PAPER *Special Section on the Architectures, Protocols, and Applications for the Future Internet*

Detecting Anomalies in Massive Traffic Streams Based on S-Transform Analysis of Summarized Traffic Entropies

Sirikarn PUKKAWANNA[†a)], ***Nonmember***, **Hiroaki HAZEYAMA**[†], **Youki KADOBAYASHI**[†], ***and*** **Suguru YAMAGUCHI**[†], ***Members***

SUMMARY Detecting traffic anomalies is an indispensable component of overall security architecture. As Internet and traffic data with more sophisticated attacks grow exponentially, preserving security with signature-based traffic analyzers or analyzers that do not support massive traffic are not sufficient. In this paper, we propose a novel method based on combined sketch technique and S-transform analysis for detecting anomalies in massive traffic streams. The method does not require any prior knowledge such as attack patterns and models representing normal traffic behavior. To detect anomalies, we summarize the entropy of traffic data over time and maintain the summarized data in sketches. The entropy fluctuation of the traffic data aggregated to the same bucket is observed by S-transform to detect spectral changes referred to as anomalies in this work. We evaluated the performance of the method with real-world backbone traffic collected at the United States and Japan transit link in terms of both accuracy and false positive rates. We also explored the method parameters' influence on detection performance. Furthermore, we compared the performance of our method to S-transform-based and Wavelet-based methods. The results demonstrated that our method was capable of detecting anomalies and overcame both methods. We also found that our method was not sensitive to its parameter settings.

key words: *anomaly detection, sketch, entropy, time-frequency analysis, S-transform*

1. Introduction

Detecting malicious traffic is a challenging task for network administrators. To detect malicious traffic, administrators typically use Intrusion Detection Systems (IDS) in which there two approaches are applied: misuse and anomaly-based detections. The misuse-based IDSs (e.g., Snort [19]) use pre-defined attack signatures to catch attacks. These IDSs precisely detect known attacks, but they are not able to detect unknown and novel attacks. While the anomaly-based IDSs look for abnormal traffic which may be malicious or benign. An advantage of the anomaly-based IDSs is that they do not need attack signatures. As a result, they are relevant to be applied to the current fast-moving Internet world where novel attacks are constantly invented.

The anomaly-based IDSs are categorized into two types: supervised and unsupervised detections. The supervised detection creates normal and anomaly models from pre-labeled training traffic datasets, and then classifies new events based on the models. The unsupervised detection does not demand pre-labeled datasets. On the other hand, it finds the major behavior of traffic being analyzed and defines the major behavior as normal behavior of such circumstance. New events that do not conform to the normal behavior are anomalous.

Currently, unsupervised anomaly detection is attracting considerable interest because it does not require any prior knowledge. For a number of years, clustering techniques were applied for unsupervised detection. In general, clustering-based techniques cluster similar instances. Instances that do not belong to the clusters are classified as anomalous [5], [15], [18]. Unfortunately, most of them still require a training phase with unlabeled traffic data to form clusters. Principal Component Analysis (PCA) is also applied for unsupervised detection. It is used to decomposes a traffic feature distribution into normal and anomalous components. Anomalies are revealed when the anomalous components exceed a specified threshold [6], [11]. A well-known drawback of the use of PCA is that it has parameter sensitivity.

Unsupervised anomaly detection via statistical analysis of aggregate traffic data is becoming a more interesting technique, especially to be deployed for massive traffic analysis or on-line detection. The authors in [17] measured two aggregate traffic features, namely packet rate and packet size. They then used Sequential Probability Ratio Test (SPRT) to detect abnormalities in the features. Efficient and widely applicable aggregate traffic features for traffic anomaly detection are entropies, in particular entropies of source and destination IPs, and source and destination ports. A well-known advantage of the entropies is that they are able to capture more fine-grained traffic patterns than volume-based traffic features [7], [16], [28].

On the other hand, the authors in [14], [20] believed that time-frequency domain analysis provides new insights into data that cannot be obtained from time-domain analysis. They applied Wavelet transform to traffic time-series and detected changes in the coefficients. Our previous works [25], [26] introduced S-transform as a new efficient time-frequency analysis tool for detecting anomalies, especially hidden Denial of Service (DOS) and network probe attacks. In the works, all frequency components of traffic time-series were exposed by S-transform and were depicted in matrix heat maps. Finally, we detected abnormalities in the heat maps based on heuristics [25] or an image process-

Manuscript received May 23, 2014.
Manuscript revised October 7, 2014.
Manuscript publicized December 11, 2014.

[†]The authors are with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630–0192 Japan.

a) E-mail: sirikarn-p@is.naist.jp

DOI: 10.1587/transinf.2014NTP0006

ing technique [26].

Sketch technique has been applied in anomaly detection because it provides scalability and helps to improve detection performance [10]. The sketch technique is typically used to process the traffic data before performing anomaly detection. The authors in [8] utilized the sketch technique to randomly aggregate traffic flows and then used CUmulative SUM (CUSUM) to identify change points. The sketch technique was also applied together with PCA [27] or Wavelet transform [9], [12], [24] to detect unusual events.

