2014 IEEE 6th International Conference on Cloud Computing Technology and Science

A Wavelet Entropy-based Change Point Detection on Network Traffic: A Case Study of Heartbleed Vulnerability

Chonho Lee, Liu Yi, Li-Hau Tan, Weihan Goh, Bu-Sung Lee and Chai-Kiat Yeo
School of Computer Engineering
Nanyang Technological University, Singapore

***Abstract*—This paper investigates network traffic before and after a vulnerability called Heartbleed becomes a public issue around March to May, 2014. To detect anomalies and potential threats due to the vulnerability, a wavelet entropy-based change-point detection method is proposed and compared with three other methods: prediction-based, clustering-based and Fourier transform-based. We show that the proposed wavelet entropy-based method outperforms the others in terms of ease of parameter setting, false alarm and detection accuracy. Using the proposed method and a visualization tool, we have studied Heartbleed vulnerability and successfully captured changes in packet volume and flow.**

***Keywords*-Cyber-security; Vulnerability; Change-point detection; Network traffic;**

I. Introduction

A vulnerability refers to a bug or hole in software that is unknown to the software vendor. Once the vulnerability becomes known, the developers must quickly rectify the problem. However, attackers exploit the security hole before the vendor rectifies it or releases a patch, known as a zero-day attack. This exploit attempts to infiltrate malware, spyware or allow unwanted access to user information.

In March 21, 2014 or before, Google Security discovers a vulnerability called "Heartbleed", a security bug in OpenSSL cryptography library [1]. The affected versions of OpenSSL allocate a memory buffer for the returned message based on the length field in the requesting message, without proper bounds checking. Thus, the returned message might include the payload, possibly followed by whatever happened is present in the active memory. Pre-emptive defense from such zero-day attacks is a significant challenge.

To try and detect such potential threats, we monitor and analyze network traffic. Our aim is to capture "symptoms" of attacks by recognizing statistical changes in features retrieved from the traffic. Change point analysis is one powerful way to detect such changes or anomalies in time series data. We propose a wavelet-based change-point detection method considering entropy of coefficients. Three other change-point detection methods are investigated and compared; they are prediction-based, clustering-based and Fourier transform-based.

Specifically, in this paper, we discuss the analysis results of network traffic on WIDE network [2] in terms of change-point detection on traffic features (e.g., volume, packet size, etc.) and node behaviors (e.g., flow, popularity, portscan, etc.) As a case study, we have observed some change points that are possibly "symptoms" regarding Heartbleed issue.

The contribution of the paper is as follows.

- We have investigated and compared performance of four different approaches of change-point detection on network traffic, and we have found that the proposed wavelet entropy-based approach outperforms the others in terms of ease of parameter setting, false alarm and detection accuracy.
- Using the proposed approach and a visualization tool, we have studied a real vulnerability called Heartbleed and successfully captured changes in packet volume and flow.

The remainder of the paper is organized as follow. Section II describes three approaches of change-point detection and the proposed wavelet entropy-based approach. Section III compares and evaluates performance of the four different approaches on synthetic dataset. Section IV discusses the analysis result of Heartbleed issue, followed by conclusion.

II. Change-point Detection

This section describes the different types of change-point detection approaches (subsection A-C) and proposes the wavelet entropy-based approach (subsection D), which is a better choice in terms of ease of parameter setting, false alarm and detection accuracy.

A. Prediction-based Approach

Holt-Winters method [3] is a forecasting technique of time series using an exponentially weighted smoothing filter of linear growth and seasonal trends of the time series.

