ADMIRE: Anomaly Detection Method Using Entropy-based PCA with Three-step Sketches

Yoshiki Kanda[a], Romain Fontugne[b],
Kensuke Fukuda[b,c], Toshiharu Sugawara[a]

[a] *Graduate School of Fundamental Science and Engineering, Waseda University, Tokyo, Japan*
[b] *The Graduate University for Advanced Studies, Tokyo, Japan*
[c] *National Institute of Informatics/PRESTO JST, Tokyo, Japan*

Abstract

Network anomaly detection using dimensionality reduction has recently been well studied in order to overcome the weakness of signature-based detection. Previous works have proposed a method for detecting particular anomalous IP-flows by using random projection (sketch) and a Principal Component Analysis (PCA). It yields promising high detection capability results without needing a pre-defined anomaly database. However, the detection method cannot be applied to the traffic flows at a single measurement point, and the appropriate parameter settings (e.g., the relationship between the sketch size and the number of IP addresses) have not yet been sufficiently studied. We propose in this paper a PCA-based anomaly detection algorithm called ADMIRE to supplement and expand the previous works. The key idea of ADMIRE is the use of three-step sketches and an adaptive parameter setting to improve the detection performance and ease its use in practice. We evaluate the effectiveness of ADMIRE using the longitudinal traffic traces captured from a transpacific link. The main findings of this paper are as follows: (1) We reveal the correlation between the number of IP addresses in the measured traffic and the appropriate sketch size. We take advantage of this relation to set the sketch size parameter. (2) ADMIRE outperforms traditional PCA-based detector and other detectors based on different theoretical backgrounds. (3) The types of anomalies reported by ADMIRE depend on the traffic features that are selected as input. Moreover, we found

Email address: y.kanda@isl.cs.waseda.ac.jp (Yoshiki Kanda)

that a simple aggregation of several traffic features degrades the detection performance.

Keywords: PCA, hash, sketch, anomaly detection, entropy

---

1. Introduction

The number of abnormalities in communication network traffic based on both malevolent and benign intentions has been increasing. The former includes network scanning, worm propagation, DDoS, and so forth, which can have detrimental effects on Internet services. The latter includes flash crowds, sudden changes in demand, equipment failures, etc. In order to constantly and safely operate communication networks and to make good use of a limited number of network resources, we need automatic detection methods that can find abnormal events.

Historically, there are two approaches for the automatic detection of anomalous events: misuse detection and anomaly detection. Misuse detection such as snort [2] matches a packet's payload's patterns to those in the predefined database. Even though it can accurately detect anomalous activities, it is unable to detect new types of worms or unknown misuse activities whose payload's patterns are not included in the database. On the other hand, anomaly detection methods using the statistical behavior of the traffic have recently been attracting a lot of researchers 'attention since they do not require a predefined database and have the potential to detect new worms under an assumption that those attacks deviate from the normal statistical behavior.

Our focus in this paper is the anomaly detection methods using the statistical behavior of the traffic. We explain several examples of statistical method applied for anomaly detection. An entropy-based approach for anomaly detection[5] computes the entropy of the distribution of packet feature (IP addresses, ports, etc.) and report anomalies if the entropy value deviates from a standard deviation. Entropy based anomaly detection provides more fine-grained insights than the traditional volume based one. ASTUTE[1] defined a model for normal traffic behavior as short-timescale uncorrelated traffic equilibrium. The equilibrium property holds if the traffic flows (a set of packets that share the same values for a given set of traffic features such as source and destination IP addresses, ports, and protocol number) are nearly independent, and is violated by traffic changes caused by correlated flows.

ASTUTE detects anomalies based on such equilibrium property assuming that a large number of flows traverses a non-saturated link. A wavelet-based approach[14, 15] detect anomalies by utilizing the difference between the time-varying signals of normal traffics and the abnormal network traffics in frequency band on condition that the energy of anomalous traffics is higher than the total energy in certain frequency band. A multi-scale gamma modeling based approach[10, 11] approximates traffic using Gamma distribution and traffic that is distant from adaptively computed reference is detected as anomaly. A Kullback-Leibler (KL) approach[19] constructs several kinds of histograms that monitor distinct traffic features by KL divergence to detect prominent change in traffic. A Principal Component Analysis (PCA) based approach[6, 4, 7, 8, 9, 21, 22] explains the main feature of traffic by dimensionality-reduction and reports the residual traffic as anomaly. PCA is probably the best-known statistical-analysis technique for network anomaly detection. *Defeat*[9] seems to be the most recent and practical approach because it helps to specify the network-wide anomalies at a per-host granularity by incorporating entropy-based PCA using sketch [13] techniques (random projection to reduce the dimensionality of the data).

Even though we admire the large contribution of *Defeat*, three points still remain to be more closely investigated including the appropriate sketch sizes, which is the IP header information (source/destination IP addresses or ports) we use as the entropy's original traffic, and a capability comparison with other types of anomaly detections using a longitudinal observation. First, *Defeat* insists that large sketch sizes decrease the missed detection rates and increase the additional detection rates. However, no theoretical explanation for this is given and the data sets they use are two backbone's week-long traces for a limited observation period that does not show the growth of the throughput and the number of unique IP addresses on the Internet. We suggest that the number of unique IP addresses as well as the throughput in the trace have a positive correlation with the appropriate sketch sizes. Also, *Defeat*'s impact of the entropy's choice has not yet been examined. They only merged the anomalies detected by the entropy of a 4-tuple (source/destination IP addresses and port numbers). We claim that the entropy of different IP header information captures different types of anomalies, and thus, it should be essential to carry out the study of the types of detected anomalies by using a different choice of entropy. Thirdly, *Defeat* only compared the result with other PCA-based anomaly detectors. To understand the PCA's merit and demerit for anomaly detection, it is necessary for us to compare the detected

anomalies of PCA with another type of anomaly detector.

The main contribution of this paper is four-fold. First, we propose ADMIRE, which is a combination of sketches and entropy-based PCA, but is different from *Defeat* in one important respect, it uses three-step sketches to deal with the packet traces measured from a single link. The proposed method using the three-step sketches performs better than the previous two-step sketches in terms of the true and false positive rate. We describe the mechanism and superiority of the three-step sketches in more detail in Section 3.3. Second, we investigate the correlation between the number of unique IP addresses and the appropriate sketch sizes for Internet traffic traces. Consequently, we can observe the positive correlation between them. To the best of our knowledge, this is the first intensive research using a real backbone trace to characterize the correlation between the appropriate sketch sizes for anomaly detection and the number of unique IP addresses. This finding will be helpful for many anomaly detectors using the sketch technique. Third, by evaluating ADMIRE, we revealed that the different entropy time series for PCA anomaly detection captured the different types of anomalies. As consistent with [5], we strongly believe that we should carefully choose the entropy when we use it for anomaly detection. Finally, we compare ADMIRE's detection capability with the gamma [10] and KL [19] methods using nine-year traces. As a result, ADMIRE performs better than the other methods in terms of its detection capability. Since each method detects different types of anomalies, their use in combination would be effective.

2. Related work

Anomaly detection in backbone network traffic has been intensely studied. Out of many different analysis techniques, PCA-based anomaly detection has recently been a hot research topic because of its ability to detect network-wide anomalies by separating the high-dimensional space occupied by a set of network traffic measurements into two distinguishable subspaces corresponding to the normal and anomalous network conditions [7, 8, 9, 18].

Lakhina et al. first applied PCA to the origin-destination (OD) flows for the structural analysis of network flows [4]. An OD flow consists of all the traffic entering the network from a common ingress point and exiting the network from a common egress point. They show that PCA can decompose the structure of the OD flows into three main constituents: common periodic trends, short-lived bursts, and noise. They have also shown that the OD

flows can be accurately modeled in time using a small number (10 or less) of independent components. Ref. [4] had no sooner been published than the authors also applied PCA to the anomaly detection of OD-flows [7]. The information in ref. [7] stated that they could detect and identify the anomalous OD-flows that span multiple network links using the time series of the packet count and size as the input into PCA. Ref. [6] suggests that the entropy time series of traffic features such as IP addresses and ports are better than the packet count and size for the accurate anomaly detection. Li et al. incorporated these works with sketch in order to detect and identify anomalous IP-flows that are more fine-grained than the OD-flow using the entropy time series of traffic features[9]. Ref. [18], on the other hand, stated it improved the time complexity of PCA by using the variance estimation achieving a logarithmic running time and space over the traffic streams in the sliding window model using theoretical guarantees. However, their works did not evaluate an important parameter, the number of normal components called "top_k" , which we explain in the section 3.1 even though Ringberg et al. suggest that PCA-based anomaly detection should be sensitive in top_k[8].

We proposed a packet count-based PCA anomaly detector and approached the top_k's sensitivity problem by using a cumulative proportion-based decision in our previous work [16]. In ref. [16], we insisted that the adaptive decision of top_k based on the cumulative proportion of the principal components outperforms the fixed decision of top_k proposed in [9]. Even though ref. [7, 8, 9, 18] takes advantage of PCA's ability to find anomalies that span multiple links, we conversely enabled PCA to work on single link packet-based traces. Thus, we can evaluate PCA for anomaly detection using much more accessible single link traces than multiple link traces.

In ADMIRE, the input to PCA is not a packet count time series as used in ref. [16], but an entropy time series, which enables us to evaluate the impact of the entropy-metric's choice for longitudinal observation. We can give insight into the question of what is the most appropriate choice for PCA's input for anomaly detection. Furthermore, we could evaluate the correlation of the appropriate sketch size and the number of IP addresses by making good use of the longitudinal observation of Internet traffic.