In this paper, we propose an unsupervised anomaly detection method based on sketch and S-transform for detecting anomalies in massive traffic streams. More specifically, at every constant time-bin, traffic summarizers, which are technically hash functions, describe the entropy of traffic in their own way. Then, we apply S-transform to the entropy time-series to detect anomalous events in the time-frequency domain of the time-series. To achieve a low positive rate, we determine an event is anomalous when it was reported as anomalous by every summarizer.

The proposed method has four key features. First, it operates on aggregate traffic data without deep-packet inspection which enables us to analyze encrypted and massive traffic. Second, it does not require pre-defined signatures of prospective targets and pre-labeled traffic datasets. Third, it employs the sketch technique for traffic summarization. These three features make the method scalable. Fourth, as it looks for abnormalities in time-frequency domains, it can detect extra anomalies apart from time-domain analysis-based methods.

In this paper, we also evaluated the performance of the method with Internet backbone traffic in terms of accuracy and false positive rates. Furthermore, with the same traffic dataset, we explored the method parameters and present their effects on detection performance. Lastly, we compared the detection performance between our method with two unsupervised anomaly detection methods: (1) S-transform-based and (2) sketch and Wavelet transform-based methods.

The rest of this paper is organized as follows. Section 2 describes related work. Section 3 describes details of the sketch technique, entropy, and S-transform. The proposed anomaly detection method is described in Sect. 4. Section 5 describes the traffic dataset, performance evaluation, and results. Section 6 describes several issued involved with this work. Section 7 describes our conclusions and future work.

2. Related Work

Some related work have been discussed in Sect. 1. Here we briefly describe remaining related work, especially time-frequency-based anomaly detection methods. The authors in [20] used Discrete Wavelet Transform (DWT), which is the most popular time-frequency representation tool, to decompose a traffic signal (e.g., SNMP data) into many scales. Then, they filtered only a set of scales and detected anomalies in the Wavelet coefficients. The works in [9], [12], [24] enhanced their DWT-based anomaly detectors by adding traffic sketching before performing anomaly detection. The authors in [4] applied Continuous Wavelet Transform (CWT) instead of DWT to detect volume-based network anomalies.

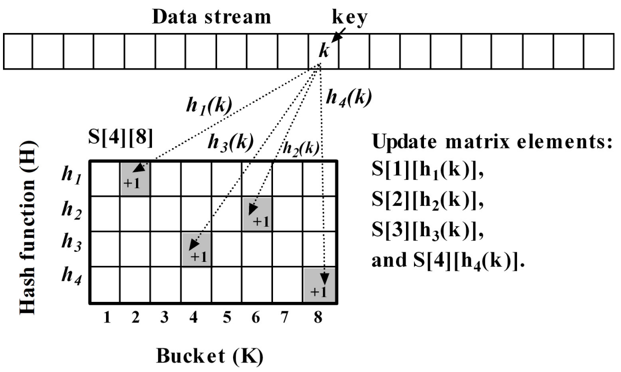

Fig. 1 A sketch and updating the sketch (H=4 and K=8).

A well-known advantage of the Wavelet transform-based methods is that they are able to detect the various behavior of anomalies due to multi-resolution analysis. However, determining which mother wavelets (e.g., Daubechies) is an important parameter that impacts detection performance [14]. Furthermore, determining how many decomposition levels still remains a point of contention.

3. Overview

3.1 Sketch

A sketch is a data structure used to summarize a data stream. Technically, a sketch is a two-dimensional $H \times K$ array $S[H][K]$, where each row $(1, 2, .., H)$ is associated with different hash functions, and the columns $(1, 2, .., K)$ are the hash buckets (see Fig. 1). The matrix element $S[i][j]$ contains the counter associated with the hash bucket j of the hash function i. To summarize a data stream, an empty sketch $S[H][K]$ is created in which all elements are set to zero. Then, hash functions $h_1, h_2, .., h_H$ linearly hash each key k (e.g., source IP) in the stream. The counters of corresponding matrix elements are updated. Figure 1 depicts summarizing a data stream and storing the summarized data in a sketch constructed by four hash functions. Each hash table has eight buckets.

Summarizing a data stream using sketch technique has two main advantages: it touches original data only one time and uses a fixed amount of memory to store the summarized data. This leads to its application for analyzing and detecting changes in massive data streams.

Technically, to detect changes in a data stream using the sketch technique, a detector constructs a sketch and then continuously updates it when an input key arrives. Until a counter in the sketch reaches a specified value, the detector raises an alarm and assumes that a change occurs.

In this work, we applied the sketch technique. Instead of detecting heavy buckets in one sketch, we constructed several sketches with the same size at different times. We then observed the transformation of the sketches. Finally, we detected culprits who made big transformations.

3.2 Shannon Entropy

Shannon entropy [13] is a measure of the randomness of a set of data. Technically, the entropy of a set of random variable X with possible values $x_1, x_2, ..x_n$ is conventionally defined as:

$$E(X) = -\sum_{i=0}^{n} p_i log_2 p_i \tag{1}$$

where p_i is the probability of value x_i that occurs in the data. The p_i is calculated by the frequency of the value x_i divided by the frequency of all possible n values. The *E(X)* is in the range of zero to $log_2(n)$. The *E(X)* is zero when there is absolutely no randomness. For example, when there is only one value in the data. The *E(X)* is close to zero when the data contains a few values. Conversely, the *E(X)* is $log_2(n)$ or close to $log_2(n)$ when every value equally participates in the data.