Considered a base B_t, a linear trend L_t and a seasonal factor S_t, the predicted h-ahead value Y_{t+h} is defined by

$$\begin{aligned} Y_{t+h} &= B_t + h \cdot L_t + S_{t+h-s} \\ B_t &= \alpha \cdot (Y_t - S_{t-s}) + (1-\alpha) \cdot (B_{t-1} + L_{t-1}) \\ L_t &= \beta \cdot (B_t - B_{t-1}) + (1-\beta) \cdot L_{t-1} \\ S_t &= \gamma \cdot (Y_t - B_t) + (1-\gamma) \cdot S_{t-s} \end{aligned} \quad (1)$$

where α, β, γ are parameters that satisfy $0 < \alpha, \beta, \gamma < 1$. The greater parameter value means that recent observations

978-1-4799-4093-6/14 $31.00 © 2014 IEEE
DOI 10.1109/CloudCom.2014.78

are given relatively more weight in forecasting than the older observations. On the contrary, the smaller value indicates that the algorithm has a larger smoothing effect. The predicted value would be less responsive to recent changes.

Change points will be captured when the distance between observed and predicted values exceeds the predefined threshold value.

B. Clustering-based Approach

Subsequence clustering is an online clustering technique applied to time series. It deals with a subsequence or a set of samples within a pre-defined period as a data point in a k-dimensional space where k is the number of samples. Individual subsequence is extracted by sliding a window with particular size and step length. The algorithm iteratively classifies the subsequences into clusters.

Vector Quantization (VQ) [4] is one of the widely used subsequence clustering methods. It initially sets the number of clusters and the cluster centroids from some samples. For each subsequence (i.e., a data point) captured by sliding a window, it finds the closest cluster, and the transition of subsequences from one cluster to another along time series is considered as a state change. The cluster centroids are updated over iterations.

The extended VQ [5] was proposed in which users do not have to know the number of clusters in advance. This method initially sets one cluster with its centroid and radius (i.e., a distance threshold). When finding the closest cluster like VQ, it compares distance between the closest cluster and input subsequence. If the distance exceeds the radius, a new cluster would be created, otherwise the input subsequence becomes a member of the closest cluster.

C. Fourier Transform-based Approach

The short time Fourier transform (STFT) represents a sort of compromise between the time- and frequency-based views of a signal. It provides some information about both when and at what frequencies a signal event occurs. However, you can only obtain this information with limited precision, and that precision is determined by the size of the window.

While the STFT is a compromise between time and frequency information, the drawback is that once the time window size is fix, that window size is used for all frequencies. Many signals require a more flexible approach – one where we can vary the window size to determine more accurately either time or frequency.

STFT suffers from the drawbacks of a resolution problem that a narrow window offers good time resolution but poor frequency resolution, and a wide window results in good frequency resolution but poor time resolution.

D. Wavelet-based Approach

The wavelet analysis provides precise measurements regarding when and how the frequency of the signal changes over time [6]. The core of wavelet analysis is wavelet transformation. Wavelet transformation converts signal from the time domain to the time-scale domain (scale can be considered as the inverse of frequency).

Discrete Wavelet Transformation (DWT) regards each feature's values as a discrete time series signal. The signal is decomposed into low frequency signals reflecting the long term trend and high frequency signals reflecting the time-to-time or day-to-day variation from the long term trend. If the high frequency signal shows a spike or a dip on a certain day, we know that the signal value was significantly changed compared to a normal day.

Specifically, DWT decomposes an original signal into a combination of linearly independent basis functions and *wavelet coefficients*. Basis functions are generated by scaling and shifting a *mother wavelet* ψ. The set of basis functions in DWT is defined as $\psi_{j,k}(t) = 2^{\frac{j}{2}} \cdot \psi(2^j \cdot t - k)$ where j and k are scaling and shifting parameters, respectively. Assume the signal is given by the sampled values $S = \{s(t) \mid t = 1, \ldots, M\}$ where M is the total number of samples in the signal. If the signal decomposition is carried out over all scales, $N_J \equiv log_2(M)$, the signal can be reconstructed by $S(t) = \sum_{j=1}^{N_J} \sum_k C_j(k) \cdot \psi_{j,k}(t)$ where $C_j(k)$ are wavelet coefficients. The wavelet coefficients can be interpreted as the residual errors between signal approximations at scale j and $j+1$.

The Proposed Wavelet Entropy-based Approach

We utilize *relative entropy* defined in [7] that time series signals are transformed to the entropy (i.e., carried by wavelet energy.)

