours or 1-day level) changes in traffic anomalies to be discussed.

We classify events into six categories (Attack, Victim, Warning, OK, Special, and Unknown) according to the *abnormality* (or harmfulness) of the events by using our heuristics based on port number, TCP flag, and communication structure as used in Himura *et al*. [13]. This heuristics is a hybrid made up of a traditional approach (based on port number) and a state-of-the-art method (based on communication pattern [15]) to leverage the advantages of both. Table 1 lists the categories and examples of heuristics, and we show all rules to identify Attack events in Appendix A. Also, we validate these heuristics in Appendix B by Reverse BLINC used in Kim *et al*. [21], showing that these heuristics can uncover more Attack events, including those identified by BLINC. Here, we define *Anomalous* as including both Attack and Victim categories, and we define *Normal* as including Warning, OK, Special, and Unknown categories. We found out that most Unknown events are generated by P2P software as shown in Appendix B. Figure 1 shows the classification results of the MAWI dataset, plotting the number of events in a trace. Figure 1(a) shows the evolution of the breakdown over 9 years. The continuously high number of Anomalous events in the last half of 2004 is due to the massive outbreak of a worm. The other figures show that of 24 h traces: (b) 3 March 2006, (c) 10 January 2007, and (d) 19 March 2008. For each figure, there are constant number of Anomalous events, whereas Normal events represent the daily periodicity. With these data, we can identify whether an event reported by the anomaly detector is an actual anomaly or not, so that these heuristics enable us to investigate parameter learning.

2.3 Automatic parameter tuning and learning period

Here we explain the automatic parameter tuning method, which is schematically represented in Figure 2. The basic idea behind this method is that the optimal threshold for upcoming traffic on date t (referred to as $\theta_\tau(t)$) is determined to be the one that has produced the best accuracy (detection performance a($\theta_\tau(t-1)$)) in the past τ traffic traces on dates $t-1, \ldots, t-\tau$. We define this τ as the *learning period*. The detailed procedure to determine θ_τ involves five steps. (modification note: we use $\theta_\tau(t)$ instead of θ_τ if needed.)

DOI: 10.1002/nem

Category	Explanation of category	Example of heuristics
Attack	The host sends many malicious packets	If SYN flagged packets account for more than 20% of all packets, then the host is regarded as 'an attacker of SYN flooding attack' and the event is classified into the Attack category
Victim	The host receives many malicious packets	If the ratio of SYN/ACK flagged packets is more than 20%, then the host is regarded as 'a victim of a SYN flooding attack' and the event is classified into the Victim category
Warning	The host is legitimate, but can be malicious in some cases	If over 50% of packets are HTTP requests, then the host is regarded as 'a sender of many HTTP requests' and the event is classified into the Warning category
OK	The host is legitimate	If more than 50% of packets are from source port 80, then the host is regarded as 'a web server' and the event is classified into the OK category
Special	The host is a server or client of a specific application such as DNS, FTP, mail, or P2P	If more than 50% of packets are from source port 53, then the host is regarded as 'a DNS server' and the event is classified into the Special category If (a) higher source and destination ports account for more than 50% of total port usage, (b) the most dominant host-to-host traffic accounts for less than 30% in the number of packets, and (c) the host sends packets to more than 10 destinations, then the host is regarded as 'a P2P application user' classified into the Special category
Unknown	The host is not classified into any of these categories	If the host cannot be classified into any of these categories, then the event is classified into the Unknown category (most of them related to P2P by BLINC validation. See also Appendix B)

Table 1. Categories and examples of heuristics. Attack and Victim events are regarded as Anomalous category, whereas events of the other four categories are referred to as Normal category. All heuristics to identify Attack events are listed in Appendix A

1. The initial optimal parameter is empirically set to $\theta_\tau(0) = 1.7$. This value is used only during the beginning of parameter learning, i.e. we do not have τ traces in the past for learning (i.e. while $t < \tau$), and once $\theta_\tau(t)$ is determined then the initial value is no longer used. In our experiment, $t = 0$ is the date 1 January 2001, $t = 1$ is 2 January 2001, and so on.
2. Run the anomaly detector on the past τ traces while changing $\theta_\tau(t)$ (increase it by $\delta = 0.1$ from 0.0) to find a (local) maximum.
3. Compute the performance of the past detection $a(\theta_\tau(t))$ for each $\theta_\tau(t)$ as $a(\theta_\tau(t)) = \frac{A}{A+N}$, where A and N are the total counts of Anomalous and Normal events detected with the threshold $\theta_\tau(t)$.
4. Find the optimal threshold for date $(t + 1)$, which produces the (local) maximum in performance: $\theta_\tau^{\text{opt}}(t+1) = \arg\max_\theta a(\theta_\alpha(t))$ that satisfies $a(\theta_\tau(t)) - \delta) < a(\theta_\tau(t))$ and $a(\theta_\tau(t)) + \delta) < a(\theta_\tau(t))$. δ is also empirically determined, and the selection of δ has a trade-off between precision of optimization and possibility of failing in finding the optimal value; for example, $\varepsilon << \delta$ leads to $a(\theta_\tau(t)) = a(\theta_\tau(t) + \varepsilon)$ in most cases, and it fails to find appropriate θ_τ^{opt} satisfying the above condition.

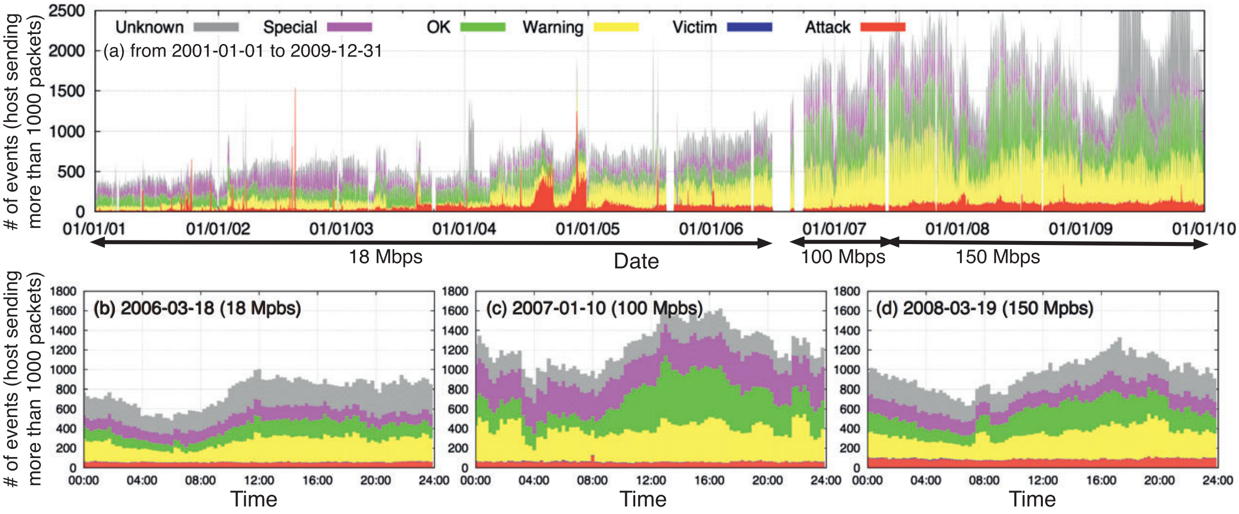
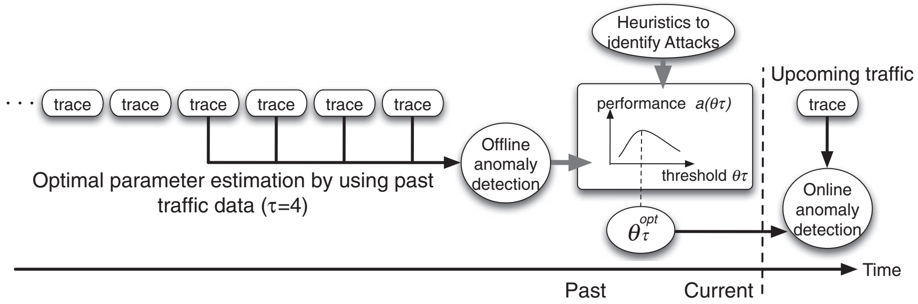


Figure 1. Breakdown of MAWI dataset: (a) 9-year 15 min traces, and 24 h traces on (b) 3 March 2006, (c) 10 January 2007, and (d) 19 March 2008. An event is defined as a source host that sends over 1000 packets per trace



Heuristics to identify Attacks
Upcoming traffic
· · · trace
trace
trace
trace
trace
trace
trace
performance $a(\theta\tau)$
threshold $\theta\tau$
Offline anomaly detection
Online anomaly detection
Optimal parameter estimation by using past traffic data (τ=4)
θ_τ^{opt}
Time
Past
Current

Figure 2. Method of parameter learning. Optimal parameter $\theta_\tau^{\mathrm{opt}}$ for upcoming traffic is estimated by using past τ traffic traces; $\theta_\tau^{\mathrm{opt}}$ maximizes the detection performance of the past τ traces ($\tau = 4$ in this figure)

5. If there is no $\theta_\tau(t)$ satisfying the above condition to determine $\theta_\tau^{\mathrm{opt}}(t+1)$, $\theta_\tau^{\mathrm{opt}}(t)$ for the previous traffic trace is used for detecting anomalies in upcoming traffic: $\theta_\tau^{\mathrm{opt}}(t+1) = \theta_\tau^{\mathrm{opt}}(t)$. The same policy is used if there are no traffic traces of corresponding τ dates.