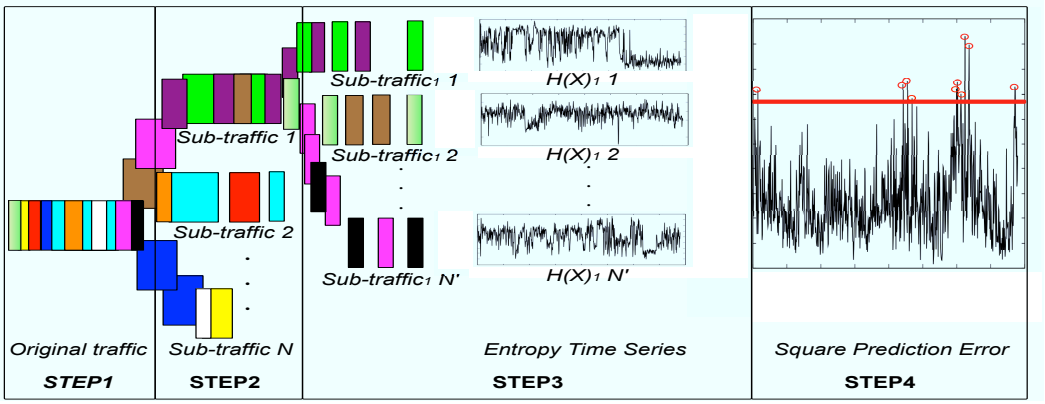


Figure 1: Proposed anomaly detection method. This schematic shows the steps of the proposed anomaly detection procedure for the one hashing function from an IP packet time series, to an aggregated times series of the hashed traffic (step 1), then using another hash function to aggregate the times series of the hashed traffic again in step 2 (The same process for a 2-N aggregated times series is omitted for brevity); then calculating the entropy time series of each sub-traffic in step 3; in step 4, using the linear operator to project the entropy times series into a residual subspace. The source IP addresses of the corresponding sub-traffic need to be saved if the residual vector (squared magnitude) exceeds the threshold. Step 5 is not shown for brevity, but it takes the intersection of all the source IP lists derived from h_m

3. Methodology

3.1. Principal Component Analysis (PCA) and subspace method

Principal Component Analysis (PCA) is a famous coordinate transformation technique to explain the characteristics of high dimensional data by reducing the dimensionality of the data. This technique is often used to detect the network wide anomalies that span multiple links[7, 9, 18] as well as various kinds of multivariate data analysis. PCA maps a given set of data points (e.g. N' dimensional traffic time series of traffic feature $X = t \times N'$) onto N' new axes called principal components. On condition that we deal with zero mean data, each principal component is drawn that it points in the direction of maximum variance remaining in the data, given the variance already accounted for in the preceding components. In order to do that, we first compute the correlation matrix $\mathbf{Y} = (1/t)\mathbf{X^T X}$. As such, the first principal component captures the greatest variance of the data ($\mathbf{v_1} = argmax||\mathbf{Yv}||$ when $||\mathbf{v}|| = 1$). The eigenvalue corresponding to $\mathbf{v_1}$ is λ_1. The next principal

components and successive ones from 2 to N' capture the maximum variance among the orthogonal directions $\mathbf{v_{N'}} = argmax||(\mathbf{Y} - \mathbf{\Sigma_{i=1}^{N'-1} Y v_i v_i^T})\mathbf{v})||$. Therefore, the principal components $(\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_{N'}})$ can be ordered by the amount of data variance $(\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_{N'})$ that they capture. Principal components $(\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_{N'}})$ can be derived either by correlation matrix or variance-covariance matrix. In this paper, we implemented an easily programable method called power method which uses correlation matrix to compute the principal components.

Subspace method can decompose the set of traffic measurement $\mathbf{y}$ into normal $\hat{\mathbf{y}}$ and anomalous $\widetilde{\mathbf{y}}$ state $(\mathbf{y} = \hat{\mathbf{y}} + \widetilde{\mathbf{y}})$. Previous work [4] has shown that the traffic timeseries have low intrinsic dimensionality. Thus, when we apply PCA to the network anomaly detection, the first few principal components (top_k components chosen as described in Section 3.3) explain the normal condition of the network since top_k components capture a large proportion of the variance in the data and k can be a small number. When the normal state of the traffic has been gleaned from the traffic traces, the residual components from k to N' called residual subspace are then treated as anomalous components. The projection of the principal components to $\hat{\mathbf{y}}$ and $\widetilde{\mathbf{y}}$ is described in Section 3.3.

3.2. Sketches

Sketch technique is a random projection method that enables to precisely identify the underlying causes of anomalies[9, 10]. In order to accomplish the identification, we first sketch (randomly shuffle) the traffic by different hashing functions (universal classes of hashing function) using traffic feature such as source IP address as a key. Then we apply the subspace method for each hashed traffic and detect anomalous time points and corresponding source IP addresses. Since each sketch randomly shuffles anomalies (anomalous source IP addresses) across different hashing entries, approximate agreement among sketches can be used to identify the anomalies per host granularity.

3.3. Anomaly detection method

Our proposed method is based on the theory for the PCA-based fault detection in multivariate process control [3] and its applications [7, 9, 18] incorporating sketches [9, 10]. ADMIRE is closely related to the method used in [9], but differs in one important aspect: it deals with packet-based traces captured from a single link by accurately identifying the anomalous source IP addresses. The basic mechanism underlying ADMIRE is three sketches:

The first sketch is for identifying the source IP addresses detected by each hash function; The second sketch is for reducing the total number of source IP addresses that the detected anomalous time points correspond to; The third sketch computes the PCA. Ref [9] does not conduct the second sketch. A performance comparison between two-step and three-step sketches will be discussed in Section 5.2. As shown in Figure 1, ADMIRE consists of the following steps in more detail. We define two kinds of ADMIRE that rely on different entropy time series. Let us denote ADMIRE-A (Step 1–Step 4-a–Step 5) for a separate entropy time series as PCA's input and ADMIRE-B (Step 1–Step 4-b–Step 5) for the combined entropy of the source/destination IP and port as PCA's input. Moreover, let us distinguish ADMIRE-A based on which IP header's information we use. Let us denote ADMIRE-A-1, ADMIRE-A-2, ADMIRE-A-3, and ADMIRE-A-4 corresponding to the entropy of the source IP address, destination IP address, source port, and destination port, respectively. The symbols for the detection procedure are listed in Table 1.

Step 1 (1st hash): Random projections (sketches). The packets from the original traffic are analyzed within the sliding time windows of duration T. For each time window, let $\{t_i, x_{i,l}, l = 1, ..., 4\}$ denote the usual 5-tuple (arrival time stamp, SrcIP, DstIP, SrcPrt, DstPrt). Let $h_m, m \in 1, ..., M$ denote the M independent universal hash functions generated from different random seeds. The original traffic is divided into N sub-traffics by the M hash functions presenting the source IP addresses as hashing key A_i. If we present the source IP addresses as the input to h_m, the return value (hash value) would be a large integer. We divide the integer by N and the remainder plus one becomes the sub-traffic's identifier (from 1 to N). We can allocate each packet to *sub-traffic1, ..., sub-trafficN* using h_m based on the packet's source IP address. Thus, for each h_m, the original traffic $\{t_i, x_{i,l}, l = 1, ..., 4\}$ is split into N sub-traffics, $\{t_i, n_{m,i} = h_m(A_i) = n, i = 1, ..., I\}_{n,m}$.

Step 2 (2nd hash): Sketches using different hashes. The *sub-traffic1, ..., sub-trafficN* are divided once again into *sub-traffic*$_n$1, ...,*sub-traffic*$_n N'$ ($n \in 1, ..., N$) by another hash function in order to apply the PCA to each *sub-traffic1, ..., N*. We use JSHash for this purpose. Accordingly, for each h_m, we have $N \times N'$ sets of sub-traffics denoted as $\{t_i, n'^{n,m}_i = h_m(A_{n,i}) = n', i = 1, ..., I\}_{n,n',m}$. We omit the image from the *sub-traffic2* to *sub-trafficN* processes in Figure 1 to save space.

Table 1: The symbols for the detection procedure

Symbol	Description
$t_i(i \in 1, ..., I)$	time series of original traffic (I=# of packets)
$x_i(i \in 1, ..., I)$	value of each packet's unique feature (e.g., SrcIP, DstPrt)
$n \in 1, ..., N$	output number of 1st sketch (N= size of hash table)
$n' \in 1, ..., N'$	output number of 2nd sketch (M= size of hash table)
$m \in 1, ..., M$	sketch number (M= # of hash functions)
$A_i(i \in 1, ..., I)$	hashing key (SrcIP) of each packet
V	least vote to report alarms (V= # of hash functions we use)
X	traffic feature (SrcIP, DstIP, SrcPrt, or DstPrt)
$H'(X)$	entropy of feature X
$H(X)$	normalized entropy of feature X
s	total number of feature X
s_0	number of distinct value x_i presented in T
$\mathbf{y}$	matrix of entropy of sub-traffic $\{t_i, n'^{n,m}_i = h_m(A_{n,i}) = n', i = 1, ..., I\}_{n,n',m}$ at any time point
$(\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_{N'}})$	principal components derived from $\mathbf{y}$
k	number of normal components ($< N'$)
$\mathbf{I}$	identify matrix of size N'
CP	cumulative proportion of principal components to decide # of normal components
$\mathbf{P}$	matrix of normal principal components (size: $N' \times k$)
$\mathbf{P^T}$	transposed matrix of $\mathbf{P}$
$\widetilde{\mathbf{C}} = (\mathbf{I} - \mathbf{PP^T})$	linear operator to project axes to residual subspace
$\|\|\widetilde{\mathbf{C}}\mathbf{y}\|\|^\mathbf{2}$	square prediction error to capture abnormal changes in $\mathbf{y}$ time series
θ	parameter of threshold to report anomalous time point

Step 3: Computation of entropy. As defined in ref. [5, 6], the entropy of packet feature X at a given time point is $H'(X) = -\sum_{i=1}^{s} p(x_i) log(p(x_i))$, where s is the total unique number of the features. We use the normalized entropy denoted as $H(X) = \frac{H'(X)}{log(s_0)} (0.0 \leq H(X) \leq 1.0)$, where s_0 is the number of distinct x_i values presented in the given time bin T. We studied the empirical distribution of 4-tuple. The entropy time series of the source IP addresses, destination IP addresses, source ports, and destination ports are denoted as H(SrcIP), H(DstIP), H(SrcPrt), and H(DstPrt) here after. For each *sub-traffic*$_n$1, ...,*sub-traffic*$_n N'$ ($n \in 1, ..., N$), we compute

$H_{n,1}(X), H_{n,2}(X), ..., H_{n,n'}(X)$ $(n \in 1, ..., N, n' \in 1, ..., N')$, where X= SrcIP, DstIP, SrcPrt, or DstPrt. For each feature x_i, we compute the probability

$$p(x_i) = \frac{\#\, packets\ with\ x_i\ as\ SrcIP, DstIP, SrcPrt, or, DstPrt}{Total\ \#\, packets\ during\ T}$$

$H(X) = 0.0$ explains the concentration of X while $H(X) = 1.0$ explains the dispersion of X, where X= SrcIP, DstIP, SrcPrt, or DstPrt.