To apply the entropy concept for unsupervised statistic-based anomaly detection, detectors typically observe the degree of randomness of a traffic feature (e.g., destination port). Then, they detect large variations of the randomness. For example, in an enterprise network where the IP entropy is normally high, if an attacker abruptly generates a huge number of packets, the detectors raise an alert because the entropy decreased.

In this work, we similarly considered the entropy as a traffic feature to represent traffic behavior. However, we did not analyze the time-domain characteristic of the entropy time-series like the example above. We instead analyzed its time-frequency domain characteristic. In other words, we considered its spectral content. To completely discover various kinds of anomalies, we concentrated on four entropies: source IP, destination IP, source port, and destination port entropies. The formulas to compute p_i of the source IP, destination IP, source port, and destination port entropies are shown below.

$$p_i = \frac{\#pktsOfSrcIP_i|DstIP_i|SrcPort_i|DstPort_i}{\#totalPktsSeen} \tag{2}$$

3.3 S-Transform

S-transform [22] is a time-frequency representation tool used to discover frequency components in a time-series. To discover frequency components, the S-transform segments the time-series using a scalable Gaussian window and then extracts frequencies of the segmented parts using Fourier transform. The S-transform of a time-series $x(t)$ is conventionally defined as

$$ST(\tau, f) = \int_{-\infty}^{\infty} x(t) \frac{|f|}{\sigma\sqrt{2\pi}} e^{\frac{-(\tau-t)^2 f^2}{\sigma^2}} dt \tag{3}$$

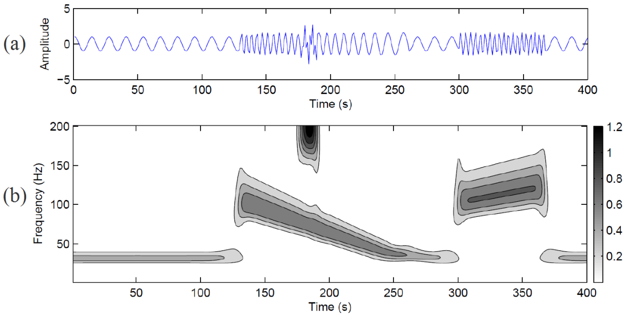

Fig. 2 (a) A time-series (b) Time-frequency domain of (a) by S-transform.

where t and τ are both time, but τ is used to control the time resolution. The f is the Fourier frequency and the σ is the scale parameter controlling the frequency resolution.

Figure 2 (a) shows a multiple frequency time-series containing 400 samples and Fig. 2 (b) shows the S-transform's output presenting the time-frequency domain of the time-series. The x-axis represents time corresponding to the time of the time-series. The y-axis represents frequency and the color represents amplitude. The figures show that the S-transform precisely shows us how each frequency behaves and when they change behavior.

The S-transform has advantages that support anomaly detection as follows. It performs multi-resolution analysis, thus it discovers various types of anomalies. It uses the Fourier kernel to provide the absolute phase information of each frequency component. This phase information is referenced to the time origin. As a result, the S-transform provides supplementary information about spectra which is not available from locally referenced phase information obtained by Wavelet transform [22]. Furthermore, it produces time-frequency plots that are easier to visually analyze for time-frequency behavior than Wavelet transform's outputs because the Wavelet transform produces time-scale plots which are intricate and cannot be analyzed directly.

In this work, we utilized the S-transform to discover time-frequency behavior of entropy time-series of aggregate traffic. We considered frequencies from 0 Hz to $\frac{length(x)}{2}$ Hz. For the σ, we set it to be one according to the conventional S-transform [22].

4. Anomaly Detection Method

In this section, we describe the proposed method for detecting anomalies in massive traffic streams. The method is based on sketch technique and S-transform. It consists of three main steps: (1) Summarizing the traffic stream, (2) Detecting suspicious time-bins, and (3) Finding the intrinsic culprits of anomalies. Below we describe the three steps in detail.

4.1 Summarizing the Traffic Stream

In this step, we summarize the traffic stream and keep the

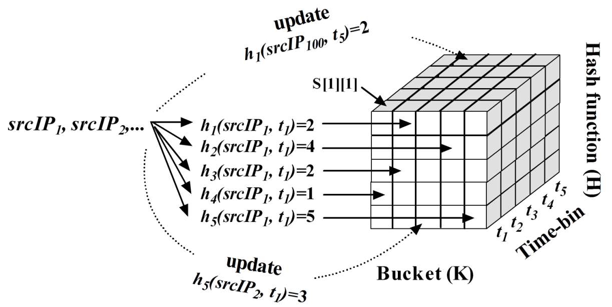

Fig. 3 Summarizing a traffic stream at different five time-bins (H=5, K=5, and key=srcIP).