Anomalies are detected separately for each 15 min traffic trace, and A is the total number of detected Anomalous events over the past τ days (the same for N), whereas the consecutive 24 h traces are concatenated into one trace in order to avoid recounting an anomaly event existing in more than one traces. We define the learning with 15 min and 24 h traces as *15 min* and *24 h learning*.

This procedure for parameter learning is not specific to the anomaly detector; learning is an optimization problem (i.e. finds the best solution that maximizes a function), and it can be applied to other parameters of different anomaly detectors. Also, this procedure is applicable to optimize multiple parameters; that is, the learning is a kind of combinatorial optimization. Searching optimal parameters is enhanced by traditional methods such as hill climbing, simulated annealing, or genetic algorithm.

The performance $a(\theta_\tau(\cdot))$ can be interpreted as *detection efficiency*—the percentage of the number of detected Anomalous events over that of total detected events—and $\theta_\tau^{\mathrm{opt}}(\cdot)$ maximizes this. In this sense,

DOI: 10.1002/nem

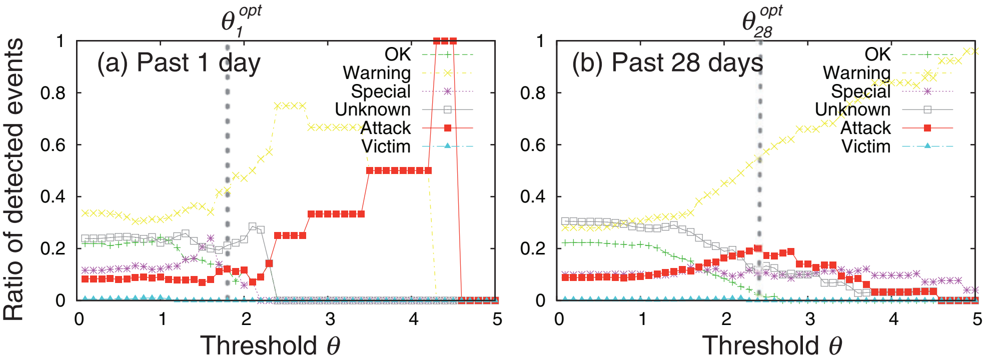


Figure 3. Example of determining optimal parameter θ_τ^{opt}: (a) $\tau = 1$ day of learning; (b) $\tau = 28$ days of learning. Detected events are classified into six categories by the pseudo ground truth generator based on our heuristics. The detection performance is the summation of Attack (red) and Victim (blue)

detected Normal events can be interpreted as false alarms. Although this metric is different from the common performance metrics (i.e. the false positive rate or the false negative rate), $a(\theta_\tau)$ is more suitable for determining $\theta_\tau^{\text{opt}}(\cdot)$; these common metrics must take into consideration all undetected events, but we may not necessarily be able to identify them. Also, since no classification method (ground truth generator) is perfect, we must consider misclassified events (e.g. some of the Unknown events are actually Anomalous). Both the denominator and numerator used by the two common metrics are affected by misclassifications, whereas the denominator of $a(\theta_\tau(\cdot))$ is not affected because $A + N$ is the total number of detected events. Another reasonable choice of performance metric might be to remove Unknown events from denominator and numerator from the common two metrics, but since we manually found most of the Unknown events are P2P, this removal of Unknown would lead to overestimation of performance. Another way to quantify performance is scoring detection results according to abnormality of detected events (1 point for Attack, −1 point for OK, and so on). However, since it is generally difficult to set reasonable weights of abnormality, we focus only on Anomalous events, which should be detected first. Therefore, $a(\theta_\tau(\cdot))$ is more suited to computing θ_τ^{opt}.

Figure 3 shows an example of parameter determination. The x-axis is the value of θ_τ and the y-axis plots the fraction of detected events of a category over the total number of detected events. Figure 3(a, b) shows the detection results of past $\tau = 1$ and $\tau = 28$ days since 6 December 2005. The performance $a(\theta_\tau)$ is the fraction of Attack (red) and Victim (blue) events over the total number of detected events. Since no Victim events were detected on this day, $a(\theta_\tau)$ is identical to that plotted by the red line.

- The best performance in Figure 3(a) is provided by $\theta_\tau(t) \in [4.4, 4.6]$, because $a(\theta_\tau(t)) = 1.0$ (i.e. 100%). However, these parameters are not reasonable due to the few detected events (only one event is detected and many Anomalous events have been missed). When only a few events are detected, $a(\theta_\tau(t))$ exhibits the step-like curve as shown in Figure 3(a) (i.e. this means $a(\theta_\tau(\cdot)) = a(\theta_\tau(\cdot) + \delta)$ and does not satisfy the condition in step 4). Hence choosing a local maximum is better for retaining a reasonable number of detected events. Here, we have one local maximum at $\theta_\tau(t) = 1.8$, and four Anomalous events are detected, and thus $\theta_\tau^{\text{opt}}(t+1) = 1.8$. Conversely, only one Anomalous event is detected for $\theta_\tau > 2.0$.
- The $\theta_\tau = 2.4$ in Figure 3(b), on the other hand, is optimal according to our heuristics. Thus a large amount of learning data makes it easier to determine a good θ_τ^{opt}, and $\theta_\tau^{\text{opt}}(t+1) = 2.4$.

Thus far, we have found that the typical longitudinal pattern of events detected by using a tuned θ_α is continuous, such as that in Figure 4(a). However, there are few spiky traffic patterns such as that in Figure 4(b), while a tuned θ_β would detect more. Since spiky traffic is easily detected unlike continuous traffic, this finding also supports our motivation to discuss θ_α prior to θ_β.

DOI: 10.1002/nem

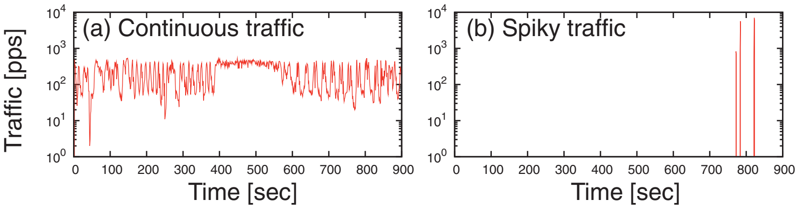


Figure 4. Example of anomalous traffic detected by tuned θ_α: (a) typical traffic and (b) atypical traffic. Both types of traffic are judged to be 'SYN flooding attacks'

Intuitively (but not analyzed with real traces), τ has a trade-off:

- When τ is lower, $\theta_\tau^{\mathrm{opt}}$ cannot be determined because there are few data points to obtain appropriate statistics.
- When τ is higher, $\theta_\tau^{\mathrm{opt}}$ is not suitable because a longer τ causes unexpected delays, affecting the ability to follow macroscopic changes in anomalies.

Thus we need to find an appropriate value for τ.

To evaluate the effect of parameter learning on detection performance, we compare detection results of predicted threshold $\theta_\tau^{\mathrm{opt}}(t)$ and the ideal threshold $\theta_0^{\mathrm{opt}}(t)$ which leads to the best performance for detecting anomalies in upcoming traffic (we use $\tau = 0$ to explain this ideal parameter). We define $\theta_0^{\mathrm{opt}}(t)$ as maximizing the detection performance of upcoming traffic on date t $\theta_0^{\mathrm{opt}}(t) = \arg\max a(\theta_0(t))$, whereas $\theta_\tau^{\mathrm{opt}}(t)$ maximizes the detection performance of past τ traces on dates $t-1, \ldots, t-\tau$. We will refer to $\theta_\tau^{\mathrm{opt}}(\cdot)$ as $\theta_\tau(\cdot)$, because we will only discuss the changes in the optimal threshold, and we basically omit '(t)' from $\theta_\tau(t)$ if t is not needed to discuss the results.

3. EVALUATION

To discuss the automatic parameter tuning, we study the changes in optimal parameters θ_τ (Sections 3.1 and 3.2), an appropriate τ, and the performance degradation in anomaly detection caused by introducing parameter learning (Sections 3.3 and 3.4), the predictability of the ideal parameters (Section 3.5), and the difference between 15 min and 24 h learning (Sections 3.6 and 3.7).