Step 4-a: PCA anomaly detection from each entropy time series (ADMIRE-A). Let $\mathbf{y} = (y_1, y_2, ..., y_{N'})$ denote the H(X) of *sub-traffic*$_n$1, ...,*sub-traffic*$_n N'$ $(n \in 1, ..., N)$ at any time step, where X is SrcIP, DstIP, SrcPrt, or DstPrt. The projection of all the sub-traffics onto the residual subspaces can be performed by the linear operator $\widetilde{\mathbf{C}} = (\mathbf{I} - \mathbf{P}\mathbf{P}^{\mathbf{T}})$, where the $N' \times k$ matrix $\mathbf{P}$ represents the normal subspaces $(\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_k})$ (k denotes the number of normal components) and $\mathbf{I}$ is the $N' \times N'$ identity matrix , which is consistent with the method in ref. [7]. For $a \in 1, ..., N'$, the principal component $\mathbf{v_a}$ can be derived from the time series of $\mathbf{y}$ by using the power method. The number of first k principal components(top_k) is known to be a sensitive parameter, which can be tuned to maintain the high detection rate and low false alarm rate. We decide the top_k based on the cumulative proportion of each sub-traffic. In detail, we use from one to $k-1$-th principal components to compose normal subspaces when the cumulative proportion of the k-th principal component exceeds $CP\%$. We will discuss this CP in Section 5.1. A useful statistic for detecting abnormal changes in $\widetilde{\mathbf{C}}\mathbf{y}$ is the Square Prediction Error (SPE). We applied the Median Absolute Deviation (MAD) [12] for detecting any abnormal changes in $\widetilde{\mathbf{C}}\mathbf{y}$. We chose MAD rather than $mean + 3 \times \sigma$ because MAD is more robust to the distribution of $\widetilde{\mathbf{C}}\mathbf{y}$. Even when $\widetilde{\mathbf{C}}\mathbf{y}$ cannot be approximated to the Gaussian distribution, the threshold of MAD would be set to a valid value for anomaly detection. We consider the time point to be anomalous if SPE $= ||\widetilde{\mathbf{C}}\mathbf{y}||^{\mathbf{2}} > Median + \theta \times MAD$, where θ denotes the threshold in the rest of this paper. The SrcIPs and the anomalous time points of each *sub-traffic1, ..., sub-trafficN* are registered to a list. Since we use M hash functions overall, we have M SrcIP lists.

Step 4-b: PCA anomaly detection from combined entropy time series (ADMIRE-B). We concatenate the H(SrcIP), H(DstIP), H(SrcPrt), and H(DstPrt) values of *sub-traffic*$_n$1, ...,*sub-traffic*$_n N'$ $(n \in 1, ..., N)$ at any

time point and denote them as follows:

$$\mathbf{y} = (y_1, y_2, ..., y_{N'}, y_{N'+1}, ..., y_{2N'}, y_{2N'+1}, ..., y_{3N'}, y_{3N'+1}, ..., y_{4N'}).$$

The projection of the sub-traffics onto the residual subspaces can also be performed by the linear operator $\widetilde{\mathbf{C}} = (\mathbf{I} - \mathbf{P}\mathbf{P}^{\mathbf{T}})$, where the $4N' \times k$ matrix $\mathbf{P}$ represents the normal subspaces $(\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_k})$ (k denotes the number of normal components) and $\mathbf{I}$ is the $4N' \times 4N'$ identity matrix. For $i \in 1, ..., 4N'$, the principal component $\mathbf{v_a}(a \in 1, ..., 4N')$ can be derived from the $\mathbf{y}$ time series by using the power method. The rest of this step is consistent with Step 4-a.

Step 5: Take the intersection of all hash functions. We use the intersection of M source IP address lists in order to specify the anomalous SrcIPs (omitted in Figure 1). We count the number of votes for M hash functions and report the attack if the V out of M lists' report the same SrcIP and time point as anomalous.

3.4. Anomaly classification method

The many researchers trying to quantitatively characterize backbone link traffic are facing exactly the same difficulty–there is no ground truth to confirm whether the detected anomalies are real anomalies with malicious intentions. However, it is inevitable to label the attacks in the traces to fairly evaluate our detection method.

In this paper, we rely on the heuristics used in refs. [10, 11], which is a combination of a typical approach based on the port numbers and a state-of-the-art method based on the communication structure to take advantage of both merits. We classify events into five categories (attack, special, unknown, warning, and benign) in advance. Table 2 lists examples of the heuristics used to label the SrcIPs to be classified into five categories. Once the SrcIPs are labeled as an attack and non-attack (warning, special, benign, and unknowns), we can match them with the SrcIPs detected by ADMIRE, the gamma method, and the KL method. We will evaluate both the attack and non-attack events detected by each method.

3.5. Evaluation method

We define the criteria to evaluate the anomaly detectors. A popular method to evaluate an anomaly detector is a simple comparison of the true

Table 2: Examples of heuristics. The port number and communication structure are used to decide whether the source IP address is an attacker.

Category	Explanation	Example of Heuristics
Attack	Sender of malicious packets	More than 20% of all packets are SYN flagged from computer
		Sasser: true of above condition and communicated with many peer hosts when 50% of packets' DstPrt is 445, 5554, or 9898
Warning	Slightly suspicious host	50% of all packets are HTTP requests
		P2P: Communicated with many peer host using many higher ports to higher ports
Special	Sender of high DNS, FTP, MAIL, SSH, PROXY, etc.	50% of packets is DNS, FTP, MAIL, SSH, PROXY, etc. related traffic
Benign	Sender of legitimate traffic	50% of all packets are from SrcPrt 80
Unknown	Unknown type of communication	Hosts that are not true of any heuristics

(TPR) and false positive rates (FPR). However, it is also important to evaluate the detection accuracy (DA) of the anomaly detectors at the same time to confirm whether they perform more accurately than randomly picking out packets from the trace. In order to efficiently evaluate the DA as well as the TPR and FPR of the detectors, we use the following criteria.

True Positive Rate (TPR): The TPR is defined as the quotient of (the number of attack SrcIPs detected by ADMIRE) divided by (the number of attack SrcIPs classified by the heuristics).

False Positive Rate (FPR): The FPR is defined as the quotient of (the number of non-attack SrcIPs detected by ADMIRE) divided by ((the number of SrcIPs in the trace) - (the number of attack SrcIPs in the trace)).

Detection Accuracy (DA): The DA is defined as the quotient of (the number of attack SrcIPs detected by ADMIRE) divided by (the number of SrcIPs detected by ADMIRE).

F-measure: In this paper, we compute the *F-measure* as the weighted harmonic mean of recall (i.e., TPR) and the precision (i.e., DA) in order to

balance the TPR and DA of the anomaly detection. The *F-measure* is defined as $F\text{-}measure = \frac{2 \times TPR \times DA}{TPR+DA}$. The basic idea underlying the *F-measure* is that a higher value represents a better anomaly detection performance.

Euclidean distance in ROC curve: To simultaneously compare the TPR and FPR, we create the Receiver Operating Characteristic (ROC) curves (x-axis: FPR, y-axis: TPR) and measure the Euclidean distance from the optimal point $(x, y) = (0.0, 1.0)$. Thus, the Euclidean distance (ED) is defined as $ED = \sqrt{(FPR)^2 + (1 - TPR)^2}$. Note that a lower value represents a better detector performance.

Euclidean distance in *(FPR, TPR, DA)*: To evaluate the anomaly detector's TPR, FPR, and DA, we measure the 3-dimensional Euclidean distance (we call it 3DED hereafter) from optimal point $(FPR, TPR, DA) = (0.0, 1.0, 1.0)$ and defined as $3DED = \sqrt{(FPR)^2 + (1 - TPR)^2 + (1 - DA)^2}$. Note that a lower value represents a better detector performance.

4. Data set

We used real backbone link traffic traces from the MAWI traffic repository to evaluate ADMIRE. It has been providing raw packet traces (15-min pcap traces from 14:00 to 14:15 JST) collected for over ten years (from 2001-2012) at one of the trans-Pacific links called samplepoint-B (18 Mbps, a committed access rate on a 100 Mbps link) between Japan and the United States. Samplepoint-B was replaced by samplepoint-F (a full 100 Mbps link) in July 2006 and upgraded in July 2007 (a full 150 Mbps link). We analyzed five weekday-traces for the month of April from 2001-2009. Even though these 15-min traces are not consecutive (every 15 minutes per day), we can sufficiently evaluate our algorithm's macroscopic (e.g., on a year granularity) performance since the measurement location and start time do not change. Table 3 lists the traffic trace information (observation period, the average number of five days' total packets, throughput, the average number of unique IP address for five days).

We only deal with IP addresses that generated more than five hundred packets because a larger amount of packets allows us to identify any abnormalities in the detected traffic. We have also changed the minimum number of packets we deal with in the experiments (e.g., one hundred packets) although with similar results. Thus, we only display the results from the IP addresses that generated more than five hundred packets for brevity.

Table 3: Data set.

Obs. period (dd/mm/yyyy)	02-06/04/2001	01-05/04/2002	07-11/04/2003
Avg. # of packets	2996811.4	4997004.6	3661983.6
Throughput (Mbps)	19.278	12.538	12.194
Avg. # of unique IP addresses	101798.8	151839.8	200167.2
Obs. period (dd/mm/yyyy)	05-09/04/2004	04-08/04/2005	17-21/04/2006
Avg. # of packets	7459728.8	6698640.4	9033659
Throughput (Mbps)	27.662	21.95	31.884
Avg. # of unique IP addresses	865426.6	497298.2	458657.4
Obs. period (dd/mm/yyyy)	02-06/04/2007	07-11/04/2008	06-10/04/2009
Average # of packets	18437985.6	25197334.2	19241447.5
Average throughput (Mbps)	101.29	147.512	112.32
Average # of IP addresses	517920.6	825909.4	740003.5

5. Evaluation

5.1. Preliminary investigation of parameter dependency

ADMIRE has several parameters to be tuned including the sketch sizes ($N \times N'$) in steps 1 and 2, the CP and threshold (θ) in step 4, and the number of votes (V) to report the anomalies in step 5 in Section 3.3. Since our focus is to compare the anomaly detectors using longitudinal observation for nine years, we fixed the parameters of ADMIRE as well as comparative method referring to the trace in 2001. The advantages of this longitudinal analysis are to evaluate the detectors with diverse anomalies and understand the robustness of the detectors to different types of anomaly. However, selecting optimal parameters for the numerous traces for analysis is a laborious task. Since the traffic characteristics (e.g. the number of IP addresses) substantially vary over time, the optimal parameters also fluctuates according to the input traffic traces. This is a common issue faced by researchers in anomaly detection. In this article we evaluated the anomaly detectors using the same parameter settings for all the traces from 2001-2009 in order that the parameter's robustness to the traffic fluctuations could also be considered. The following subsections disclose how we explored the parameter space.