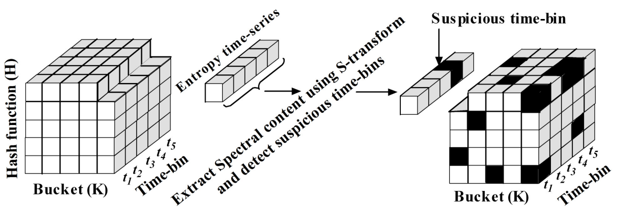
Hash function (H)
Bucket (K)
t_1 t_2 t_3 t_4 t_5
Time-bin
Entropy time-series
Extract Spectral content using S-transform
and detect suspicious time-bins
Suspicious time-bin
Bucket (K)
t_1 t_2 t_3 t_4 t_5
Time-bin

Fig. 4 Detecting suspicious time-bins in entropy time-series.

summarized data in sketches. In every constant time-bin we perform the following operations. Firstly, we create an empty sketch $S[H][K]$, where H is the number of hash functions used to summarize the traffic and K is the number of buckets per hash table (sketch size). All elements in $S[H][K]$ are initially set to zero. Secondly, we use the H hash functions to group the keys (e.g., source IPs) in the time-bin to K buckets. h_1 hashes the keys to its buckets, h_2 hashes the same keys to its buckets, and so on. The keys that have the same property (same hash value) are grouped into the same buckets. Thirdly, we compute the entropy of the keys in each bucket using Eq. (1) and (2). Then, we store the entropy value in the associated sketch element. For example, the entropy value of the keys that were hashed by h_1 and dropped to its first bucket is stored in $S[1][1]$.

Figure 3 illustrates summarizing a traffic stream at time-bins $t_1, t_2, .., $ and t_5 using five hash functions. The sketch size is five and the keys are source IPs in the stream. At t_1, the entropy value in $S[1][2]$ is updated because $h_1(srcIP_1)$ is two, $S[2][4]$ is updated because $h_2(srcIP_1)$ is four, $S[3][2]$ is updated because $h_3(srcIP_1)$ is two, and so one. At t_2 to t_5, new four sketches with the same size are created and updated based on new keys in the particular time-bins. After constructing and updating the sketches, we can see how the entropy of keys grouped to the same bucket fluctuates at every time-bin as an entropy time-series. According to Fig. 3, we obtain 25 entropy time-series which have a length of five.

4.2 Detecting Suspicious Time-bins

The goal of this step is to investigate the entropy time-series obtained from the previous step and detect suspicious time-bins containing anomalies or changes. Firstly, we remove the DC component of the entropy time-series $x(t)$ by the following equation.

$$x'(t) = x(t) - MEAN \tag{4}$$

where $MEAN$ is the mean value of the whole entropy time-series $x(t)$. All points in the $x(t)$ are subtracted by its mean. The purpose of this process is to remove the constant values that are added to the time-series. These values distort the frequency components in the time-series. Secondly, the S-transform converts the $x'(t)$ to a time-frequency domain and produces a matrix heat map like Fig. 2 (b). Thirdly, we produce two additional time-series from the heap map: 1) a time-series that is obtained by vertically summing all matrix elements in the upper half and 2) a time-series that is obtained by vertically summing all matrix elements in the lower half of the matrix. The first time-series shows the amplitude variation of the high frequency components and the latter shows the amplitude variation of the low frequency components over time. In this work, we consider time-bins that hold deviant amplitudes as suspicious time-bins. Thus, in the first time-series we find time-bins that hold amplitude values above a given SD-based upper threshold (*upper_thres*). For the second time-series, we find time-bins that hold amplitude values below a SD-based given lower threshold (*lower_thres*). Figure 4 illustrates the processes of detecting suspicious time-bins in the entropy time-series. The detected suspicious time-bins are highlighted in black.

4.3 Finding the Intrinsic Culprits of Anomalies

In the previous steps, the H summarizers (hash functions) aggregated the traffic according to their own scheme and produced HxK entropy time-series. Then, the S-transform-based detector analyzed the time-series individually to identify suspicious time-bins which tend to contain anomalies.

Instinctively, all keys exist in the suspicious time-bins of one summarizer are possible to be culprits who cause the changes. As most unsupervised anomaly detection techniques produce a number of false positives, to reduce the false positives we combine the suspicious keys in the detected suspicious time-bins obtained from all summarizers by taking the intersection. The keys in the intersection result are considered as intrinsic culprits of anomalies in the traffic.

5. Performance Evaluation and Results

In this work, we evaluated the performance of the proposed method in terms of accuracy and false positive rates. We also investigated the effect of the method parameters on detection performance. Furthermore, we performed two performance comparisons: our method vs. non-sketch-based method and our method vs. Wavelet transform-based method.

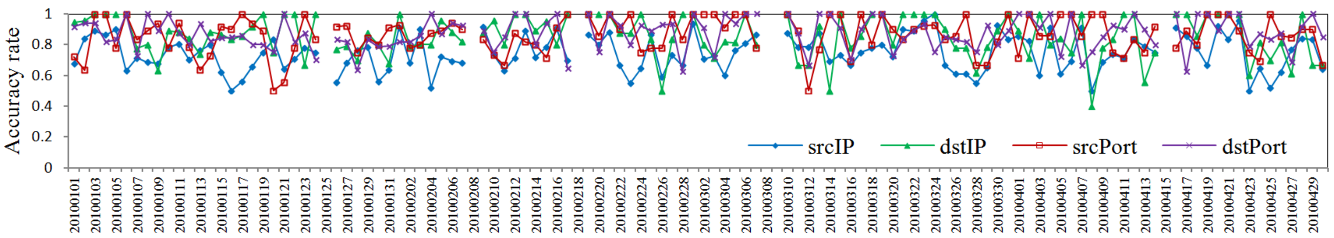

Fig. 5 Accuracy in detecting anomalies in MAWI traces collected from Jan. 1 to Apr. 30, 2010.