The wavelet energy of signal S is defined by

$$E_{total} = \sum_{j=1}^{N_J+1} E_j \tag{2}$$

where

$$E_j = \sum_k |C_j(k)|^2 \tag{3}$$

$$E_{N_J+1} = \sum_k |A_{N_J}(k)|^2 \tag{4}$$

at each scale $j \leq N_J$).

Evaluating the Shannon Entropy [8] on the energy distribution over different scales (i.e., frequency bands) leads to the measurement of wavelet entropy (WE) of signal S [9] defined by

$$WE(S) = -\sum_j \frac{E_j}{E_{total}} \cdot log \frac{E_j}{E_{total}} \tag{5}$$

and normalized as

$$WE = \frac{WE(S)}{log(N_j + 1)}. \tag{6}$$

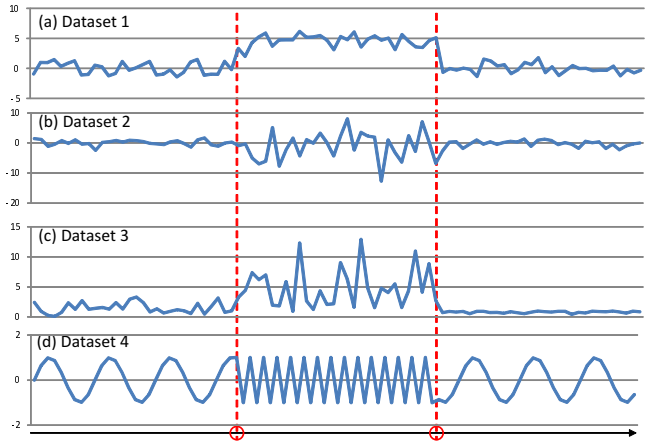

Figure 1. Synthetic dataset: (a) different mean and same variance, (b) same mean and different variance, (c) different probability distribution, (d) different phases. Red circles indicate change points.

Eventually, the relative entropy sequence $S_e = \{s_e(t') \mid t' = \lfloor \frac{t-1}{\Delta} \rfloor + 1\}$ is obtained with the help of a sliding window Δ. At each time period t', we compute $WE_{t'}$ only with sample points within the time period specified by the sliding window. The value of $s_e(t')$ is calculated as

$$s_e(t') = \frac{WE_{t^*} - WE_{t'-1}}{WE_{t'-1}} \text{ if } WE_{t^*} > WE_{t'-1} \qquad (7)$$

and 0 otherwise, where WE_{t^*} is the recent wavelet entropy computed with sample points within the previous two time periods 2*Δ.

If there is no change in $s(t)$ within $\Delta_{t'}$, there will be no significant difference between $s_e(t'-1)$ and $s_e(t')$. On the other hand, an increase or decrease in the volume causes $s(t)$ within $\Delta_{t'}$ to appear increase or decrease. This is translated into an increase or decrease of wavelet entropy. Therefore, $s_e(t')$ encodes how much the change is.

III. Comparative Evaluation

Figure 1 plots synthetic data with varying attributes: (a) different mean and same variance, (b) same mean and different variance, (c) different probability distribution and (d) different phases. The two vertical lines indicate t=31 and t=61, which are the change points in the synthetic data set.

Figure 2 shows the change points detected by the 4 different algorithms. In each subfigure, the four lines respectively correspond to Datasets 1, 2, 3 and 4 shown in Figure 1. "Bump"s on the lines indicate the detected change points. For HW and SC, the x-axis is time while it is window ID for STFT and Wavelet.

Holt-winter algorithm (with window size of 10 and step size of 1) results in a time lag of change-point detection compared to the others. Since generated forecasting points rely on recent data points, it seems that this algorithm works fine for sudden spike or drop (e.g., t=61) but not for gradual changes (e.g., t=36).