Sketch sizes (i.e., size of hash tables): The sketch sizes ($N \times N'$) should be carefully set by taking two important key elements into consideration:

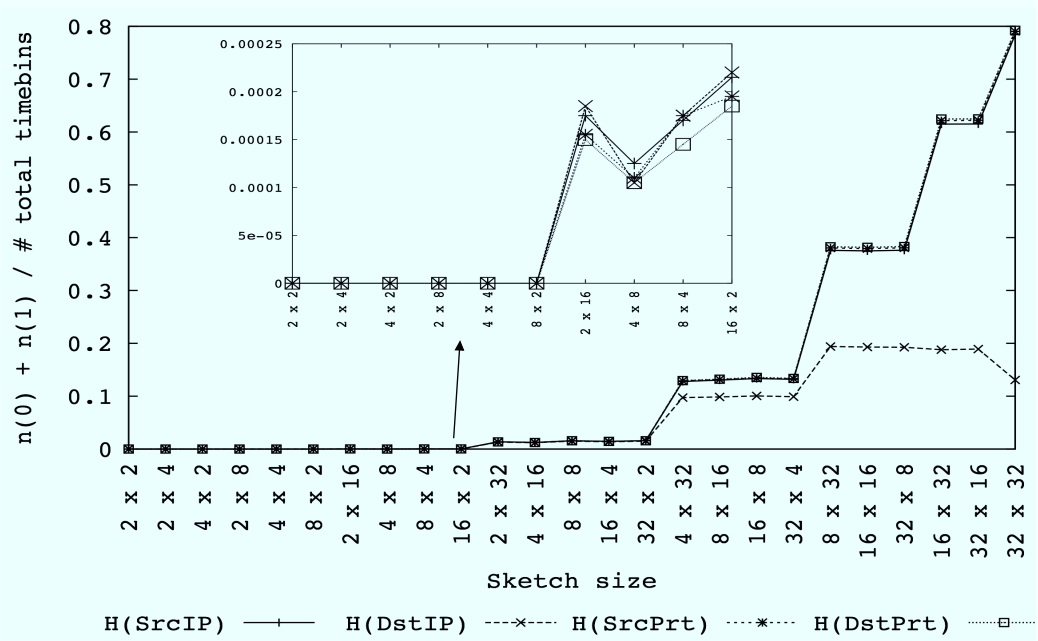


Figure 2: Sketch size vs. Number of zero and one value / number of total time bins (02/04/2001).

the shape of the sub-traffic's entropy time series and the number of principal components we can acquire for the separation of normal and residual subspaces. Sketch sizes that are too large would lead to sparse sub-traces consisting of only a few packets for a time bin. If only one or two packets exist in a particular sub-traffic, the entropy discretely takes 0.0 or 1.0 values since one packet is regarded as a concentration and two different packets are regarded as a dispersion in terms of the entropy. Accordingly, only a few packets is not enough to obtain a reasonable entropy value for explaining the normal and anomalous states of the traffic and are unsuitable for anomaly detection. As shown in Figure 2, the number of one- and zero-value entropies ($n(0) + n(1)$) increases as a function of the sketch sizes except for H(DstIP). H(DstIP) is the most robust to large sketch sizes because DstIP's space is larger than those of the other features since we sketch the traffic by using the SrcIP (port space is much smaller than address space). Since the number of zero- and one-value entropies start to dramatically increase when $N \times N' \geq 64$, the sketch sizes $N \times N'$ must be set to less than 64.

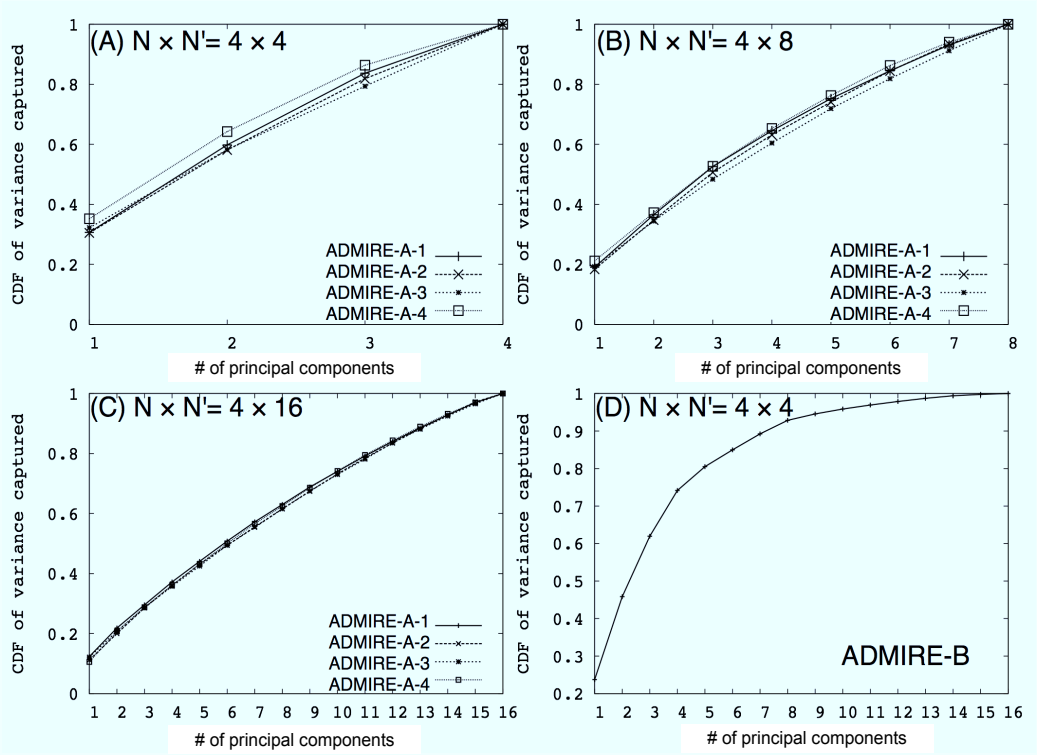


Figure 3: CDF of variance captured by principal components (02/04/2001).

On the other hand, if the sketch size N' is too small, we occasionally cannot use the subspace method. For example, when N' is set to 2, the first principal component captures most of the variance in the sub-traffic. We the regard the 0 to $k-1$ components as a normal subspace, where a 1 to k principal component's cumulative proportion exceeds $CP = 70\%$ for instance, the two principal components are plotted onto the anomalous subspace, which does not make sense for the purpose of the separation of normal and anomalous subspaces. Figure 3 shows ADMIRE-A's cumulative distribution of the variance captured by each principal component when the sketch sizes are $N \times N' = 4 \times 4$ (A), 4×8 (B), 4×16 (C), and ADMIRE-B's CDF ($N \times N' = 4 \times 4$) (D) using the trace for 02/04/2001 (The CDF's shape is quite similar even if we choose other traces). Note that these graphs are examples out of N CDFs since we conducted PCA on N sub-traffics. Additionally, $N' = 4$ derives 16 principal components from ADMIRE-B because

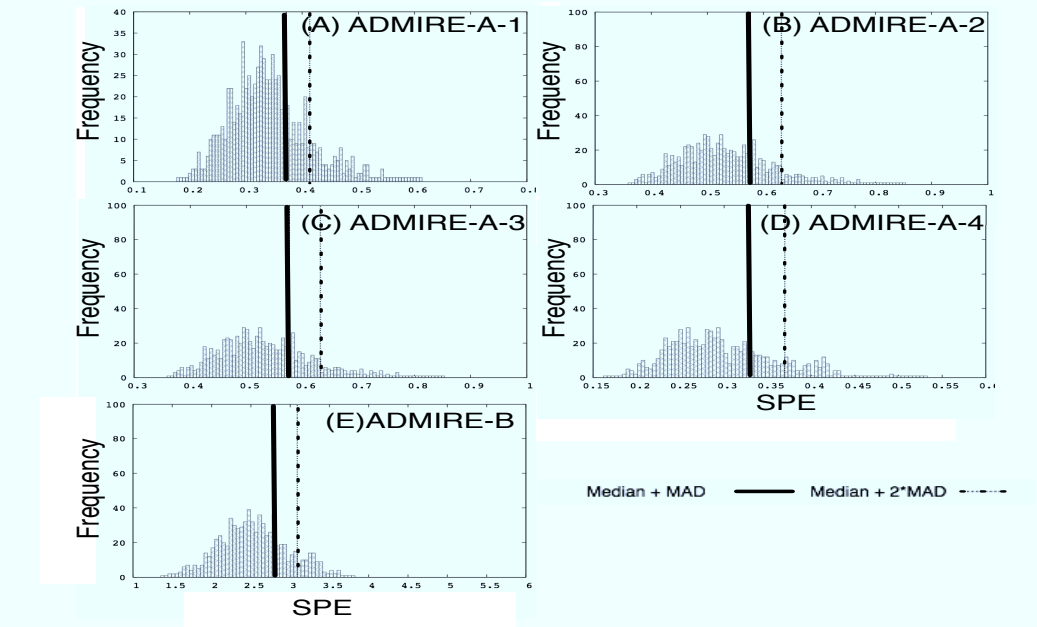


Figure 4: Examples of histograms for residual vector squared magnitude (02/04/2001).

its input to PCA is four kinds of entropy time series (H(SrcIP), H(DstIP), H(SrcPrt), and H(DstPrt)). ADMIRE-B's first few principal components capture most of the variance in the data when $N \times N' = 4 \times 4$, as shown in Figure 3 (D). We chose the 4×4 sketch sizes for the experiments because larger sketch sizes would increase the one- and zero-value entropies for all the metrics, as shown in Figure 2. The larger sketch sizes yield similar CDFs among different entropy features on 02/04/2011, as shown in Figure 3(B) and (C), which would not be suitable for investigating the difference in the types of anomalies captured by each entropy metric.