3.1 Changes in values of optimal parameter and number of detected events

First, we assess changes in the value of optimal parameter θ_τ for 15 min learning with some fixed learning periods τ. Figure 5 plots the transition in θ_τ from the beginning of 2001 to the end of 2009. The upper figure depicts microscopic (day-to-day) changes in θ_0 (the ideal parameters), while the lower figure plots macroscopic (monthly) changes in θ_0 (ideal) with the red line, θ_3 (3 days of learning) with the green line, and θ_{28} (28 days of learning) with the blue line, plotting the averages and standard errors for each month. The x-axes plot the dates, and the y-axes have the values of θ_τ in both figures. The upper figure indicates that the ideal parameter θ_0 fluctuates day by day around $\theta_0 \approx 1.5$. The average and standard deviation of θ_0 for 9 years are 1.43 and 0.57. The fluctuations in θ_0 emphasize how important it is to dynamically tune parameters for statistics-based anomaly detectors, because θ_0 is not stable over time. On the other hand, the lower figure shows that the parameter prediction with a higher τ (blue) leads to a larger deviation in the value of θ_τ from the ideal parameter (red) than that with a lower τ (green). Since Pearson's correlation coefficient between θ_0 and θ_3 is 0.80, but that between θ_0 and θ_{28} is 0.35, a higher τ fails to follow the changes in θ_0. A significant case is the change in θ_τ from April 2003 to May 2003. The θ_0 decreases from 1.54 to 1.25, but θ_{28} increases from 1.52 to 2.91, whereas θ_3 varies from 1.80 to 1.82. The high value of θ_{28} during May 2003 derives from $\theta_\tau = 3.9$ from 14 May 2003

DOI: 10.1002/nem

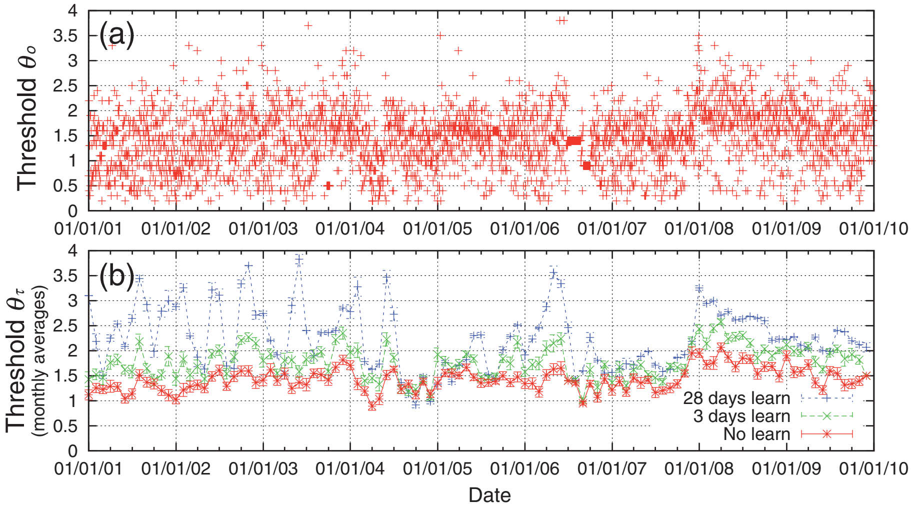


Figure 5. Changes in θ_τ for 15 min learning: (a) microscopic (daily) changes in ideal threshold θ_0; (b) monthly averages in θ_τ for $\tau = 0$, $\tau = 3$, and $\tau = 28$ (past 3 or 28 days of learning). The ideal threshold θ_0 fluctuates daily, and longer learning period τ leads to higher and more unstable θ_τ

to 22 May 2003. By manually checking this cause, we found that the data from 24 April 2003 and 12 May 2003 had an influence on θ (one Anomalous event is detected with $\theta_{28} = 3.9$ for the traces of 24 April 2003 and 12 May 2003, respectively). The data from 12 May 2003 triggered the $\theta_{28} = 3.9$ from 14 May 2003, and the $\theta_{28} = 3.9$ rapidly decreased when the date of 14 April 2003 was beyond the range of the past τ days. This case study also points out the inefficiency of a longer τ. A higher τ, on the other hand, yields a higher and more fluctuating θ_τ over 9 years, but the standard deviation of each plot (i.e. the variance within a month) is small. The averages and standard deviations of the plots are 1.42 ± 0.23 for θ_0, 1.78 ± 0.32 for θ_3, and 2.26 ± 0.62 for θ_{28}. A plausible reason for the inappropriateness of a longer τ is that the prediction of optimal θ_τ can be affected by significant events during the learning period, and these events will fix θ_τ to a certain value.

Figure 6(a–c) highlights the microscopic (day-to-day) changes in θ_0, θ_3, and θ_{28} (from November 2005 to April 2006), and Figure 6(d–f) presents the results of anomaly detection with the thresholds determined from parts (a–c), plotting the number of detected Anomalous and Normal events. The x-axes are the dates, the y-axes in the upper figures are the values of θ_τ, and the y-axes in the lower figures are the values of detected events (red: Anomalous events; green: Normal events). Obviously, a higher τ is inappropriate for estimating the optimal threshold, because θ_{28} stably remains higher and there are fewer detected events than those with θ_0. The stable values of θ_{28} also reminds us of the influence of significant events on determining θ_τ. In other words, the parameter estimation with a longer τ cannot follow the changes in anomalous traffic. Thus learning with a longer τ is inappropriate for obtaining a suitable θ_τ to detect anomalies in practice, and we need to find an appropriate τ.

Consequently, the ideal threshold θ_0 fluctuates daily, and a longer learning period (higher τ) produces higher and more stable θ_τ, which decreases the number of detected events. The next section discusses our further investigations into the reason for worsened performance caused by a longer learning period.

3.2 Variable learning period

We study the strong influence of specific events on the estimates of optimal threshold θ_τ by computing the best performance with *unfixed* τ among [0,28]. We choose an optimal learning period of

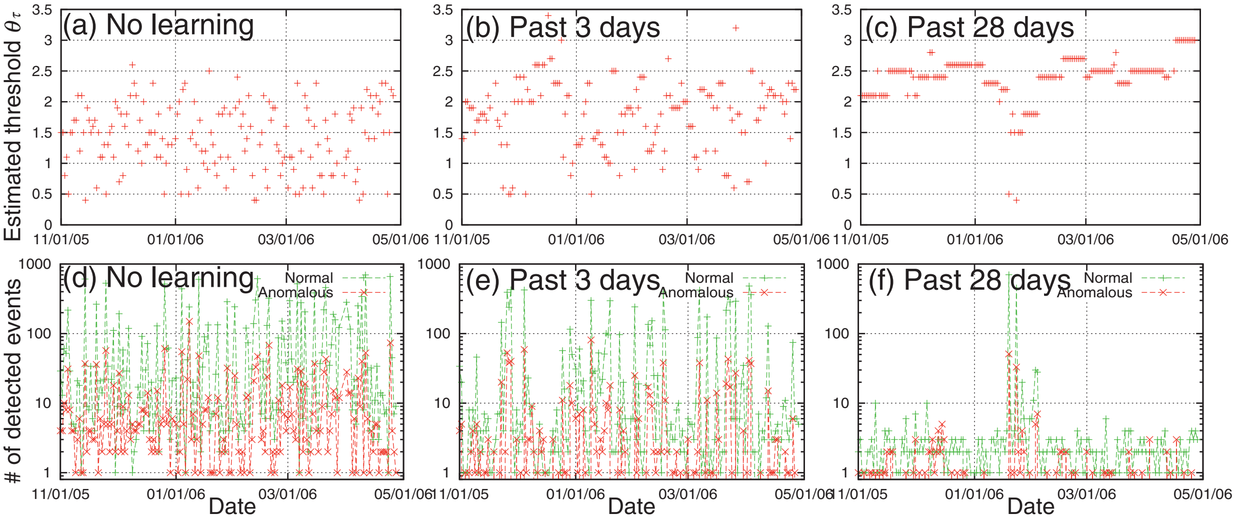


Figure 6. Changes in θ_τ and the number of detected events (from November 2005 to April 2006) for 15 min learning: (a, d) no learning ($\tau = 0$), i.e. the ideal threshold; (b, e) past $\tau = 3$ days of learning; (c, f) past $\tau = 28$ days of learning. Longer learning period provides high and continuous θ_τ, which leads to lower number of detected events, and this learning misses dynamics of traffic anomalies

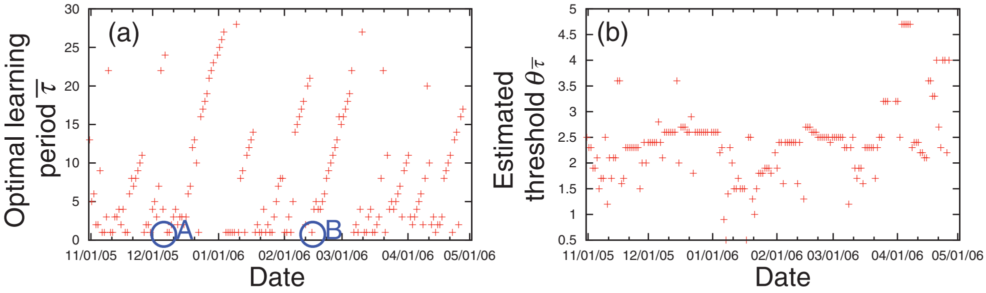


Figure 7. Changes in (a) *unfixed* optimal learning period $\bar{\tau}$ (from 1 to 28) and (b) corresponding threshold $\theta_{\bar{\tau}}$. Learning is highly affected by significant days' data, because the best performance is provided by increasing τ (slanted lines in (a)), which is tuned to include influential dates such as A or B. Correspondingly, θ_τ is fixed to a certain value by significant events, so τ should be appropriately fixed to avoid such effect of significant events

$\bar{\tau} = \arg\max a(\theta_\tau)$, and investigate changes in the values of $\bar{\tau}$ and $\theta_{\bar{\tau}}$. Figure 7 plots the results with the data from November 2005 to April 2006. The *x*-axes are the dates, the *y*-axis in Figure 7(a) is $\bar{\tau}$, and the *y*-axis in Figure 7(b) is $\theta_{\bar{\tau}}$ derived from Figure 7(a). In Figure 7(a) there are several slanted lines with a slope of one, which means that some data had a strong influence on determining θ_τ during the period of τ days that included these influential traces (until τ reaches 28, i.e. the upper bound of our experiment). To investigate the reasons for this effect, we inspect labels A and B, which are two of the roots of these lines.