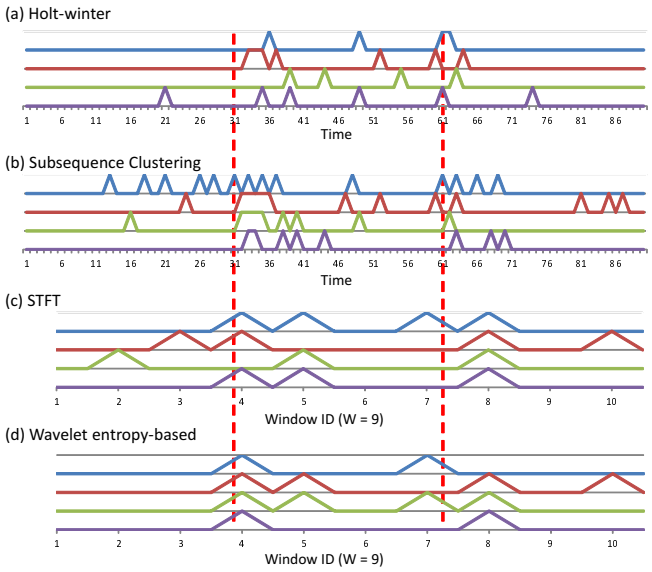

Figure 2. Detected change points by three algorithms.

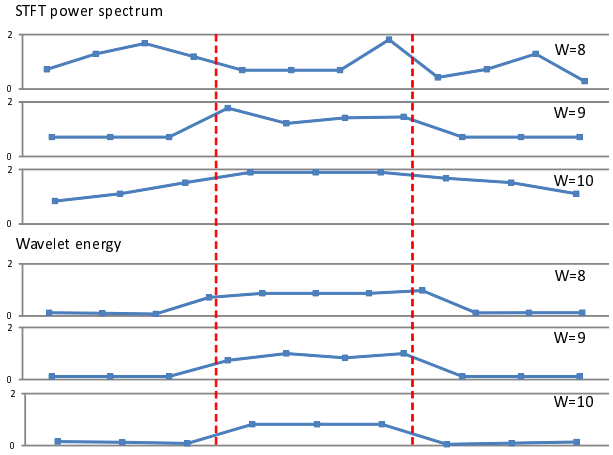

Figure 3. STFT power spectrum (upper subfigures) and wavelet energy (lower subfigures) to Dataset 4 by different window sizes.

Subsequence clustering algorithm (with window size of 10 and step size of 1) correctly detects the change points right after t=31 and 61. However, several false positives are also detected. It seems that this algorithm is sensitive to local data change.

STFT-based method (with window size of 9) and Wavelet entropy-based method (with window size of 9) detect the change points at 4th (t=25-36) period and 7th (t=55-63) or 8th (t=64-72) period. Some change-points at t=61 (i.e. in 7th period) could not be detected because there are only three points were available for computation. However, both

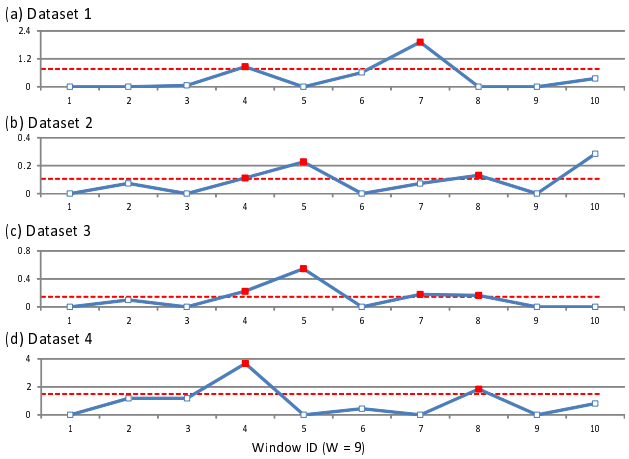

Figure 4. Relative wavelet entropy by the proposed method. The dotted line indicates threshold to determine change points.

methods capture data points in 8th period, totally different from those in 7th period. In addition, they have less false positive points than the others.