Cumulative proportion of principal components: We use 1 to $(k-1)$-th principal components to be plotted onto the normal subspace where the cumulative proportion (CP) captured by the k-th principal components exceeds 70% for the evaluation. CP was empirically decided to be 70% based on the results in ref. [8, 16]. Ref [8] suggests that the Cattell's scree test [20] (principal component's variance based decision of top_k) should currently be the best method to use to decide the top_k. We conducted a scree test for

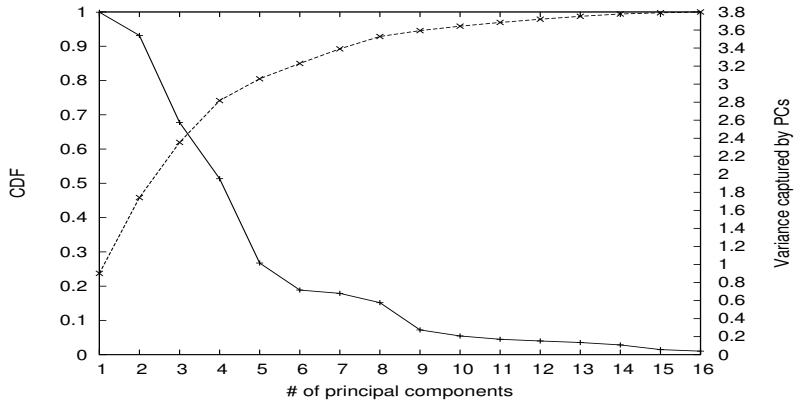


Figure 5: CDF and scree plot captured by using principal components (02/04/2001).

our data set and concluded that the top_k when the principal component's cumulative variance exceeds $CP = 70\%$ was most frequently equal to the top_k of the scree test's knee for our data set (Figure 5 is a typical example), thus we decided $CP = 70\%$.

Threshold for anomaly detection by Square Prediction Error (SPE): We examined the histogram of SPE for each entropy feature to properly set the detection threshold. As discussed in Section 3.3, sketch sizes $N \times N'$ construct N sets of SPE time series. Thus, if we use M hash functions, we have $N \times M$ sets of SPE time series. For each entropy metric (H(SrcIP), H(DstIP), H(SrcPrt), and H(DstPrt)), we have 32 histograms when $N = 4$ and $M = 8$. We plot only one histogram for each entropy feature out of 32 histograms from 02/04/2001 to save space, but note that the shapes of the distributions basically remain the same for the 32 histograms. We present the SPE histograms of ADMIRE-A and ADMIRE-B in Figure 4. Since the valid threshold differs depending on the observation period of the data because we used real backbone traces, we empirically used two thresholds, $\theta = 1$ (loose threshold) and $\theta = 2$ (strict threshold) to capture both anomalies with significant deviation and anomalies with subtle deviation in the SPE. For (E) ADMIRE-B in Figure 4, the loose threshold $Median + MAD$ is 2.981 residual vector squared magnitude while $Mean + \sigma$ is 2.971 residual vector squared magnitude. For the strict threshold, $Median + 2 \times MAD$ is 3.168 residual vector squared magnitude while $Mean + 2 \times \sigma$ is 3.113 residual vec-

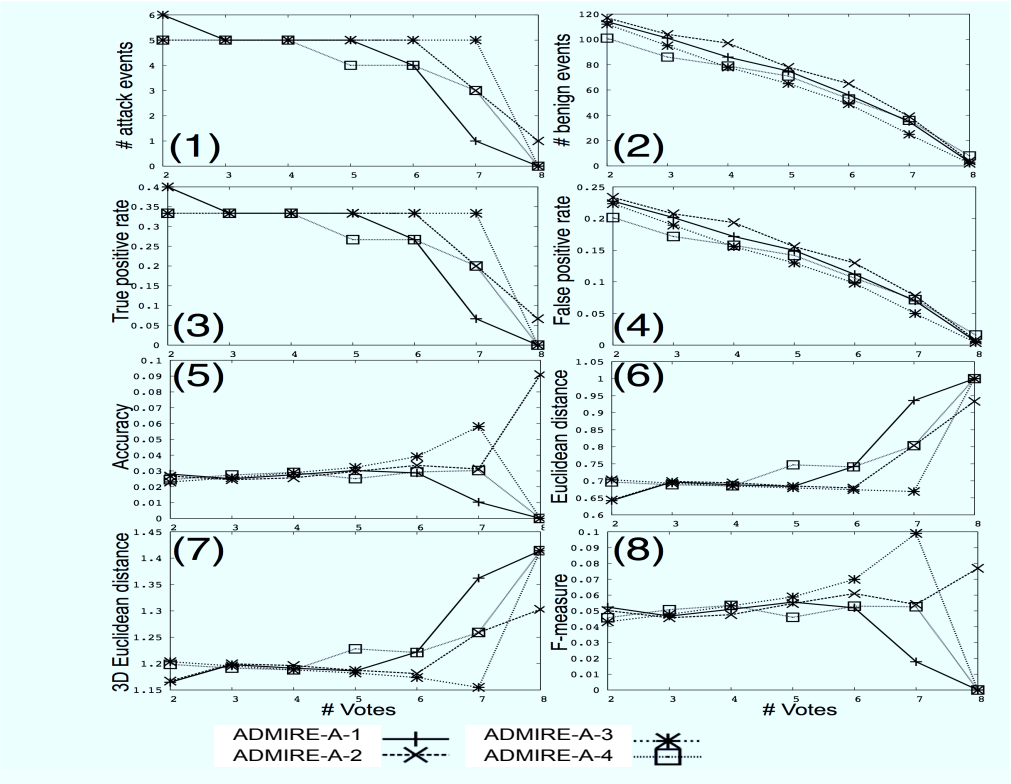


Figure 6: (1) Number of attacks, (2) number of benign events (3) TPR, (4) FPR (5), DA (6) ED, (7) 3DED, and (8) F-measure of ADMIRE-A when $\theta = 1$ (02/04/2001).

tor squared magnitude. Considering the fact that about 68% of the values drawn from a normal distribution are within one standard deviation σ away from the mean; about 95% of the values lie within two standard deviations; practically, the larger θ scarcely detects the anomalies; $\theta = 1, 2$ must be a reasonable choice for the threshold to fairly evaluate the performance of ADMIRE compared to the other detectors.

Voting schemes: The number of votes (V) required to report the anomalies is used in ADMIRE in order to adjust the balance of the TPR, FPR, and DA. Figure 6 shows (1) the number of attacks, (2) the number of non-attack events, (3) TPR, (4) FPR, (5) DA, (6) ED, (7) 3DED, and (8) the F-measure for different numbers of votes V for ADMIRE-A when the thresh-

Table 4: Number of votes setting for each entropy metric

	A-1	A-2	A-3	A-4	B
$\theta = 1$	5	6	7	4	7
$\theta = 2$	2	5	4	2	3

Table 5: Results from packet-based PCA of two and three-step sketches when $\theta = 1.0$ and $\theta = 2.0$ from 2001 to 2009

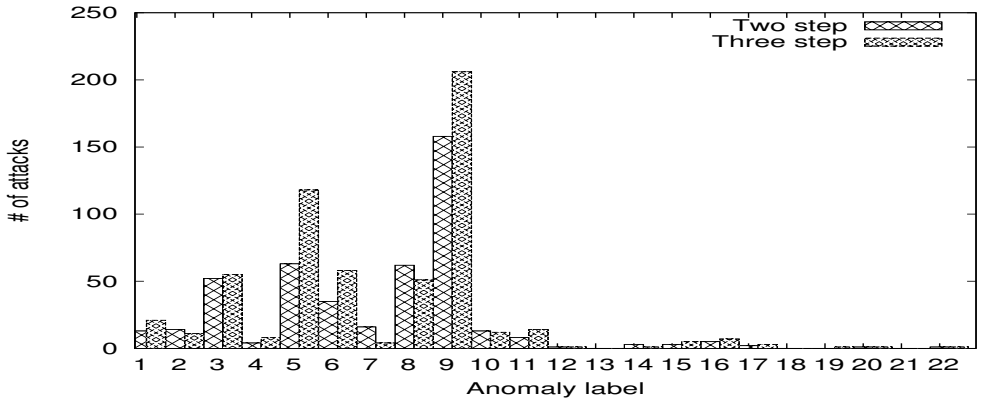
AD type	θ	attack	non-attack	TPR	FPR	ED	DA	3DED	F-val.
2 step	1.0	454	8081	0.155	0.0019	0.845	0.053	1.27	0.0792
3 step	1.0	578	9350	0.198	0.0022	0.802	0.0582	1.24	0.0899
2 step	2.0	224	4437	0.077	0.001	0.923	0.048	1.33	0.0591
3 step	2.0	316	3729	0.108	0.00088	0.892	0.0781	1.28	0.0907

old $\theta = 1.0$ (02/04/2001). If we only take into consideration Figure 6-(5) DA for ADMIRE-A-2, we may choose $V = 8$. However, as can be seen in Figure 6-(1), only one attack event is reported when $V = 8$. Thus, we have to take into consideration (3) TPR and (4) FPR as well as (5) DA. As (7) 3DED seems to give us the most suitable number of votes since 3DED takes into account the TPR, FPR, and DA simultaneously, we decide on the number of votes setting based on (7). Table 4 lists the number of votes setting for the evaluation of ADMIRE in Section 5.4. Table 4 indicates that the strict threshold value ($\theta = 2$) lowers the appropriate value of V.

5.2. Two-step sketch vs. Three-step sketch

In order to experimentally show that our proposed three-step sketch algorithm outperforms the two-step sketch algorithm proposed in ref. [9], we compared the ED, 3DED, and F-measure as well as the types of attacks detected by both the two- and three-step sketch algorithms using 9-year traces from 2001 to 2009. We set the voting parameter $V = 8$ for both methods in order to capture only the conspicuous anomalies. As discussed in Section 5.1,

Figure 7: Types of attacks detected by two and three-step sketches.



even though parameter tuning for each trace considering traffic characteristics (e.g. the number of IP addresses) would higher the detection capabilities, we fixed the parameter for both 2- and 3- step sketches because we focus on the evaluation of the robustness of the detectors to different types of anomaly. As can be seen in Table 5, the three-step sketches took a higher F-measure and lower ED and 3DED than the two-step sketches. This means that our three-step sketch method outperforms the two-step sketch under the ED, 3DED and F-measure for both loose and strict thresholds. We confirmed that our proposed approach (three-step sketch) detects more attacks than the previous approach (two-step approach) with almost the same number of false positives.