A: Data for 9 September 2005 (label A) set $\theta_{\bar{\tau}} = 2.6$. The detected events on the data consist of five Attacks.

B: The combination of data from 18 and 19 December 2006 (label B) sets $\theta_{\bar{\tau}} = 2.5$. This threshold detected three Attacks, four Warnings, and one Unknown.

DOI: 10.1002/nem

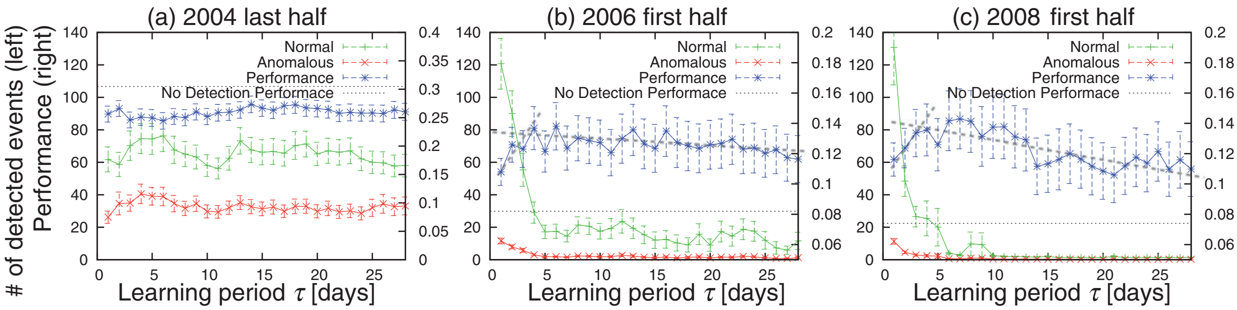


Figure 8. Evaluation of τ for 15 min learning: (a) last half of 2004; (b) first half of 2006; (c) first half of 2008. Appropriate τ is around 3 days except for period of worm (a), considering the balance between the number of detected events and detection performance. 'No Detection Performance' indicates the fraction of Attack events over all events in analyzed traffic

Since such events have a strong effect on $a(\theta_\tau)$, variable τ is inappropriate to follow the macroscopic changes in Internet traffic. In addition, the changes in $\theta_{\bar{\tau}}$ for variable τ (Figure 7(b)) are almost the same as those for $\tau = 28$ (Figure 6(f)), and the performance with $\bar{\tau}$ is lower than that with $\tau = 3$. Thus τ must be appropriately fixed, and we discuss our evaluation of this in the following section.

3.3 *Optimal learning period*

Now let us discuss the most appropriate learning period for setting the automatic and dynamic thresholds. Figure 8 summarizes the detection results and performance as a function of learning period τ with different periods of data: (a) the last half of 2004; (b) the first half of 2006; and (c) the first half of 2008. The x-axes are τ (from 1 to 28 days), the left-hand y-axes plot the number of events detected with θ_τ (red: Anomalous; green: Normal), and the right-hand y-axes depict the detection performance $a(\theta_\tau)$ (blue). Each plot shows the averages and their standard errors in the 15 min detection results for a half year. In the last half of 2004 there was a massive outbreak of a worm lasting until the end of this year. This caused the τ-independent result shown in Figure 8(a); any τ retains certain values for the number of detected events and performance. This should derive from the long-term dominance of only one kind of anomaly. In contrast, for other periods (Figure 8(b, c)), we can confirm the trade-off in τ.

- The red and green lines indicate that a longer τ yields to fewer detected events than a shorter τ. This is because a longer τ results in high θ_τ (Figure 6), and a higher θ_τ leads to fewer detected events.
- The blue line explains that a shorter τ (1 or 2 days) leads to worsened performance; i.e. a shorter τ cannot obtain a sufficient amount of data to appropriately determine θ_τ.

The increasing value of performance for shorter τ means that the decrease in the number of detected Normal events is larger than that of Anomalous events; that is, parameter learning with appropriate τ can follow the trends in anomalies. From these results, we can conclude that a τ of around 3 is empirically the most appropriate because $a(\theta_\tau)$ is larger and the number of detected Anomalous events is also higher in our dataset. In addition, we compare the performance $a(\theta_\tau)$ with the percentage of Anomalous events in the original dataset—(a) 30.5%, (b) 8.2%, and (c) 7.4%—which are indicated by the gray lines in the figures. Increase in $a(\theta_\tau)$ with respect to the percentage for any τ in Figure 8(b, c) indicates the algorithm's efficiency. Conversely, in Figure 8(a), the scanning activities of the worm form statistical referential behavior, so that they spoil the anomaly detection method.

3.4 *Performance degradation caused by introducing parameter learning*

Next, we compare the performance of 15 min learning with $\tau = 0$ (the ideal threshold) and that with $\tau = 3$ (the most appropriate prediction of θ_τ) in Figure 9. The x-axis is the date, the y-axis in Figure 9(a)

DOI: 10.1002/nem

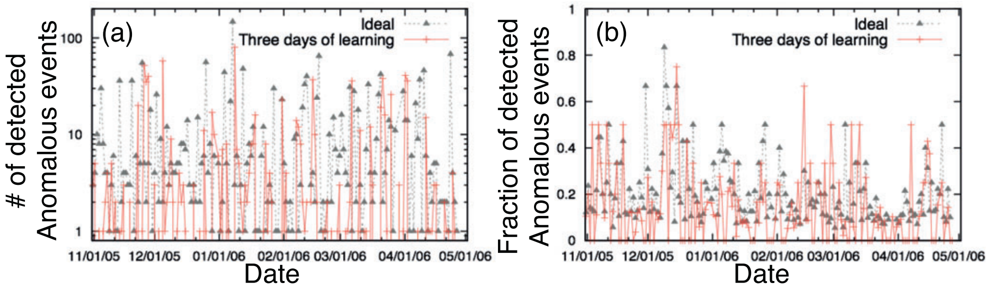


Figure 9. Performance degradation caused by introducing parameter learning: (a) number of detected Anomalous events; (b) performance for $\tau = 0$ and $\tau = 3$

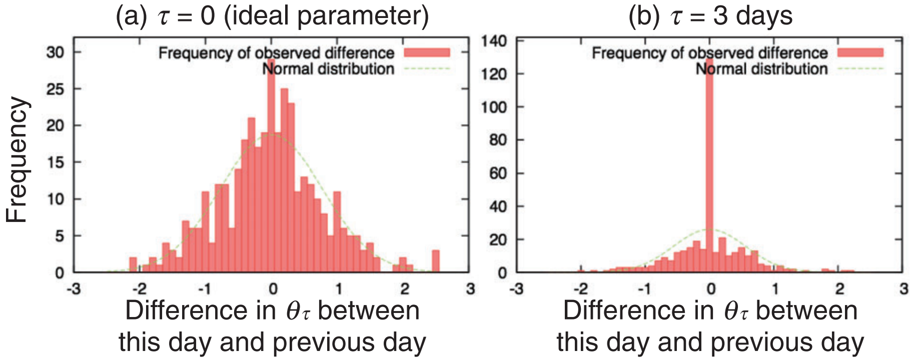


Figure 10. Random process analysis of changes in (a) θ_0 and (b) θ_3 of 15 min learning for 150 Mbps link during 2008 (related to Figure 6(a, b)). θ_0 can be treated as a random process followed by normal distribution

is the number of detected Anomalous events, and the y-axis in Figure 9(b) is the fraction of the number of detected Anomalous events over that of all detected events for each day. In both figures, the gray lines plot the results for $\tau = 0$ (Figure 6(d)), and the red lines plot those for $\tau = 3$. Figure 9(b) represents the degradation in performance of parameter learning; however, the degradation is acceptable for detecting anomalies in practice. The performance with parameter learning worsens by 16.6% on average, and its standard error is 9.2%. Also, learning reduces the number of detected anomalies by about 50% (Figure 9(a)). $\tau = 0$ results in 10.8 ± 1.3 detected Anomalous events, whereas $\tau = 3$ produces 5.4 ± 0.9.