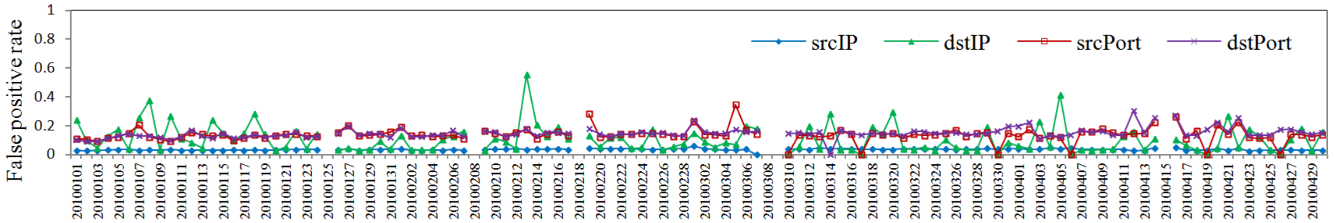

Fig. 6 False positive in detecting anomalies in MAWI traces collected from Jan. 1 to Apr. 30, 2010.

5.1 Traffic Dataset and Anomaly Labels

So far, the performance of traffic anomaly detection methods were evaluated using real-world traffic, synthetic traffic, or private traffic collected in organization networks. Intuitively, evaluation results obtained from testing with real-world traffic are much more reliable.

In this work, we evaluated our method with real-world traffic traces from the MAWI dataset [2]. More specifically, the MAWI trace contains Internet backbone traffic collected daily for 15 minutes at samplepoint-F of a 150 Mbps transit link between the United States and Japan. We tested the method with 114 traces collected from January 1st to April 30th, 2010. Note that six traces during this period are unavailable in MAWI dataset.

To verify results, we used anomaly labels from MAWILab [3] as benchmark. The MAWILab labels are trustworthy because they were derived by combining detection results from four anomaly detectors, namely Hough transform, Gamma distribution, Kullback-Leibler divergence, and PCA-based detectors [21]. The MAWILab provides four types of labels, namely anomalous, suspicious, notice, and benign. In this work, we considered only the anomalous labels for verification. The reason we selected the traces from 2010 instead of newer traces is that the MAWILab provides complete labels only until April, 2010.

5.2 Results

In the experiment, the parameters were set as follows. The H was three. The three hash functions are general purpose hash functions from [1], namely RSHash, PJWHash, and ELFHash. The K was 64 and the time-bin size was one second. The keys were source IPs, destination IPs, source ports and destination ports. For the $upper\_thres$ and $lower\_thres$, we set them based on a three-sigma rule. The $upper\_thres$ was $2SD$ and the $lower\_thres$ was $-2SD$. This means that we detected time-bins containing 5% of the amplitude values that are more than $2SD$ or less than $-2SD$.

In this experiment, accuracy and false positive rates of detection were measured. The accuracy rate was computed as the total number of anomalies that were correctly detected by our method divided by the total number of anomalous labels from the MAWILab. The false positive rate was computed as the total number of normal instances that were incorrectly detected as anomalies by our method divided by the total number of normal instances in the trace.

Figure 5 and 6 show the accuracy and false positive rates in detecting anomalous source IPs, destination IPs, source ports, and destination ports in different 114 MAWI traces. The figures indicate that the overall accuracy rate is above 60% and in some traces our method succeeded in detecting anomalies with 100% accuracy. The average accuracy rates of detected anomalous source IPs, destination IPs, source ports, and destination ports are 75%, 86%, 86%, and 88% respectively. The false positive rate of anomalous source IP detection is rather stable at about 3%. The false positive rates in detecting anomalous source and destination ports are about 12%. While the false positive rate of anomalous destination IP detection is rather inconclusive fluctuating. In summary, our method could moderately detect anomalous source IPs with low false positive rates. For anomalous destination IPs, source ports, and destination ports, our method could detect anomalies with more precision, but with increased false positive rates.

We also measured the detection time for one analysis. We randomly tested our detector with five MAWI traces. Our detector ran on a 10.04 Ubuntu virtual machine with 3.4GHz CPU and 8GB of RAM. We found that it took approximately two minutes to automatically performed the three steps (until printing out intrinsic culprits) described in Sect. 4. For example, it took 1.5 minutes to detect anomalies in the trace collected on January 1st, 2011. The trace size is