We have also investigated the types of attacks detected by the two- and three-step sketches. Each detected attack is classified into 22 categories in Appendix A Table A.7 by the heuristics, which is consistent with the anomaly label in Figure 7. The biggest difference between the two- and three-step sketches is that anomaly labels 5 (many connections less than 5 packets), 6 (Sasser worm), 9 (looking for network open ports), and 11 (network scan for Microsoft MySQL) can be better detected by the three-step sketches in Figure 7. However, anomaly labels 7 (network scan for MS File/LPTR share) and 8 (network scan for undefined port) are better detected by the two-step sketches. On the other hand, almost the same number of anomaly labels 1 (SYN flood), 2 (sending many SYN/ACK), 3 (target real server), 4 (scanning for ftp servers), 10 (flooding, source spoofed with destination IP), 12 (scan all ports of a computer), and 14 (sending much not-connected/termination

Table 6: Results of six anomaly detections (# of attacks, # of non-attacks, TPR, FPR, ED, DA, 3DED, & F value with different thresholds) for 2001-2009

AD type	thres.	atk	non-atk	TPR	FPR	ED	DA	3DED	F-val.
ADMIRE-A-1	$\theta = 1.0$	1042	18708	0.356	0.00433	0.644	0.0528	1.145	0.092
ADMIRE-A-2	$\theta = 1.0$	933	15373	0.319	0.00356	0.681	0.0572	1.163	0.097
ADMIRE-A-3	$\theta = 1.0$	738	11333	0.252	0.00262	0.748	0.0611	1.2	0.0984
ADMIRE-A-4	$\theta = 1.0$	1123	21082	0.384	0.00488	0.616	0.0506	1.132	0.0894
ADMIRE-B	$\theta = 1.0$	712	11827	0.243	0.0027	0.757	0.0567	1.209	0.0921
Packet PCA	$\theta = 1.0$	578	9350	0.198	0.0022	0.802	0.0582	1.237	0.0899
Gamma	$\alpha = 1.0$	707	14812	0.242	0.0035	0.758	0.0456	1.219	0.0767
KL	σ	601	11918	0.205	0.0028	0.7945	0.048	1.24	0.0778
ADMIRE-A-1	$\theta = 2.0$	785	14914	0.268	0.00345	0.732	0.05	1.199	0.0843
ADMIRE-A-2	$\theta = 2.0$	349	5668	0.119	0.00131	0.881	0.058	1.29	0.078
ADMIRE-A-3	$\theta = 2.0$	556	8887	0.1901	0.0021	0.81	0.0589	1.242	0.09
ADMIER-A-4	$\theta = 2.0$	745	14981	0.255	0.0035	0.745	0.0473	1.21	0.0799
ADMIRE-B	$\theta = 2.0$	545	10507	0.186	0.00243	0.814	0.0493	1.251	0.078
Packet PCA	$\theta = 2.0$	316	3729	0.108	0.000863	0.892	0.0781	1.283	0.0907
Gamma	$\alpha = 1.3$	299	6056	0.102	0.00142	0.898	0.047	1.31	0.0644
KL	3σ	194	4287	0.066	0.001	0.934	0.0433	1.337	0.0524

TCP traffic) are detected by both sketches.

Three-step sketches detected more types of anomalies with less false positives than the two-step sketches because there might be hidden anomalies that do not deviate from the sub-traffic divided once by the two-step sketches but that do deviate from the sub-traffic divided twice by the three-step sketches. This fact insists that the number of sketches we divide the traffic by will be a factor in catching the slight change in traffic and will possibly improve the detection capabilities of sketch-based anomaly detectors.

Reversely, it might be easier for the three-step sketches to miss the network scan events than that for the two-step sketches since the network scan events accompanied by a large number of packets (more than 10,000 packets) contaminate the normal components because the main characteristic of the traffic at the time point is anomalous. Since three step sketches conduct PCA much more times than the two-step sketches, it would be more vulnerable to the events. However, these events can be detected by using a simple mechanism such as the detection of the packet count time series, and we make more of the capability to detect elaborate attacks.

5.3. Impact of PCA's input entropy for anomaly detection

In order to investigate the impact of different time series as the input to PCA for anomaly detection, we evaluated ADMIRE and the packet-based PCA proposed in our previous work [16] using the data set from 2001 to 2009. Interestingly, despite the combination of four entropy metrics, ADMIRE-B displayed neither the highest F-measure nor the lowest 3DED, as outlined in Table 6. Instead, ADMIRE-A-2 and 3 at $\theta = 1.0$, ADMIRE-A-1, 2, 3 and packet-based PCA at $\theta = 2.0$ had higher F-measures than ADMIRE-B. ADMIRE-A-3 performed the best under the F-measure criteria at $\theta = 1.0$. It detected 26 additional attacks and 494 less false positives compared to ADMIRE-B.

If we compare the results for each threshold, even though a strict threshold ($\theta = 2.0$) detects less false positives than a loose one ($\theta = 1.0$), it degrades the performance under all ED, 3DED, and F-measure criteria as $\theta = 2.0$ decreases the number of true positives more rapidly than the false positives for ADMIRE. On the other hand, the packet count-based PCA's DA increased when we increased the threshold even though the TPR was not as high as ADMIRE.

Interestingly, the attack types also differ depending on which time series we choose for the PCA. Figure 8 shows that ADMIRE-A-1, 2, and 4 detected more attack events that look for open network ports than ADMIRE-B (anomaly label is corresponding to Appendix A Table A.7). ADMIRE detected more SYN flood attack event, sent many SYN/ACK, target servers, many connections less than five packets, and network scans for undefined ports than packet count-based PCA.

Even though a larger θ decreases the DA in ADMIRE, it still detects more types of attacks than the packet count-based PCA. Thus, we should choose the input time series to the PCA based on the objective of the anomaly detection. For instance, if a user wants to detect anomalies with significant spikes in a time series without less mistakes, then a packet count-based PCA with a very strict threshold would be suitable. On the other hand, if we want to detect various types of anomalies with loosely permitted limits on false alarms, ADMIRE would be more suitable. In addition, we also suggest that the false alarms reported by ADMIRE should also contain suspicious communications that were not profiled in the heuristics. We observed the IP header spaces of such communications in Section 5.4.2.

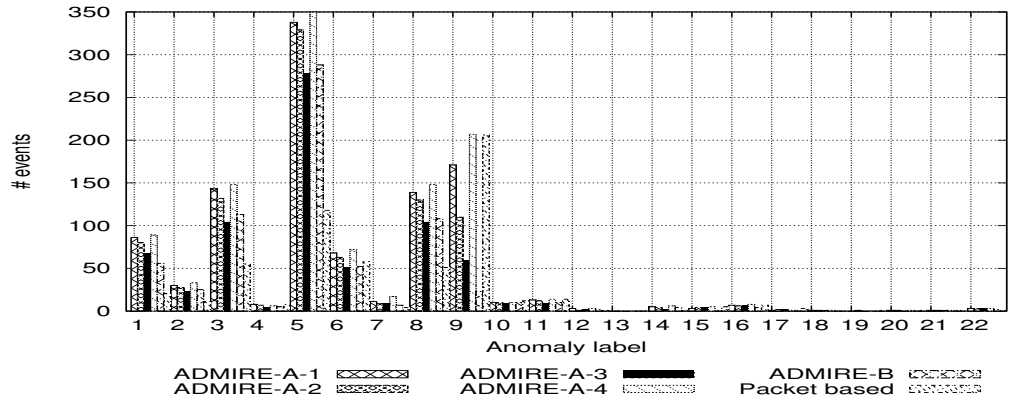


Figure 8: Categorization of attacks by different PCA's inputs for 2001-2009 ($\theta = 1.0$).

5.4. Comparison with other anomaly detectors

To validate ADMIRE as well as the heuristics used in this paper, it is important to compare ADMIRE with other anomaly detectors from different theoretical backgrounds. For this purpose, we compared the evaluation results of ADMIRE with those of two other methods, the gamma method [10] and the KL method [19] using nine-year traces. These two methods both use the sketch technique to identify the anomalous IP flows just like ADMIRE does although each method uses different types of histograms in their detection phase. Thus, we can fairly evaluate each anomaly detection's capabilities by comparing the detected anomalies and false positives. We used two kinds of thresholds. As loose threshold in which we used $\theta = 1.0$ for ADMIRE, one standard deviation σ of the KL distance time series for the KL method (see [19] in details), and $\alpha = 1.0$ (see [10] in details). As a strict threshold, we used $\theta = 2.0$ for ADMIRE, 3σ of the KL distance time series for the KL method, and $\alpha = 2.0$ for the gamma method. Table 6 indicates that ADMIRE outperformed the gamma and KL methods in terms of the ED, 3DED, and F-measure for both the loose and strict thresholds. In particular, ADMIRE-A-3 detected 31 more attacks and 3479 less false positives than the gamma-based method when $\theta = 1.0$ and $alpha = 1.0$. ADMIRE-A-3 also detected 137 more attacks and 585 less false positives than the KL method when we used $\theta = 1.0$ for ADMIRE and σ for the KL method as the thresholds. The true positive rates of the anomaly detectors for evaluation seem to be relatively low because our focus is to compare the anomaly