3.5 Predictability of ideal threshold

In addition to evaluating degradation in detection performance, we investigate the predictability of θ_0 for upcoming traffic from the past θ_0 by doing time series analysis. This is also practically important for real-time anomaly detection.

First, we inspect the time correlation of θ_0 for 15 min learning, and Figure 14 highlights the results of (a) autocorrelation $R_{\theta\theta}(t) = \sum_{n=0}^{N-t-1} \frac{(\theta_0(n) - \mu_\theta)(\theta_0(n+t) - \mu_{\theta_0})}{\sigma_{\theta_0}^2}$, and (b) power spectrum $|X(f)|^2 = \left| \sum_{n=0}^{N-1} \theta_0(n) \times \exp\left(-2\pi i \frac{f}{N} n\right) \right|^2$, where $\theta_0(n)$ is θ_0 on date n as shown in Figure 5(a). Here, $\theta_0(0)$ is the estimated threshold of 1 January 2001 ($\theta_0(1)$ is that of 2 January 2001, and so on), μ_{θ_0} and $\sigma_{\theta_0}^2$ are the average and variance of $\theta_0(\cdot)$, π is the circle ratio, and i is an imaginary unit. N is the total number

DOI: 10.1002/nem

of $\theta_0(\cdot)$, i.e. $N = 1826$ days for the sequence from 1 January 2001 to 31 December 2005, so that we can avoid the long-term lack of data in 2006. These two figures reveal that there is no strong periodic correlation in θ_0. Also, the low value of $R_{\theta\theta}(t)$ for lag $t = 1,2,3$ represents almost no time correlations among consecutive $\theta_0(\cdot)$. These results also imply that anomalous traffic does not have significant periodicity, and the optimal learning period $\tau = 3$ shown in the previous section results from the well-balanced amount of data rather than the macroscopic tendencies of anomalies, i.e. shorter learning leads to insufficient amount of information for estimating $\theta_0(\cdot)$, while longer learning is affected by events on significant days, so that it cannot follow fluctuating $\theta_0(\cdot)$. In summary, the nonexistence of periodicity in $\theta_0(\cdot)$ suggests that we have to find alternative methods to predict $\theta_0(\cdot)$ for upcoming traffic.

Second, we investigate how much $\theta_0(\cdot)$ changes daily. Figure 10 displays the distribution of differences in the values of θ_τ (i.e. $\theta_\tau(n) - \theta_\tau(n - 1)$) for (a) $\tau = 0$ and (b) $\tau = 3$ during 2008 (Figure 6(a, b) shows the changes in the values of $\theta_\tau(\cdot)$ for both τ). The x-axes represent the day-to-day differences in $\theta_\tau(\cdot)$, and the y-axes count the frequency of the differences. The histogram in Figure 10(a) is bell-shaped, and we fit it as a normal distribution $f(x) = \frac{1}{\sqrt{2\pi\sigma}}\exp\left(-\frac{(x-\mu)^2}{\sigma^2}\right)$ with $\mu = 0.00$ and $\sigma = 0.78$; that is, the $\theta_0(\cdot)$ for upcoming traffic is predicted to be the same as that for the previous day's traffic with a prediction error of 0.78 × 2 at 95% probability, according to the two-sigma rule. The histogram in Figure 10(b), on the other hand, is more skewed ($\mu = 0.01$ and $\sigma = 0.56$). Considering that a longer τ leads to higher counts in 0, this figure also implies that events on significant days influence the determination of θ_τ. To obtain a better approximation of this model, we may use zero-inflated models, even though the model of the changes in $\theta_3(\cdot)$ makes less practical sense.

In summary, the changes in $\theta_0(\cdot)$ have no temporal correlation, and represent a random process followed by a normal distribution (i.e. we observed few significant jumps in the changes in $\theta_0(\cdot)$).

3.6 Parameter learning with consecutive 24 h traffic traces

We evaluate 24 h learning, i.e. parameter learning with the consecutive 24 h traces measured on the same link (also stored in the same repository as 15 min traces). The same set of parameters for 15 min learning is applied to that of 24 h learning. However, this 24 h learning is slightly different from the 15 min one in that the anomaly detection method is run for consecutive traffic traces to avoid repeatedly recounting an anomaly event included in more than one trace. Therefore, as the number of events detected with 24 h data will be fewer than that with 15 min data, the amount of data required should be larger.

Figure 11(a–c) highlights the changes in the value of θ_τ for (a) $\tau = 0$, i.e. the ideal threshold, (b) $\tau = 4$, i.e. 1 h of learning, and (c) $\tau = 12$, i.e. 3 h of learning. Also, Figure 11(d–f) shows the changes in the number of Anomalous and Normal events detected with θ_τ in the above figures. The x-axes plot the dates, the y-axes in the upper figures are the values of θ_τ, and the y-axes of the lower figures are the number of detected events. The fluctuations in θ_τ and the number of detected events with 24 h traces are similar to those with 15 min traces (Figure 6). Also, a longer τ leads to higher and continuous values for θ_τ, which results in fewer detected events. The averages and standard deviations of optimal thresholds θ_τ are (a) 1.43 ± 0.49 for $\tau = 0$ (ideal), (b) 1.74 ± 0.59 for τ of 1 h, and (c) 2.45 ± 0.59 for τ of 3 h. In addition, the averages and standard errors for the number of detected events are (d) 69.2 ± 13.9 for Normal and 5.8 ± 1.1 for Anomalous, (e) 53.0 ± 16.8 for Normal and 4.5 ± 1.3 for Anomalous, and (f) 4.6 ± 1.6 for Normal and 0.6 ± 0.2 for Anomalous. This also suggests that an appropriate value for τ can be chosen to efficiently detect anomalies on an hourly scale.

Figure 12 plots the performance as a function of τ with the data from (a) 3 March 2006, (b) 10 January 2007, and (c) 19 March 2008. For each figure, the x-axis represents the value for τ, the left-hand y-axis plots the number of detected Anomalous (red) and Normal (green) events, and the right-hand y-axis shows the performance (blue). Both y-axes plot the average values with standard errors in 24 h traces. These figures reveal that the number of detected events decreases with the increase in τ, and appropriate learning periods are (a) $\tau = 7$, i.e. 1.75 h, (b) $\tau = 6$, i.e.1.5 h, and (c) $\tau = 3$, i.e. 0.75 h,

DOI: 10.1002/nem

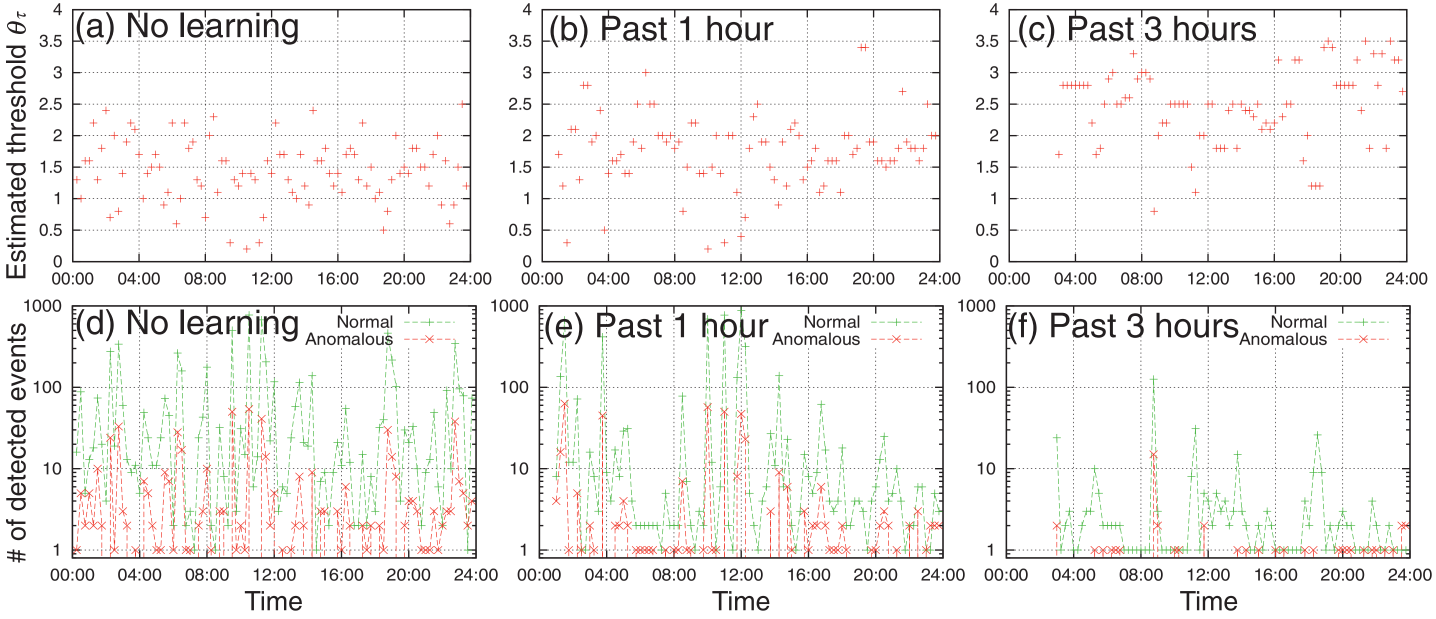


Figure 11. Changes in θ_τ for consecutive 24 h traffic data of 18 March 2006: (a) no learning, i.e. ideal threshold; (b) past 1 h of learning; (c) past 3 h of learning. Similar to 15 min learning, higher τ leads to high and continuous θ_τ, decreasing the number of detected events