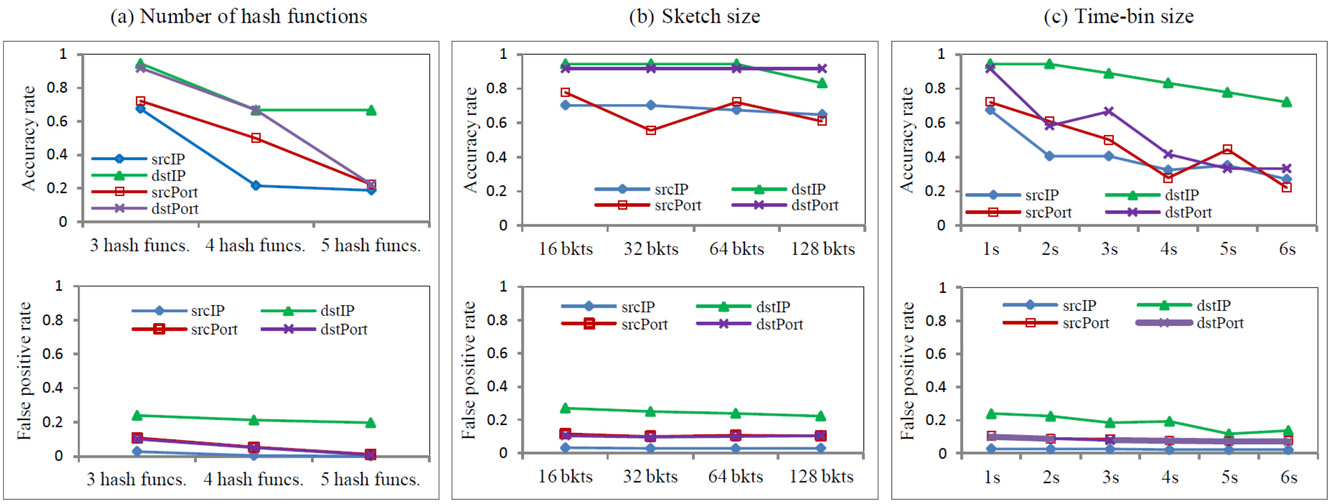

Fig. 7 Detection performance as a function of (a) the number of hash functions, (b) sketch size, and (c) time-bin size.

1.6 GB containing 22 million packets and the traffic rate is 152 Mbps. Note that our detector's detection time depends on many factors such as traffic characteristics, the number of hash functions, and sketch size.

5.3 Exploring Parameters

This section describes how each of the parameters affects the detection performance of our method. We explored three method parameters: the number of hash functions, sketch size, and time-bin size. Note that we randomly tested our detector with several MAWI traces and the results were consistent. All figures referred to in this section illustrate the results derived from testing on the trace collected on January 1st, 2010.

Number of hash functions Fig. 7 (a) depicts the accuracy (above) and false positive (below) rates as a function of the number of hash functions. We found that using smaller numbers of hash functions provided higher accuracy rates, while the false positive rates slightly decreased. The accuracy rates decreased when we used more hash functions because our method detected the keys in the intersection result.

Sketch size We tested the method with various values of K, such as 16, 32, 64, and 128, and measured accuracy and false positive rates. Figure 7 (b) depicts the effect of the sketch size. The results show that the sketch size apparently affected the performance. Moreover, we found that 16 is the best sketch size for detecting anomalies in the MAWI traces because it used the lowest amount of memory and gave the performance similar to the remaining sketch sizes

Time-bin size Fig. 7 (c) plots the performance as a function of time-bin size. We observed the consequences of adjusting the time-bin size from one second to six seconds. The results show that the smaller time-bin sizes increased accuracy as well as false positive rates. Thus, to get the best performance from our method, a small time-bin size should be taken into account.

In summary, our method was slightly sensitive to the number of hash functions and time-bin size in terms of accuracy. Smaller numbers of hash functions and time-bin size provided better detection performance.

5.4 Sketch vs. Non-sketch

The proposed method utilized sketch technique for improving detection performance and scalability. In this work, we also compared the performance between our method with a single S-transform-based method to confirm that the sketch technique enhances time-frequency-based anomaly detection methods. In this experiment, our method's parameters were set to the same values described in Sect. 5.2. For the single S-transform-based detector, it read traffic traces and produced entropy time-series without using sketches. The size of time-bin is one second. Then, S-transform transforms the time-series to heat maps. Finally, it used *upper_thres* and *lower_thres* to detect suspicious time-bin. All keys in the detected suspicious time-bins were determined as culprits. Both detectors considered the same frequencies (0 Hz to $\frac{length(x)}{2}$ Hz) and used the same threshold values for analysis. Ten MAWI traces collected between 2007 and 2011 were investigated in this experiment. Figure 8 shows the accuracy comparison. Figure 9 shows the false positive comparison of the two detectors to detect anomalous source IPs. The results indicate that our detector mostly outperformed the single S-transform-based detector in terms of both accuracy and false positive rates.

5.5 S-transform vs. Wavelet Transform

We also compared our results with results from a Wavelet transform-based anomaly detection method proposed in [12]. More specifically, the authors in [12] applied a DWT-based method already described in [20] to detect anomalies

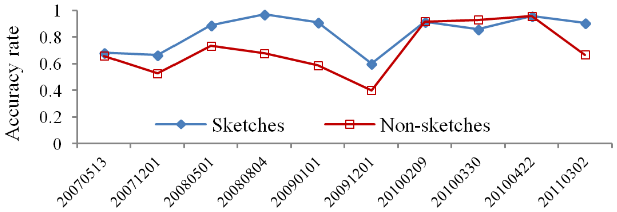

Fig. 8 Accuracy in detecting anomalous source IPs of sketches and non-sketch-based methods.