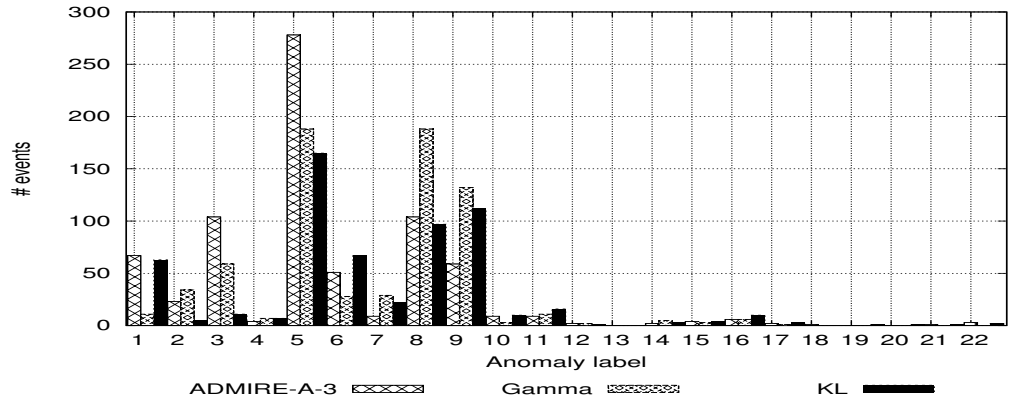


Figure 9: Categorization of detected attacks by three different anomaly detectors for 2001-2009 ($\theta = 1.0$).

detectors using longitudinal observation for nine years. Since the traffic characteristics (e.g. the number of IP addresses) substantially vary over time, the optimal parameters also fluctuates according to the input traffic traces. This is a common issue faced by researchers in anomaly detection. We notice that when the parameters are optimally tuned for a short-term traffic observation the detection rate is greatly improved. For instance, using a week traces in 2005, we could have more promising results for ADMIRE-B (TPR: 0.691, FPR: 0.0061, detection accuracy: 0.31) outperforming the other two best-tuned comparative methods (gamma method ($\alpha = 1.0$): TPR 0.461, FPR 0.01, DA 0.0968, KL-method (3σ of the KL distance time series): TPR 0.352, FPR 0.0056, DA 0.134).

Under the condition where these anomaly detectors reported about the same number of events, ADMIRE performed the best because there were hidden anomalies that did not deviate from the sub-traffic divided once for the sketch in the other two methods but deviated from the sub-traffic divided twice for the three-step sketches in ADMIRE. We explain such hidden anomalies in Section 5.4.1.

5.4.1. Analysis of attack events

We are also interested in the difference in attack types reported by ADMIRE, the gamma method, and the KL method. We classified the detected attacks into 22 categories listed in Appendix A Table A.7 by the heuristics,

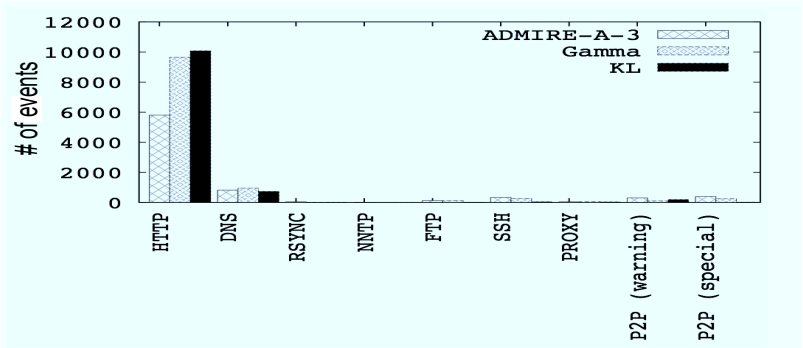


Figure 10: Categorization of detected non-attack events for 2001-2009 ($\theta = 1.0$).

which correspond to the "Anomaly label" in Figure 9. Figure 9 shows that ADMIRE reported events labeled 3 (Target Realserver) and 5 (many connections less than 5 packets) more than the other two methods. About the same number of label 1 (TCP SYN flood) was detected by ADMIRE and the KL method while the gamma-based method detected less. Labels 8 (Network scan for undefined port) and 9 (looking for network open ports) were best detected by the gamma method. Labels 6 (Sasser (dstprt: 445, 5554, 9898)), 11 (Network scan for MS MySQL), and 16 (Scanning for SSH servers) were best detected by using the KL method.

Since many connections less than five packets (label 5) was prominently detected by ADMIRE, we investigated the SrcIP vs. DstIP/DstPrt space of the events detected by ADMIRE-A-3. We plotted the IP header space of the events detected by ADMIRE-A-3 at $\theta = 2.0$ in Figure 11. We confirmed that the event seems to be an elaborate low-profiled scan that sends packets to a lot of DstIPs using a lot of DstPrts from many different source hosts. ADMIRE tends to detect this type of coordinated low-profiled attack. This type of hidden anomaly, which does not deviate from the traffic, has been difficult to automatically detect. ADMIRE made it possible to detect these anomalies and we believe that it was a decided improvement over its predecessor.

5.4.2. Analysis of non-attack events

It is also important to conceive of the detected type of non-attack events detected by ADMIRE, the gamma method and the KL method. Figure 10

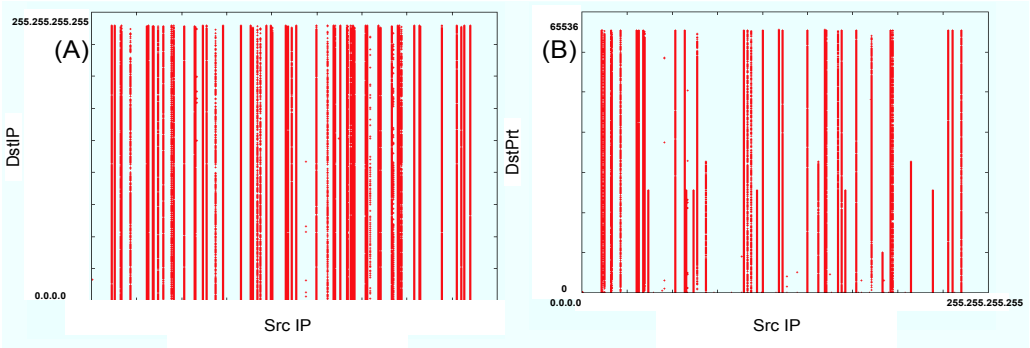


Figure 11: (A) Source IP address vs. Destination IP address, (B) Source IP address vs. Destination port of many connections less than five packets detected by ADMIRE-A-3.

shows that high HTTP, DNS, RSYNC, NNTP, FTP, SSH, PROXY traffic, warning P2P (many peers, many higher ports to higher ports), and special P2P (Many peers, using port 4444, 7777, etc) are mistakenly reported as anomalies by three anomaly detectors with loose thresholds. ADMIRE-A-3 is less likely to report the HTTP traffic as anomalous compared to the gamma and KL methods. The gamma-based method is not inclined to detect P2P traffic.

In order to clarify the reason why this traffic has been reported as anomalous by ADMIRE-A-3, we have investigated the SrcIP vs. SrcPrt space of the detected events. As a result, we confirmed that the WWW server, HTTP client, DNS traffic, and SSH traffic detected by ADMIRE-A-3 included the communication patterns that might be mistaken as the dispersion of the SrcPrt entropy, and thus, the source hosts would be detected.

By investigating the IP header space of the non-attack events detected by ADMIRE-A-3, we confirmed that the detected events commonly present the densely plotted lines in the SrcIPs vs. SrcPrt space shown in Figure 12. One of our future works is to investigate the abnormality or dangerousness of traffic by using payload matching.

5.5. Number of unique source IP addresses vs. appropriate sketch sizes

One of the benefits of using the MAWI data set is that we can explore the parameter space for a longitudinal observation. In order to fairly compare our method with the gamma-based method, we use one set of sketch sizes in our evaluation, but we assume that the most appropriate sketch sizes differ

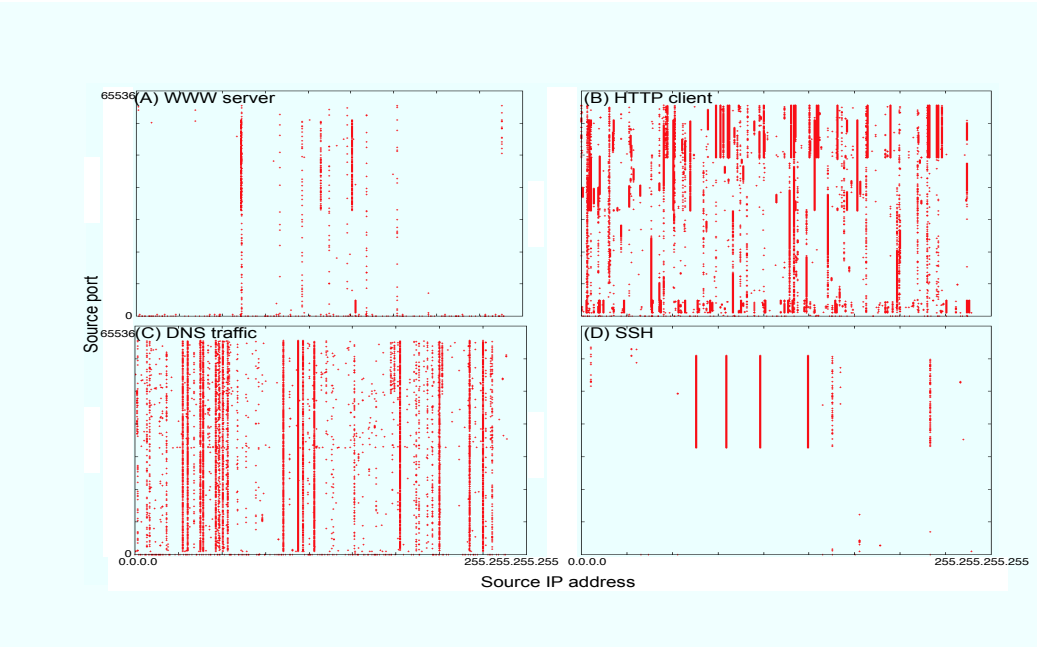


Figure 12: Source IP address vs. source port of (A)WWW server, (B) HTTP client, (C) DNS traffic, and (D) SSH traffic detected by ADMIRE-A-3 ($\theta = 2.0$) for 2001-2009.

depending on the number of unique IP addresses in a trace because the sketch sizes greatly affect the shape of the entropy time series as evaluated in Section 5.1. In this section, we discuss how to decide the appropriate sketch sizes for different periods of observation with different numbers of unique SrcIPs.