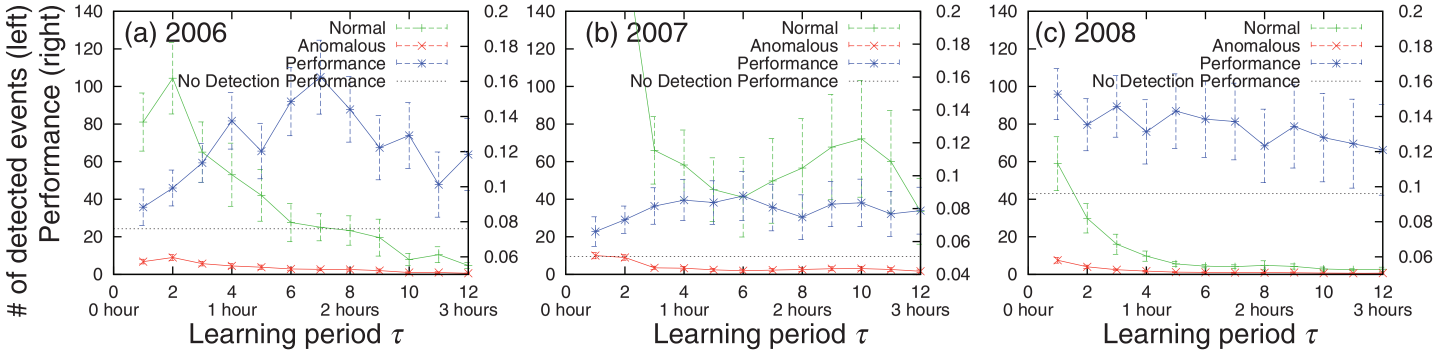


Figure 12. Evaluation of τ with 24 h consecutive traces: (a) 3 March 2006; (b) 10 January 2007; (c) 19 March 2008. 'No Detection Performance' indicates the fraction of Attack events over all events in analyzed traffic. Appropriate τ are around 1.75 h for (a), 1.5 h for (b), and 0.75 h for (c), considering the balance between the number of detected events and the performance

DOI: 10.1002/nem

considering both the performance and the number of detected events. These τ balance the trade-off between the amount of information to determine θ_τ versus the influence of significant events on parameter learning. In addition, the average percentages of Anomalous events for each trace in the 24 h traces without detection are (a) 7.6%, (b) 5.1%, and (c) 9.6%, as plotted by the gray lines in the figures. These figures also indicate the algorithm's efficiency because any increase in the percentage of detected Anomalous events resulted from the anomaly detector.

Therefore, θ_τ also fluctuates on an hourly scale, and an optimal θ_τ should be determined with a well-balanced number of past traces (e.g. about 1.5 h). We note that θ on a minute scale cannot be discussed, because a few minutes of traffic have too little information to detect statistical outliers.

3.7 Performance comparison between 15 min and 24 h learning

Finally, we study the difference between 15 min and 24 h learning. Since the datasets for both types of learning have overlap traces starting from 14:00 on the dates that 24 h measurement were conducted, these two types can be compared by using past τ traces since 14:00 on these dates. Figure 13 compares the two types of learning with three microscopic (daily and hourly) examples: (a) 3 March 2006; (b) 10 January 2007, and (c) 19 March 2008. All the x-axes are the learning periods τ ($\tau = 1$ means 1 day for 15 min learning and 15 min for 24 h learning), the y-axes of the upper figures plot the values for θ_τ, those of the middle figures show the number of detected Anomalous and Normal events, and those of the lower figures highlight the detection performance $a(\theta_\tau)$.

The upper figures show two similar lines of 15 min and 24 h learning on the same date, though the lines of 15 min learning vary among the three dates (also for 24 h learning). A possible reason for these similar lines is that the characteristics of Internet traffic anomalies fluctuate on at least an hourly scale, so the results for 15 min and 24 h learning do not present significant differences. Also, the learning method can be applied on both a daily scale and an hourly scale. Additionally, the middle figures indicate that the number of events detected with 15 min learning is higher than that with 24 h learning for (b) and (c), so 15 min learning can macroscopically (roughly) capture changes in anomalies. However, 15 min learning cannot detect any Anomalous events in case (a). On the other hand, the lower figures also reflect the optimal τ of $\tau = 3$ for 15 min learning, and $\tau = 6{,}7$, and 8 for 24 h learning, considering the balance between the number of detected events and the detection performance. 24 h learning provides better performance than 15 min learning on average, so 24 h learning can capture microscopic (subtle) changes in specific anomalies.

Here we present an example of events detected by θ_3 for 15 min learning and θ_8 for 24 h learning on 19 March 2008. For upcoming traffic, 15 min learning detected three Anomalous (two continuous and one spiky) and six Normal (one continuous and five spiky) events, whereas 24 h learning found one Anomalous (spiky) and one Normal (continuous) event. For learning traffic, 15 min learning extracted six Anomalous (four continuous and two spiky) and 14 Normal (12 continuous and two spiky) events, while 24 h learning uncovered six Anomalous (six spiky) and eight Normal (eight continuous) events. The detector with 15 min learning was tuned to detect continuous Anomalous events, because those kinds of events were dominant in the learning traces. On the other hand, the detector with 24 h learning was tuned to find spiky Anomalous events due to the dominance of spiky anomalies in the learning data. Hence the learning method could capture the typical behavior of Anomalous events.

Therefore, the characteristics of 15 min learning and 24 h learning are similar, so the parameter tuning method can be applied to both daily and hourly scales.

4. DISCUSSION

4.1 Trends in traffic usage and changes in optimal parameter

The MAWI link has been upgraded twice and the application breakdown is evolving (Figure 1), but changes in optimal parameter (θ_0^{opt}) do not follow the upgrade or evolution, unlike other observations