Figure 3 plots STFT power spectrum (i.e., the sum of square of coefficients) and Wavelet energy (Equation 4) with different window sizes. The figures show that STFT results are affected by window sizes. In fact, signals for Dataset 4 is generated with frequency of 9 (t=1-30 and 61-90) and 2 (t-31-60). Thus, when the window size is 9, we see a dramatic change in power spectrum and energy around t=31 and t=61. However, when the window size is not 9, STFT does not show the big increase/decrease around t=31/61 (except STFT with W=8) while it is detected with Wavelet energy. It follows that we need to find a proper window size for STFT, but the window size does not have to be accurate for Wavelet-based algorithm.

Figure 4 shows relative wavelet entropy by the proposed method for four synthetic datasets. X-axis is window ID of size 9. The dotted line indicates threshold to determine change points. When the values exceed the threshold, the points are considered as change points, which are illustrated in Figure 2-(d).

IV. Case Study: Heartbleed

In March 21, 2014 or before, Google Security discovers a vulnerability called ”Heartbleed”, a security bug in OpenSSL cryptography library. The affected versions of OpenSSL allocate a memory buffer for the returned message based on the length field in the requesting message, without proper bounds checking. Thus, the returned message might include the payload, possibly followed by whatever happened is present in the active memory. We apply the proposed wavelet entropy-based approach to analyze the change points on network traffic as a consequence of the release of heartbleed information.

A. Heartbleed Timeline

Based on some of the major reported Heartbleed timeline [10], we show the main stream here for space limitation.

1) Friday, March 21 or before - Neel Mehta of Google Security discovers Heartbleed vulnerability.
2) Monday, March 31 or before - Someone tells content distribution network CloudFlare about Heartbleed and they patch against it.
3) Tuesday, April 1 - Google Security notifies ”OpenSSL team members” about the flaw found in OpenSSL, which later becomes known as ”Heartbleed”. Mark Cox at OpenSSL says on social network Google Plus.
4) Wednesday, April 2 - Finnish IT security testing firm Codenomicon separately discovers the same bug that Neel Mehta of Google found in OpenSSL.
5) Friday, April 4 - Rumours begin to swirl in open source community about a bug existing in OpenSSL, according to one security person at a Linux distribution Fairfax spoke to. No details were apparent so it was ignored by most.
6) Saturday, April 5 - Codenomicon purchases the Heartbleed.com domain name, where it later publishes information about the security flaw. OpenSSL (not public at this point) publishes this (since taken offline) to its Git repository.
7) Monday, April 7 - The National Cyber Security Centre Finland (NCSCF) reports Codenomicon’s OpenSSL ”Heartbleed” bug to OpenSSL core team members via encrypted email. A new OpenSSL version is uploaded to OpenSSL’s web server with the filename ”openssl-1.0.1g.tgz”. OpenSSL publishes a Heatbleed security advisory on its website. CloudFlare and Codenomicon post a blog entry about the bug and tweet about its blog post. Most of the world finds out about the issue through heartbleed.com.
8) Monday, April 8 - NCSCF issues a security advisory on its website in English.

B. Observed Dataset

The Measurement and Analysis for the WIDE Internet (MAWI) working group provides a traffic data repository [2], collected from the WIDE backbone networks. The WIDE network is a Japanese academic network connecting universities and research institutes. The MAWI repository has provided anonymized packet traces since 1999.

In this paper, we have investigated header (e.g., source IP and port, destination IP and port, protocol and size) of packets captured daily at Samplepoint-F [2] (of the 150Mbps link) from 2:00pm to 2:15pm in March to May, 2014. The dataset size of packets dumped for 15 minutes (i.e., per day) is in range between 5G and 10G.