Figure 13 shows that the number of unique SrcIPs in a trace versus the most appropriate sketch sizes under the F-measure criteria for the traces from sample point B. The results for 2004 are not shown since a lot of packets in the trace were contaminated by the Sasser activity [17] and the ratio of the attack events to the total is conspicuously high. Thus, the appropriate parameters including the sketch sizes differ from what is usual. Instead, we used the data sets 07-11/07/2003, 10-14/11/2003, 17-21/11/2003, 24-28/11/2003, and 12-16/06/2006 for our evaluation. These traces' ratio of attack is not notably high and is thus suitable for the evaluation of the appropriate sketch sizes. We changed the sketch sizes from $N \times N' = 2 \times 4$ to 32×32 and plotted the log-scale product of $N \times N'$ that took the highest F-measure value. Despite a few outliers, most of the appropriate sketch sizes were positively correlated

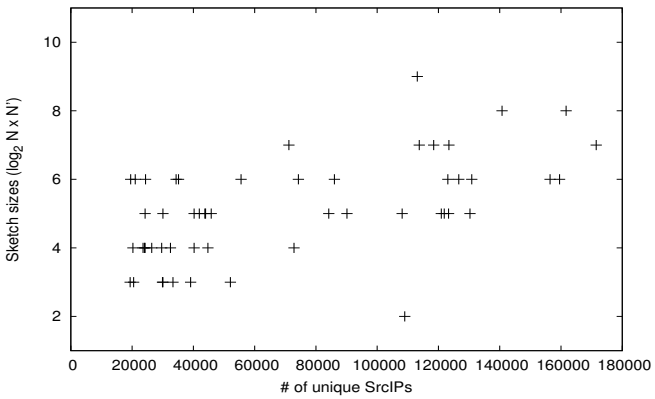


Figure 13: F-measure based appropriate sketch sizes vs. Number of unique SrcIP that generated more than 500 packets for 2001, 2002, 2003, 2005, and 2006 traces.

with the number of unique SrcIPs that generated more than five hundred packets in the trace (correlation coefficient: 0.57). Therefore, the sketch sizes should be decided based on the number of unique SrcIPs in a trace.

A lot of anomaly detectors use a sketch technique and setting the sketch sizes is one of the most crucial keys to successful anomaly detection. We recommend the network operators preliminarily check the number of source IP addresses before using sketch-based anomaly detectors.

5.6. Time and space complexity of ADMIRE

ADMIRE requires a time complexity of $O(hNtN'^2n)$ and space complexity $O(hNtN'n)$ (h: the number of hash functions, N: the number of first sketch size, N': the number of second sketch size, t the number of total time bin, and n: the number of distinct feature such as SrcIP (objective of the empirical entropy) in Section 3.3). PCA requires a Singular Value Decomposition (SVD) whose time complexity is $O(tN'^2)$ and a space requirement of $O(tN')$ because SVD is computed through a $t \times N'$ matrix. ADMIRE also calculates the entropy time series, which requires a time complexity of $O(n)$.

In practice, we can calculate the N sets of the SVD matrix using different CPUs in parallel, and thus, the time and space complexity of ADMIRE can be theoretically improved to $O(htN'^2n)$ and space complexity $O(htN'n)$. In practice, ADMIRE-A took about 2.5 minutes for execution using a 15 min. trace on a laptop PC (Core2Duo, 2.53 GHz, 4 GB of RAM).

6. Conclusion

We presented ADMIRE, a new anomaly detection method that analyzes traffic traces captured at a single link. The novelty in ADMIRE's design is the extra sketch step that allows to compute PCA with single link traces. Moreover, unlike in the previous method, we separately tested each entropy metric's influence upon the type of detected events using nine-year traces. We could also test the parameters of ADMIRE using these longitudinal observation of traffics.

Our main finding is as follows. (1) The proposed method using three-step sketches outperforms the previous two-step sketches in terms of the true and false positive rates. The idea of increasing the number of steps we use to sketch the traffic can be applicable to all anomaly detections that use the sketch technique. (2) We could observe the positive correlation between the appropriate sketch sizes and the number of unique IP addresses in a trace. To our understanding, this is the first investigation into the appropriate sketch sizes using real backbone traces with different numbers of IP addresses. We believe that this correlation is useful to automatically select the sketch sizes of anomaly detectors and it deserves more attention in the future works. (3) Since the entropy time series of different IP header information captures different types of anomalies, their combined usage might be effective, but we need to carefully choose the entropy combination because it might degrade the detection capabilities. (4) We could also confirm that ADMIRE is superior to the gamma and KL methods in terms of the TPR, FPR, and DA.

7. Acknowledgement

We would like to express our cordial gratitude to Guillaume Dewaele for providing us the source code of the gamma-based method.

References

[1] F. Silveira, C. Diot, N. Taft, and R. Govindan, "ASTUTE: Detecting a Different Class of Traffic Anomalies," In ACM SIGCOMM 2010, pp.267-278, New Delhi, India, Aug. 2010.

[2] M. Roesch, "Snort -Lightweight Intrusion Detection for Networks,", USENIX LISA'99, pp.229-238, Nov. 1999.

[3] R. Dunia and S. J. Qin, "Multi-dimensional Fault Diagnosis Using a Subspace Approach," In American Control Conference, Jun. 1997.

[4] A. Lakhina, M. Crovella, and C. Diot, "Structural analysis of network traffic flows," In ACM SIGMETRICS 2004, pp.61-72, New York, NY, USA, Jun. 2004.

[5] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang, "An empirical evaluation of entropy-based traffic anomaly detection" In ACM SIGCOMM IMC 2008, pp.151-156, Vouliagmeni, Greece, Oct. 2008.

[6] A. Lakhina, M. Crovella, and C. Diot, "Mining Anomalies Using traffic Feature Distributions, " In ACM SIGCOMM 2005, pp.217-228, Philadelphia, PA, Oct. 2005.

[7] A. Lakhina, M. Crovella, and C. Diot. "Diagnosing Network-Wide Traffic Anomalies," In ACM SIGCOMM, Portland, Aug. 2004.

[8] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for Traffic Anomaly Detection," In ACM SIGMETRICS 2007, pp.109-121, San Diego, CA, Jun. 2007.

[9] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina, "Detection and Identification of Network Anomalies Using Sketch Subspaces," In ACM IMC 2006, pp.14-19, Rio de Janeiro, Brazil, Oct. 2006.

[10] G. Dewaele, K. Fukuda, P. Borgnat, P. Abry, and K. Cho, "Extracting Hidden Anomalies using Sketch and Non Gaussian Multiresolution Statistical Detection Procedures," In ACM SIGCOMM LSAD 2007, pp.145-152, Kyoto, Japan, Aug. 2007.

[11] Y.Himura, K.Fukuda, K.Cho, and H.Esaki, "An Evaluation of Automatic Parameter Tuning of a Statistics-based Anomaly Detection Algorithm," International Journal of Network Management, pp.295-316, vol.20, no.5, Wiley, 2010.

[12] M. H. Arshad and P. K. Chan, "Identifying Outliers via. Clustering for Anomaly Detection," Florida Institute of. Technology, Department of Computer Sciences. Technical Report CS-2003-19, 2003.

[13] B. Krishnamurty, S. Sen, Y. Zhang, and Y. Chen, "Sketch-based Change Detection: Methods Evaluation and Applications, "In ACM IMC 2003, pp. 234-247, Oct. 2003.

[14] C. Callegari, S. Giordano, M. Pagano, and T. Pepe, "On the use of sketches and wavelet analysis for network anomaly detection," In ACM IWCMC 2010, pp. 331-335, Caen, France, Jul. 2010.

[15] S. Pukkwanna and K. Fukuda, "Combining Sketch and Wavelet models for Anomaly Detection," In IEEE ICCP 2010, Romania, Aug. 2010.

[16] Y. Kanda, K. Fukuda, and T. Sugawara, "An Evaluation of Anomaly Detection Based on Sketch and PCA," In 2010 IEEE GLOBECOM 2010, p.5, Florida, USA, Dec. 2010.

[17] R. Fontugne, P. B., Patrice Abry, K. Fukuda. "MAWILab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking," CoNEXT 2010, p.12, Philadelphia, USA, Dec. 2010.

[18] Y. Liu, L. Zhang, and Y. Guan, "Sketch-Based Streaming PCA Algorithm for Network-Wide Traffic Anomaly Detection," In ICDCS 2010, Genoa, Italy, pp.807-816, Jun. 2010.

[19] D. Brauckhoff, X. Dimitropoulos, A. Wagner, and K. Salamatian. "Anomaly extraction in backbone networks using association rules," IMC 2009, Chicago, USA, pp.28-34, 2009.

[20] R. B. Cattell, "The scree test for the number of factors," Multivariate Behavior Research 1, pp.245-276, 1966.

[21] M. A. Kramer, "Nonlinear Principal Component Analysis Using Autoassociative Neural Networks," Aiche Journal 37(2), pp. 233-243, 1991.

[22] A. Herrero, E. Corchado, P. Gastaldo, and R. Zunino, "A Comparison of Neural Projection Techniques Applied to Intrusion Detection Systems," In IWANN 2007, LNCS 4507, pp.1138-1146, San Sebastian, Spain, Jun. 2007.

Appendix A. Category of attacks

A detailed classification categories of detected attacks are listed in Table A.7. A heuristics based on the port number, TCP flag, and communication pattern classifies the attack events into 22 categories. For instance, if 20% of the IP flows generated by a host is SYN flagged and communicating with many different destination IP addresses with destination ports 445, 5554, or 9898, then the host is classified as an activity of the Sasser worm.

Table A.7: Category of attacks

Anomaly label	Number
SYN flood	1
Sending many SYN/ACK	2
Target Realserver	3
Scanning for FTP servers	4
Many connections less than 5 packets	5
Sasser (dstprt: 445, 5554, 9898)	6
Network scan for MS File/LPTR share (dstprt: 139)	7
Network scan for undefined port	8
Looking for network open ports	9
Flooding, source spoofed with destination IP	10
Network scan for MS MySQL (1433)	11
Scan all ports of a computer	12
The Prayer Trojan	13
Sending much not-connected/termination TCP traffic	14
Network scan for Radmin (remote adm.)	15
Scanning for SSH servers	16
Network scan for Sun RPC (111)	17
Network scan for Redhat SWAT, VMWare, or Net Devil worm (901)	18
Network scan for Appletalk (202)	19
Network scan for Daneware (remote adm.) (6129)	20
Network scan for Limewire (gnutella clone) (6346)	21
Network scan for Milkit trojan / Kuang 2 virus (17300)	22