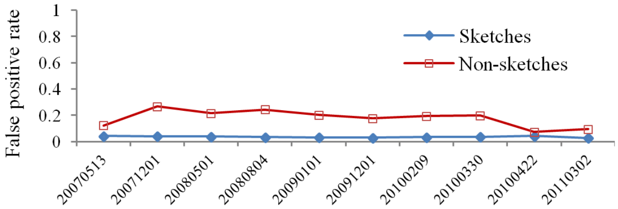

Fig. 9 False positive in detecting anomalous source IPs of sketches and non-sketch-based methods.

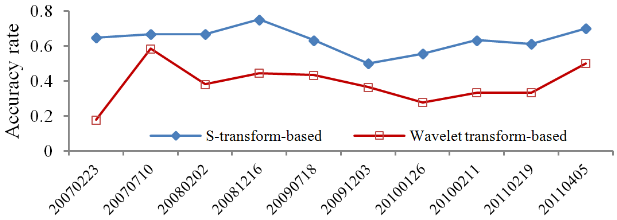

Fig. 10 Accuracy in detecting anomalous source IPs of S-transform and Wavelet transform-based methods.

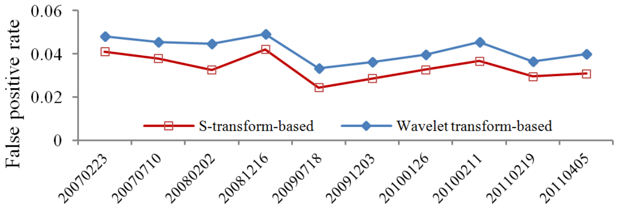

Fig. 11 False positive in detecting anomalous source IPs of S-transform and Wavelet transform-based methods.

in time-series given by the temporal evolution of a sketch bucket. For the Wavelet transform-based detector, we set the number of hash functions, the sketch size, and the time-bin size were equivalent to the values in our detector. The mother wavelet was Daubechies D4 and the maximum decomposition level was three. The detection threshold was four. In this experiment, both detectors investigated the same ten MAWI traces. The accuracy and false positive rates of both detectors are shown in Fig. 10 and 11 respectively. Figure 10 shows that our detector detected about 64% of anomalies, while the Wavelet transform-based detector detected about 37% of anomalies. About the false positive rate, the Wavelet transform-based detector generated more false positive alarms than our detector.

6. Discussion

Our performance analysis was limited by the availability of traffic datasets that contain real and modern traffic with reliable labels. Due to the limitation, we evaluated the performance of the method with traffic traces from only MAWI dataset. Even though the investigated traffic is real-world traffic that can be representative of nowadays Internet traffic, however the obtained evaluation results shown in Sect. 5 still are specific to these traffic and the parameter values which were set during the experiments.

To apply the method in other networks, to get expected detection performance, parameter tuning may be required. For example, if network administrators wish to see as much anomalous traffic as possible, they should set the thresholds to be $\pm SD$.

In this paper, we applied sketch technique and S-transform toward detecting anomalies. Compared to existing anomaly detection methods based on sketch technique and Wavelet transform, our proposed method has two main differences. First, it considers entropy fluctuation. By contrast, the existing methods consider volume-based traffic features (e.g., packet count). Second, it employs S-transform, which overcomes shortcomings of Wavelet transform: 1) it retains absolute phase information and 2) it produces outputs that are easier for visual analysis [22].

Sketches theoretically only contain counters and do not preserve original keys in a data stream. Thus, in this work, to identify the culprits, the original keys were kept temporarily. For real-world applications that must avoid high memory consumption to maintain the original keys, reversible sketches [23] can be efficiently applied. As the focus of our work is detection performance, we did not utilize or implement the reversible sketches.

7. Conclusion and Future Work

In this paper, we proposed an unsupervised anomaly detection method based on sketch and S-transform. We evaluated the detection performance of the method over four months of MAWI backbone traffic. We found that the method could detect anomalies with 60% to 100% accuracy. The false positive rates are between 3% to 12%. We also analyzed the effect of the method parameters and the results indicate that our method was not highly sensitive to parameter tunings. Furthermore, we compared the performance of our method with two unsuperivsed anomaly detection methods: (1) single S-transform-based and (2) sketch and Wavelet transform-based methods. The results show that our method outperformed both methods in terms of accuracy and false positives.

Future work will be devoted to overcome the limitation of the method, which is represented by the need for manual tuning some parameters. In particular, the main efforts will be focused on the search for an alternate automatic way of setting the thresholds. Furthermore, the application of the reversible sketches will be taken into consideration for improving the performance in terms of memory and time consumptions.

References

[1] General Purpose Hash Function Algorithms. http://www.partow.net/programming/hashfunctions

[2] MAWI Traffic Archive. http://mawi.wide.ad.jp

[3] MAWILab. www.fukuda-lab.org/mawilab

[4] A. Dainotti, A. Precape, and G. Ventre, "NISO4-1: Wavelet-based detection of Dos attacks," Proc. GLOBECOM, pp.1–6, 2006.