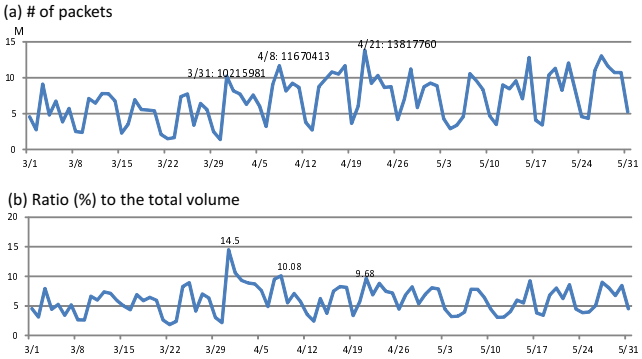

Figure 5. (a) Volume of packets with port 443, (b) Ratio of the packets with port 443 to the total volume

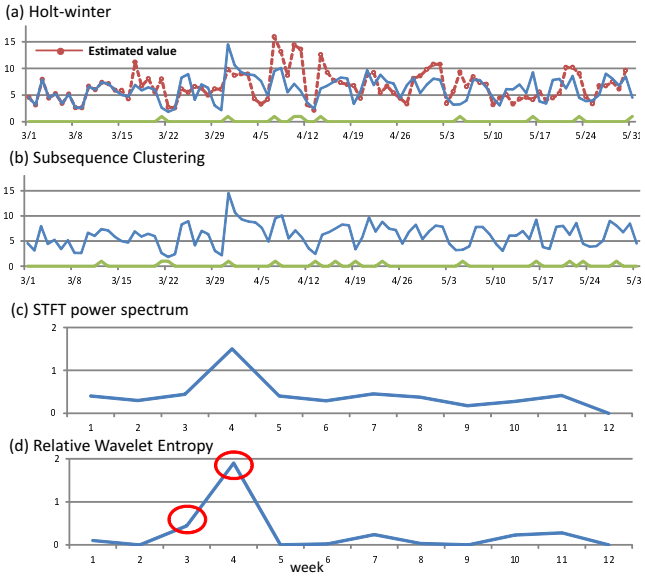

Figure 6. Detected change-points by four different approaches

C. Packet Analysis

We focus on packets with destination port 443. Figures 5-(a) and (b) show the volume of packets with destination port 443 and its ratio to the total packet volume, respectively. We have observed several increases on Monday[1] and a few big spikes (i.e., with over 10% of packets), especially on March 31 and April 8 that correspond to the aforementioned Heartbleed timeline A-2 and A-8. The spike on April 21 might have occurred in the aftermath.

Figures 6-(a) and (b) show the detected change-points by Holt-winter and clustering algorithms, respectively, applied on the packet volume ratio (i.e., Figure 5-(b)). Both algorithms captured the changes on around March 31 and April 8. However, they also detected several change-points that are not along Heartbleed timeline.

On the other hand, STFT-based and the proposed wavelet entropy-based methods successfully captured the big change within 4th week (between 3/29 and 4/4) as shown in Figures 6-(c) and (d). We set the window size, W=7, i.e., a week, for a weekly trend observed in Figures 5-(a) and (b). The result corresponds to Heartbleed timeline from A-2 to A-5 that was ignored by most.

By the wavelet entropy-based method, the smaller change was also detected from the 3rd week (3/22-3/28) corresponding to A-1 while STFT-based method could not distinguish the change clearly. From April 7 (i.e., the 5th week), a new OpenSSL version is uploaded and most of the world started to learn about the issue. The number of packets related to port 443 kept increasing; however, the wavelet entropy remained relatively low.

D. Flow Analysis

Portscan/Netscan is one of the security attacks that involves a remote host scanning TCP ports on victim machines to exploit vulnerable services. Portscan attempts to scan a single host for all open ports to determine what services are currently provided by the host. Netscan attempts to scan a whole branch of network prefix for the same open port. The portscan is generally used by an attacker actively looking for open ports on a target machine, while the netscan is usually performed by a tool (e.g., Nmaps) or a worm on a compromised host, looking for other vulnerable machines. By sending TCP SYN packets to the corresponding port(s), the remote host confirms that a port(s) is open when the local host responds with an ACK.

Figures 7-9 show the frequency of packets captured over the particular time period with regard to source-destination pairs. Different colors indicate different frequencies. Those figures provide us intuitive knowledge about portscan and/or netscan. In the case of portscan, we may see horizontal lines from multiple sources while we may see vertical lines in the case of netscan.