DOI: 10.1002/nem

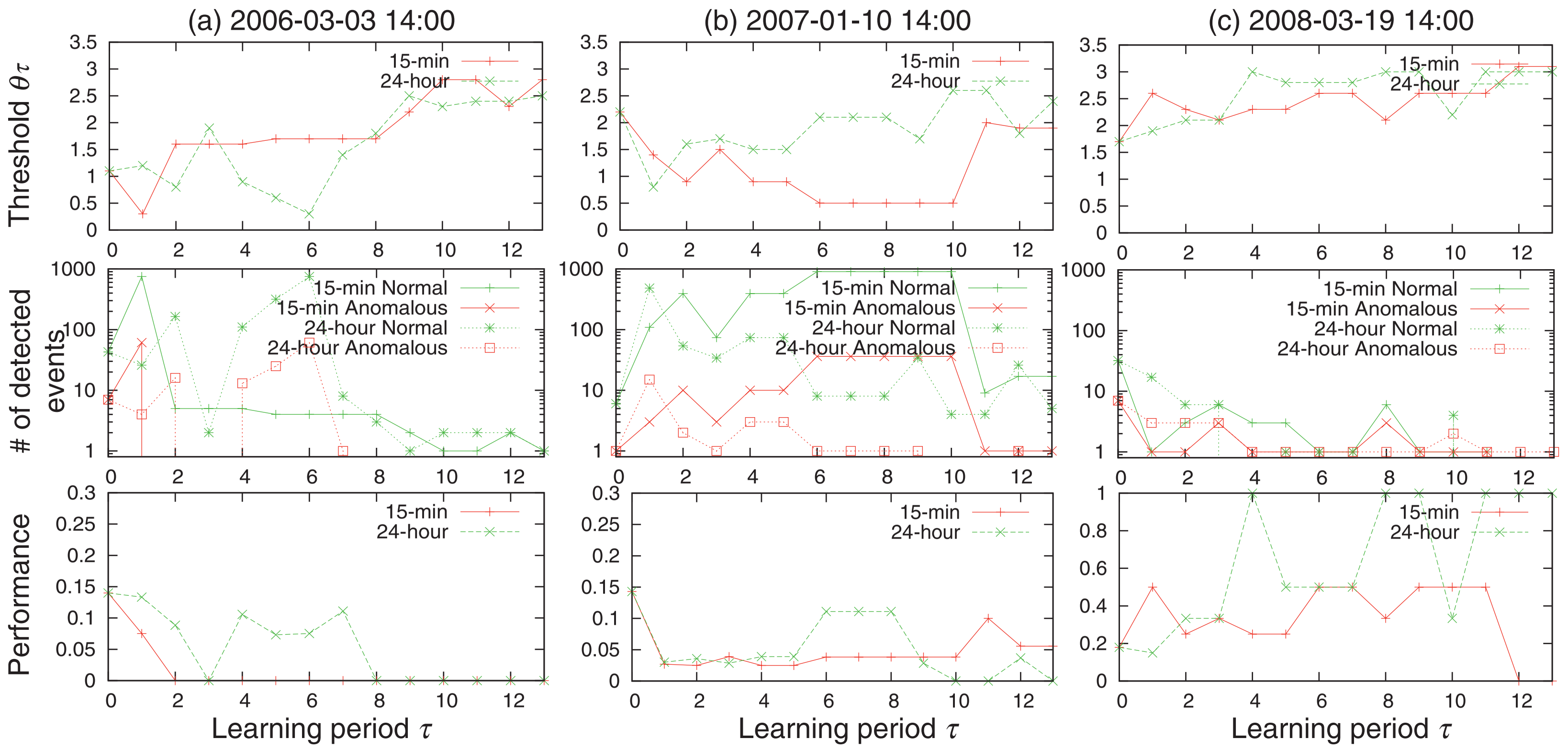


Figure 13. 15 min versus 24 h learning. $\tau = 1$ represents one trace (1 day for 15 min learning and 0.25 h for 24 h learning). Characteristics of 15 min and 24 h learning are similar on a certain date, though the lines of 15 min learning vary among the three dates (also for 24 h learning). Thus the parameter-tuning method can be applied to both daily and hourly scales, and it is favorable to macroscopically (e.g. yearly) update the appropriate τ

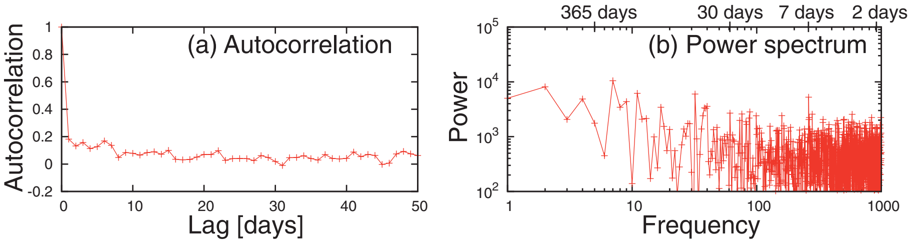


Figure 14. Frequency analysis of changes in θ_0 of 15 min learning for 18 Mbps link (related to Figure 6(a)). Changes in θ_0 have no strong frequency, i.e. there is no typical periodicity in θ_0

in the 3G network [14]. Since threshold θ_α indicates the degree of normalized deviation from referential statistics, the changes in $\theta_\alpha^{\text{opt}}$ should be affected by anomalies changing day by day. This explanation can be supported by the trends in referential behavior (unnormalized average α and β) shown in Himura *et al*. [22]. The average values of α and β evolve according to traffic breakdown and link upgrades.

4.2 High degree of variability in optimal parameter

Our results illustrate the importance of dynamic parameter tuning in statistical methods for detecting anomalies so that they can be deployed in the real Internet. An inappropriate parameter significantly degrades detection performance, because the performance is not constant relative to the value of the parameter. In addition, even if we set an optimal parameter at a certain time, we cannot use the same value consistently, because the optimal parameter is not invariable, as shown in Figures 5, 6, and 11. This feature is not specific to our method. Since Internet traffic includes the trade-off, other methods are also likely to require parameter tuning at daily and/or hourly levels so that they can follow the macroscopic behavior of network anomalies. Also, our results show that supervised learning-based anomaly detectors must consider their training dataset (e.g. the amount of traces).

4.3 Typical timescale of anomalous traffic

The comparison between 15 min and 24 h learning (Figure 13) points out the influence of the number of events on parameter learning, so the microscopic trends in anomalous traffic on hourly and daily scales are weak. Also, the weakness of the trends in anomalous traffic can be confirmed by the highly variable θ_0 (Figure 5(a)) and the nonexistence of periodicity of θ_0 (Figure 14). On the other hand, Figure 5(b) plots the transition in typical θ_0 for each month, so fluctuating θ_0 should form macroscopically standard anomalous behavior on a large scale (e.g. monthly levels), but higher τ cannot capture this trend because it is affected by significant events. Consequently, the trends of anomalous traffic are weak on each timescale from the viewpoint of parameter learning, and the number of events and existence of significant events have stronger influence on determining optimal parameters.

4.4 Practical use of parameter learning

To use parameter learning in real situations, one can predict an optimal threshold for upcoming traffic by evaluating the detection performance of the past τ traces before detecting anomalies in upcoming traffic. The τ can be set to 3 if one captures snapshots of a certain time on each day, and can be set to around 1.5 h if one measures consecutive traces. Also, 15 min and 24 h learning can be combined to

DOI: 10.1002/nem

predict an optimal threshold, and this prediction will be more robust in detecting anomalies, because the combination of the two types of learning considers both subtle and rough changes in anomalous behavior of Internet traffic.

4.5 Advantage of using multi-scale gamma model

Figures 5, 6, and 11 indicate that the ideal parameter θ_0 is quite scattered, so that the macroscopic behavior of the anomalous traffic exhibits no typical patterns. In addition, since α determines the shape of the histogram for the number of packets on a certain timescale, the multi-scale gamma model can follow the fluctuating traffic patterns; otherwise θ_α would be constant. Also, Figures 8 and 12 indicate the efficiency of the detection algorithm, as previously discussed. Therefore, the anomaly detector based on the multi-scale gamma model is a promising approach.

4.6 Events of strong effect on parameter prediction

Figures 6, 7, 8, and 11 suggest that parameter learning is strongly affected by significant traces. Such a trace provides high performance $a(\theta)$ for a certain θ, plausibly because of high number of detected Anomalous events with higher thresholds, etc. One way to avoid the influence of specific data is to use time series forecasting methods, e.g. to impose moving weights on the past data, while another way is to adopt a median value of θ_τ among ideal parameters θ_0 of each of past τ traces. We will enumerate possible learning methods and evaluate them in the future.

4.7 Formal expression of determining appropriate learning period τ

In this paper we selected appropriate τ by balancing the detection performance $a(\theta_\tau)$ and the number of detected Attack events. Since the number of detected events can be interpreted as the fraction of undetected (missed) Attack events over total Attack events $b(\theta_\tau)$, one way to formulate the balance is to use weighted harmonic mean between $a(\theta_\tau)$ and $b(\theta_\tau)$: $\dfrac{1}{w\dfrac{1}{\alpha}+(1-w)\dfrac{1}{b}}$ with $w \in [0.1]$. The weight w is a parameter to be determined manually according to whether $a(\theta_\tau)$ (or $b(\theta_\tau)$) should be focused or not, and we can computationally obtain an appropriate τ maximizing the harmonic average.

5. CONCLUSION

We have discussed automatic and dynamic parameter tuning (defined as *parameter learning*) for a statistics-based anomaly detector. The main idea underlying parameter learning was that we predicted an appropriate parameter for upcoming traffic by considering the results of detection over the past several traffic traces. We assessed this learning method by evaluating a statistical anomaly detection method with real traffic traces measured at a trans-Pacific link over 9 years (15 min from 14:00 JST every day and 24 h for some dates) with pseudo ground truth generator validated by BLINC. We also analyzed the predictability of the ideal parameter with respect to periodicity and a random process. Our main findings were as follows. (1) The ideal parameter fluctuates daily. (2) Parameter learning with a longer τ is affected by significant data included in the period, and the appropriate τ is about three traces (days) for learning with daily 15 min traces and around 1.5 h for that with 24 h traces. (3) The performance degradation caused by introducing parameter learning is 17% with $\tau = 3$ for the daily 15 min traces. (4) Even though the changes in the ideal parameter had no periodicity, it could be modeled as a random process followed by a normal distribution. Our contribution was to clarify and quantify the importance of setting dynamic parameters for statistical methods of detecting anomalies in network traffic in the real world; i.e. it is ineffective to use fixed values for parameters. In future

DOI: 10.1002/nem

Category	Explanation	Example of heuristics
Attack	SYN scan/flooding	If SYN flagged packets account for over 20% of all packets sent by a host, then the host is regarded as a 'SYN attacker'. If the attacker targets more than 100 hosts, then the attacker is a scanner, else a SYN flooding attacker
Attack	Ping scan/flooding	If ICMP packets account for more than 20% of traffic, then the sender host is a 'ping attacker'. If the attacker targets more than 50 hosts, then the attacker is recognized as a ping scanner, else ping flooding attacker
Attack	A kind of scanner	If a host targets more than 50 destination hosts, and if the 10th dominant host-to-host traffic consists of fewer than 5 packets, then the source host is a 'scanner'
Attack	Strange TCP traffic	If over 30% of traffic is composed of TCP packets that do not include SYN or ACK flag (e.g. RST only), the sender host is an 'unusual traffic generator'

Table 2. Heuristics to identify Attack events

		Heuristics based on port, flags, and communication pattern												
		Attack			OK	Warning		Special						Unknown
BLINC		scan 620	flood 709	strange 4	web srv 6538	web cli 5457	ssh 645	dns 807	mail 632	p2p 361	ftp 129	other 170	proxy 113	unkn 3262
SCAN	193	191					2							
WEB	10 383	26	30		4414	5243	52		6	68	3	34	3	504
CHAT	24				1					3		7		13
FTP	255									4	53			198
DNS	920	30	49		15	10	34	725	4	1	2	4		46
MAIL	819	3	1		16	8	24		587	7	1		2	170
P2P	1 587	71	58	1	21	38	17	78	1	232	25	6	3	1 036
UNKN	5 266	299	571	3	2071	158	516	4	34	46	45	119	105	1295

Table 3. Validation of classification heuristics with BLINC (row: our heuristics; and column: BLINC). Our heuristics can cover Attack events identified by BLINC, whereas BLINC can reduce Unknown events

work, we intend to extend the evaluation of parameter learning to other learning schemes and other anomaly detection methods with multiple parameters, by using traffic traces collected on other links as well as improving the classification framework.