[5] A. Kind, M. Stoecklin, and X. Dimitropoulos, "Histogram-based traffic anomaly detection," IEEE Trans. Network Service Management, vol.6, no.2, pp.110–121, 2009.

[6] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," Proc. SIGCOMM, pp.217–228, 2005.

[7] A. Wagner and B. Plattner, "Entropy based worm and anomaly detection in fast IP networks," Proc. WETICE, pp.172–177, 2005.

[8] C. Callegari, A. Casella, S. Giordano, M. Pagano, and T. Pepe, "Sketch-based multidimensional IDS: A new approach for network anomaly detection," Proc. CNS, pp.350–358, 2013.

[9] C. Callegari and S. Giordano, "On the use of sketches and wavelet analysis for network anomaly detection," Proc. IWCMC, pp.331–335, 2010.

[10] C. Callegari, L. Gazzarrini, S. Giordano, M. Pagano, and T. Pepe, "When randomness improves the anomaly detection performance," Proc. ISABEL, pp.1–5, 2010.

[11] C. Callegari, L. Gazzarrini, S. Giordano, M. Pagano, and T. Pepe, "A novel PCA-based network anomaly detection," Proc. ICC, pp.1–5, 2011.

[12] C. Callegari, S. Giordano, M. Pagano, and T. Pepe, "Combining sketches and wavelet analysis for multi time-scale network anomaly detection," Computers and Security Journal, vol.30, no.8, pp.692–704, 2011.

[13] C.E. Shannon, "A mathematical theory of communication," Bell Syst. Tech. J., vol.27, pp.379–423, 623–656, 1948.

[14] C. Huang, S. Thareja, and Y. Shin, "Wavelet-based real time detection of network traffic anomalies," Int. J. Network Security, vol.6, no.3, pp.309–320, 2008.

[15] G. Munz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," Proc. GI/ITG-Workshop MMBnet, 2007.

[16] G. Nychis, V. Sekar, D. Andersen, H. Kim, and H. Zhang, "An empirical evaluation of entropy-based traffic anomaly detection," Proc. IMC, pp.151–156, 2008.

[17] G. Thatte, U. Mitra, and J. Heidemann, "Parametric methods for anomaly detection in aggregate traffic," IEEE/ACM Trans. Netw., vol.19, no.2, pp.512–515, 2010.

[18] L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion detection with unlabeled data using clustering," Proc. CSS Workshop on DMSA, 2001.

[19] M. Roesch, "Snort-lightweight intrusion detection for networks," Proc. USENIX LISA, pp.229–238, 1999.

[20] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," Proc. IMW, pp.71–82, 2002.

[21] R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda, "MAWILab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking," Proc. CONEXT, pp.1–12, 2010.

[22] R.G. Stockwell, L. Mansinha, and R.P. Lowe, "Localization of the complex spectrum: the s-transform," IEEE Trans. Signal Process., vol.44, no.4, pp.998–1001, 1996.

[23] R. Schweller, A. Gupta, E. Parsons, and Y. Chen, "Reversible sketches for efficient and accurate change detection over network data streams," Proc. IMC, pp.207–212, 2004.

[24] S. Pukkawanna and K. Fukuda, "Combining sketch and wavelet models for anomaly detection," Proc. ICCP, pp.313–319, 2010.

[25] S. Pukkawanna, H. Hazeyama, Y. Kadobayashi, and S. Yamaguchi, "Building better unsupervised anomaly detector with s-transform," Proc. NSS, LNCS, vol.7873, pp.582–589, 2013.

[26] S. Pukkawanna, H. Hazeyama, Y. Kadobayashi, and S. Yamaguchi, "Investigating the utility of s-transform for detecting denial-of-service and probe attacks," Proc. ICOIN, pp.282–287, 2014.

[27] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccrone, and A. Lakhina, "Detection and identification of network anomalies using sketch subspaces," Proc. IMC, pp.147–152, 2006.

[28] Y. Kanda, R. Fontugne, K. Fukuda, and T. Sugawara, "Admire: Anomaly detection method using entropy-based PCA with three-step sketches," Comput. Commun., vol.36, no.5, pp.575–588, 2013.

Sirikarn Pukkawanna received her M.S. degree in Computer Science from Mahidol University in 2008. Currently, she is a Ph.D. student in the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. Her research interests include Internet traffic measurement and analysis and anomaly detection.

Hiroaki Hazeyama received his Ph.D. degree in Engineering from Nara Institute of Science and Technology (NAIST), Japan, in 2006. He is currently an associate professor at NAIST. His research interests include network operation, network security, and large-scale network test-bed.

Youki Kadobayashi received his Ph.D. degree in Computer Science from Osaka University, Japan, in 1997. He is currently an associate professor in the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. Since 2009, he has also been working as an associate rapporteur of ITU-T Q.417 for cybersecurity standardization. His research interests include cybersecurity, web security, and distributed systems.

Suguru Yamaguchi received his Ph.D. degree in Computer Science from Osaka University in 1997. Currently, he is a Professor in the Graduate School of Information Science, Nara Institute of Science and Technology. His research interests include information sharing, multimedia communication over high speed communication channels, network security, and network management for the Internet.