Using such tool, we could also observe the netscan to the port 443. For example, Figure 7 is the plot on March 4, 2014 (2pm to 2:15pm). Filtering out all packets except the ones with port 443, Figure 8 shows the frequency plot on March 4, 2014 (2pm to 2:15pm). Figure 9 is the plot for destination port 443 only on March 31, 2014 (2pm to 2:15pm). Clear volume increase on March 31 has been shown, and the possible netscan to port 443 from at least 4 different sources has been observed in those figures.

V. Conclusion

This paper investigates the performance of four different approaches for change-point detection on network traffic. We have found that the proposed wavelet entropy-based ap-

[1]These increases might be due to the check-up before a patch release on Tuesday, e.g., Microsoft.

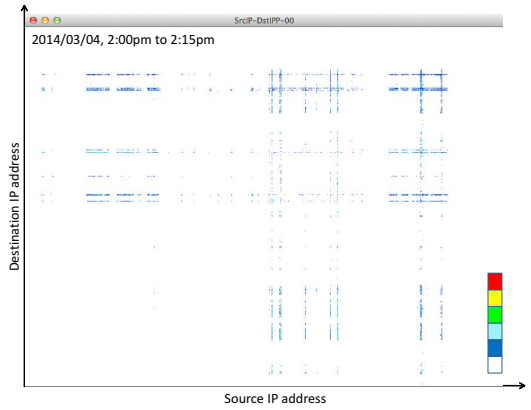

Figure 7. Frequency of all packets on Mar 4, 2014

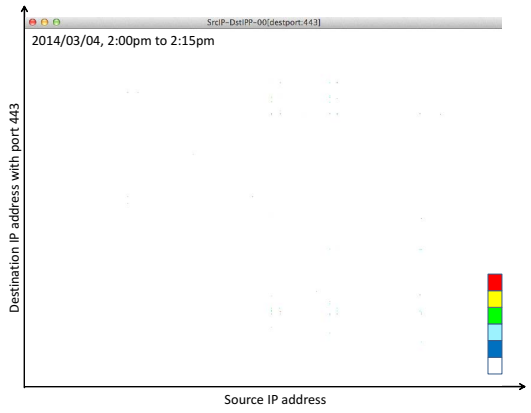

Figure 8. Frequency of packets with destination port 443 on Mar 4, 2014

Figure 9. Frequency of packets with destination port 443 on Mar 31, 2014

proach outperforms the others in terms of ease of parameter setting, false alarm and detection accuracy. We make use of vulnerability called Heartbleed and apply the proposed approach to detect change-points on network traffic.

Further analysis will be conducted by investigating more packets not only daytime but night, different ports other than 443, and their correlations. Besides, the wavelet entropy-based method will be extended to evaluate its performance with different step size. We set the same step size as the window size because we have seen a clear trend on the observed dataset. The proposed method still requires threshold setting. We will consider dynamic and adaptive thresholding technique.

REFERENCES

[1] "Heartbleed blog," Codenomicon Ltd., http://heartbleed.com.

[2] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the wide project," *USENIX 2000*, June 2000.

[3] C. Chatfield, "The holt-winters forecasting procedure," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 27, 1978.

[4] R. Gray, "Vector quantization," *IEEE ASSP Magazine*, 2002.

[5] Y. Kang, "Real-time change detection in time series based on growing feature quantization," *Proc. of International Joint Conference on Neural Networks*, 2012.

[6] G. Kaiser, *A friendly guide to wavelets*. Birkhauser Boston Inc., 1994.

[7] J. Wang and B.-S. Lee, "Event detection in twitter," *ICWSM 2011*, July 2011.

[8] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, 1948.

[9] O. Rosso, "Wavelet entropy: a new tool for analysis of short duration brain electrical signals," *Journal of Neuroscience Methods*, vol. 105, 2001.

[10] B. Grubb, "Heartbleed disclosure timeline: who knew what and when," the Sydney Morning Herald, Apr 2014.