APPENDIX A: HEURISTICS TO IDENTIFY ATTACK EVENTS

Table 2 lists the rules to identify Attack events. The set of rules is derived from our best knowledge based on port, flag, and communication pattern of traffic, and used in creating pseudo ground truth. The heuristics basically focus on scanning activity, flooding attack, and strange use of TCP traffic.

APPENDIX B: VALIDATION OF CLASSIFICATION HEURISTICS WITH BLINC

We compare our heuristics with Reverse BLINC used in Kim *et al*. [21], whose parameters are set to the default values. Since BLINC is a flow-level traffic classifier, we converted its outputs into host level by finding the most dominant category in a host (except for unknown). Table 3 displays the comparison by using the hosts (events) observed in traces of every 15th from January to December 2008, and the table shows the following:

- Attack events: since BLINC only classified host scans in terms of harmful traffic, there are unclassified events such as SYN flooding attacks and ICMP traffic. Also, our classification results include most of the scan labeled by BLINC (191 of 193).
- Unknown events: BLINC outperforms our heuristics in terms of unknown traffic. Since most events unclassified by our heuristics are P2P and none of them is an attack, BLINC is useful in reducing the fraction of unknown events.

In summary, our Attack heuristics outperform the attack detection part of BLINC rules. BLINC can be used in reducing the amount of unknown events, which are mainly P2P traffic.

ACKNOWLEDGEMENTS

We wish to thank Patrice Abry, Pierre Borgnat, and Guillaume Dewaele of École Normale Supérieure de Lyon, and Romain Fontugne of the Graduate University of Advanced Studies for their invaluable advice. Also, we are greatly thankful to Hyun-chul Kim of Seoul National University, Dhiman Barman of Juniper Networks, and Thomas Karagiannis of Microsoft Research for allowing us to use their implementation of BLINC. This work was partially supported by the Strategic International Cooperative Program of CNRS and JST.

REFERENCES

1. Roesch M. Snort: lightweight intrusion detection for networks. In *USENIX LISA 99*, November 1999; 229–238.
2. Brutlag J. Aberrant behavior detection in time series for network monitoring. In *USENIX LISA 00*, December 2000; 139–146.
3. Barford P, Kline J, Plonka D, Ron A. A signal analysis of network traffic anomalies. In *ACM IMW 02*, November 2002; 71–82.
4. Krishnamurty B, Sen S, Zhang Y, Chen Y. Sketch-based change detection: methods evaluation and applications. In *ACM IMC 03*, October 2003; 234–247.
5. Lakhina A, Crovella M, Diot C. Diagnosing network-wide traffic anomalies. In *ACM SIGCOMM 04*, August 2004; 219–230.

DOI: 10.1002/nem

6. Soule A, Salamatian K, Taft N. Combining filtering and statistical methods for anomaly detection. In *ACM IMC 05*, October 2005; 331–344.
7. Gu Y, McCallum A, Towsley D. Detecting anomalies in network traffic using maximum entropy estimation. In *ACM IMC 05*, October 2005; 345–350.
8. Kim Y, Lau WC, Chuah MC, Chao HJ. PacketScore: a statistics-based packet filtering scheme against distributed denial-of-service. *IEEE Transactions on Dependable and Secure Computing* 2006; **3**(2): 141–155.
9. Dewaele G, Fukuda K, Borgnat P, Abry P, Cho K. Extracting hidden anomalies using Sketch and non Gaussian multiresolution statistical detection procedure. In *ACM SIGCOMM 07: LSAD Workshop*, August 2007; 145–152.
10. Stoeklin MP, Boudec JYL, Kind A. A two-layered anomaly detection technique based on multi-modal flow behavior models. In *PAM 2008*, April 2008; 212–221.
11. Brauckhoff D, Wagner A, Dimitropoulos X, Salamatian K. Anomaly extraction in backbone networks using association rules. In *ACM IMC 09*, November 2009; 28–34.
12. Ringberg H, Soule A, Rexford J, Diot C. Sensitivity of PCA for traffic anomaly detection. In *ACM SIGMETRICS 2007*, June 2007; 109–120.
13. Himura Y, Fukuda K, Cho K, Esaki H. An automatic and dynamic parameter tuning of a statistics-based anomaly detection algorithm. In *IEEE ICC 2009*, June 2009; 6.
14. D Alconzo A, Coluccia A, Ricciato F, Maierhofer PR. A distribution-based approach to anomaly detection for 3G mobile networks. In *IEEE GLOBECOM 2009*, November–December 2009; 8.
15. Karagiannis T, Papagiannaki K, Faloutsos M. BLINC: multilevel traffic classification in the dark. *ACM SIGCOMM 05*, August 2005; 229–240.
16. Scherre A, Larrieu N, Owezarski P, Borgnat P, Abry P. Non-Gaussian and long memory statistical characterisations for Internet traffic with anomalies. *IEEE Transactions on Dependable and Secure Computing* 2007; **4**(1): 56–70.
17. Cho K, Mitsuya K, Kato A. Traffic data repository at the WIDE Project. In *USENIX 2000 FREENIX Track*, June 2000; 263–270.
18. Claffy KC. A day in the life of the Internet: proposed community-wide experiment. In *ACM SIGCOMM Computer Communication Review* 2006; **36**(2): 39–40.
19. Borgnat P, Dewaele G, Fukuda K, Abry P, Cho K. Seven years and one day: sketching the evolution of Internet traffic. In *IEEE INFOCOM 2009*, April 2009; 711–719.
20. Fontugne R, Himura Y, Fukuda K. Evaluation of anomaly detection method based on pattern recognition. *IEICE Transactions on Communications* 2010; **E93-B**(2): 328–335.
21. Kim H, Claffy KC, Fomenkov M, Barman D, Lee MFKY. Internet traffic classification demystified: myths, caveats, and the best practices. In *ACM CoNEXT 2008*, December 2008; 12.
22. Himura Y, Fukuda K, Cho K, Esaki H. Quantifying host-based application traffic with multi-scale gamma model. In *PAM2009 Student Workshop*, April 2009; 2. [Online]. Available: http://pam2009.kaist.ac.kr/workshop_paper/yourconf1-paper12.pdf [11 July 2010].

AUTHORS' BIOGRAPHIES

Yosuke Himura is a master course student in Department of Information and Communication Engineering, Graduate School of Information Science and Technology, the University of Tokyo. His research interests are Internet traffic analysis and Internet security.

Kensuke Fukuda is an associate professor at the National Institute of Informatics (NII) and is a researcher, PRESTO, JST. He received his Ph.D degree in computer science from Keio University at 1999. He worked in NTT laboratories from 1999 to 2005, and joined NII in 2006. His current research interests are Internet traffic measurement and analysis, intelligent network control architectures, and the scientific aspects of networks. In 2002, he was a visiting scholar at Boston University.

Kenjiro Cho is Deputy Research Director at Internet Initiative Japan, Inc. He received the B.S. degree in electronic engineering from Kobe University, the M.Eng. degree in computer science from Cornell University, and the Ph.D. degree in media and governance from Keio University. He was with Sony Computer Science Laboratories, Inc. during 1996–2004, and is with IIJ since 2004. He is also an adjunct professor at Japan Advanced Institute of Science

DOI: 10.1002/nem

and Technology, and a board member of the WIDE project. His current research interests include traffic measurement and management, and operating system support for networking.

Hiroshi Esaki received Ph.D from University of Tokyo, Japan, in 1998. In 1987, he joined Research and Development Center, Toshiba Corporation. From 1990 to 1991, he has been at Applied Research Laboratory of Bellcore Inc., New Jersey, as a residential researcher. From 1994 to 1996, he has been at Center for Telecommunication Research of Columbia University in New York. From 1998, he has served as a professor at the University of Tokyo, and as a board member of WIDE Project. Currently, he is executive director of IPv6 promotion council, vice president of JPNIC, IPv6 Forum Fellow, director of WIDE Project and Emeritus Board of Trustee for Internet Society.

DOI: 10.1002/nem