INT J COMPUT COMMUN, ISSN 1841-9836
Vol.7 (2012), No. 3 (September), pp. 509-517

An Entropy-based Method for Attack Detection in Large Scale Network

T. Liu, Z. Wang, H. Wang, K. Lu

Ting Liu
SKLMS Lab and MOE KLNNIS Lab,
Xi'an Jiaotong University
Xi'an, Shaanxi, 710049, P.R.China
E-mail: tingliu@mail.xjtu.edu.cn

Zhiwen Wang, Haijun Wang, Ke Lu
MOE KLNNIS Lab, Xi'an Jiaotong University
Xi'an, Shaanxi, 710049, P.R.China
E-mail: wzw@mail.xjtu.edu.cn
{hjwang,klu}@sei.xjtu.edu.cn

Abstract:
Intrusion Detection System (IDS) typically generates a huge number of alerts with high false rate, especially in the large scale network, which result in a huge challenge on the efficiency and accuracy of the network attack detection. In this paper, an entropy-based method is proposed to analyze the numerous IDS alerts and detect real network attacks. We use Shannon entropy to examine the distribution of the source IP address, destination IP address, source threat and destination threat and datagram length of IDS alerts; employ Renyi cross entropy to fuse the Shannon entropy vector to detect network attack. In the experiment, we deploy the Snort to monitor part of Xi'an Jiaotong University (XJTU) campus network including 32 C-class network (more than 4000 users), and gather more than 40,000 alerts per hour on average. The entropy-based method is employed to analyze those alerts and detect network attacks. The experiment result shows that our method can detect 96% attacks with very low false alert rate.
Keywords: Network Security, Entropy-based, IDS, Shannon Entropy, Renyi Cross Entropy.

1 Introduction

Network attacks are defined as the operations that disrupt, deny, degrade, or destroy information resident in computer networks or the networks themselves. In recent years, more and more network attacks threatened the reliability and QoS of Internet, compromised the information security and privacy of users. KSN (Kaspersky Security Network) recorded 73 million Internet browsers attacks on their users in 2009, and that number skyrocketed to 580,371,937 in 2010 [1]. Symantec reported that they recorded 3 billion attacks from their global sensor and client [2].

Intrusion Detection System (IDS) is used to monitor and capture intrusions into computer and network systems which attempt to compromise their security [3]. With the development of networks, a large number of computer intrusions occur every day and IDSs have become a necessary addition to the security infrastructure of nearly every organization. However, IDSs still suffer from two problems: 1) large amount of alerts. In fact, more than 1 million alerts are generated by Snort each day in our research; 2) high false alerts rate. Gina investigated the extent of false alerts problem in Snort using the 1999 DARPA IDS evaluation data, and found that 69% of total generated alerts are considered to be false alerts [4]. These problems result in a huge challenge on the efficiency and accuracy of the network attack detection.

Several methods have been applied to resolve the problems of large amount of alerts and high false rate. Pietraszek used the adaptive alert classifier to reduce false alerts, which is trained with lots of labeled

past alerts [5]. Whereas, it is difficult to label large volume alerts generated in large-scale network. In order to reduce the false alarms, Mina propose the extend DPCA to standardize the observations according to the estimated means [6]. Spathoulas and Katsikas propose a post-processing filter based on the statistical properties of the input alert set [7]. Cisar employ EWMA to detect attacks by analyzing the intensity of alerts [3]. In our research, 32 C-class subnets are monitored by Snort and more than 1 million alerts are generated every day. Therefore, we propose a method to spot anomalies which is more tolerable for the operator rather than reduce false alerts.

In information theory, entropy is a measure of the uncertainty associated with a random variable, which is widely used to analyze the data and detect the anomalies in information security. Lakhina et al argue that the distributions of packet features (IP addresses and ports) observed in flow traces reveal both the presence and structure of a wide range of anomalies. Using entropy as a summarization tool to analyze traffic from two backbone networks, they found that it enables highly sensitive detection of a wide range of anomalies, augmenting detections by volume-based methods [8]. Brauckhoff ind that entropy-based summarizations of packet and flow counts are affected less by sampling than volume-based method in large networks [9]. A. Wagner and B Plattner applied entropy to detect worm and anomaly in fast IP networks [10]. Relative entropy and Renyi cross entropy can be used to evaluate the similarity of different distributions. Yan et al use a traffic matrix to represent network state, and use Renyi cross entropy to analyze matrix traffic and detect anomalies rather than Shannon entropy. The results show

Renyi cross entropy based method can detect DDoS attacks at the beginning with higher detection rate and lower false rate than Shannon entropy based method [11]. Gu et al proposed an approach to detect anomalies in the network traffic using Maximum Entropy estimation and relative entropy [12]. The packet distribution of the benign traffic was estimated using Maximum Entropy framework and used as a baseline to detect the anomalies.

In this paper, an entropy-based method is proposed to detect network attack. The Shannon entropy and Renyi cross entropy are employed to analyze the distribution characteristics of alert features and detect network attack. The experimental results under actual network data show that this method can detect network attack quickly and accurately. The rest of the paper is organized as follows: the method is introduced in Section 2, and the experimental results are shown in Section 3. Section 4 is the conclusion and future work.

2 Methodology

In this paper, Snort is used to monitor the network and five statistical features of the Snort alert are selected. The Shannon entropy is used to analyze the distribution characteristics of alert that reflect the regularity of network status. When the monitored network runs in normal way, the entropy values are relatively smooth. Otherwise, the entropy value of one or more features would change. The Renyi cross entropy of these features is calculated to measure the network status and detect network attacks.

2.1 Snort Alert and Feature Selection

Each Snort alert consists of tens of attributions, such as *timestamp*, *source IP address* (*sip*), *source port*, *destination IP address* (*dip*), *destination port*, *priority*, *datagram length* and *protocol*, etc. Suppose there are n alerts generated in time interval t. The alerts set in time interval t is denoted as $Alert(t) = \{alert_1, alert_2, \ldots, alert_n\}$.

Assuming there are m distinct *sip* and k distinct *dip* in $Alert(t)$, we can generate the distinct source IP addresses set (*SIP*) and distinct destination IP addresses set (*DIP*):

$$SIP = \{sip_1, sip_2, \ldots, sip_m\},$$

$$DIP = \{dip_1, dip_2, \ldots, dip_k\}.$$

Suppose the number of alerts come from sip_i is $snum_i$, and the number of alerts send to dip_i is $dnum_i$. The alert number of each source IP ($SNUM$) and destination IP ($DNUM$) can be calculated:

$$SNUM = \{snum_1, snum_2, \ldots, snum_m\},$$

$$DNUM = \{dnum_1, dnum_2, \ldots, dnum_k\}.$$

There are 4 default priorities of Snort alert: 1, 2, 3 and 4. The threat severity gradually weakens from 1 to 4(high, medium, low, info). In order to strengthen the threat degree of high severity alerts, the threat degree of the $alert_i$ is denoted as $threat_i = 5^{(4-priority_{alert_i})}$ in present work. Suppose the threat degree sum of all alerts come from sip_i is $sthreat_i$, and the threat degree sum of all alerts send to dip_i is $dthreat_i$. The threat degree of each source IP ($STHREAT$) and destination IP ($DTHREAT$) can be calculated:

$$STHREAT = \{sthreat_1, sthreat_2, \ldots, sthreat_m\},$$

$$DTHREAT = \{dthreat_1, dthreat_2, \ldots, dthreat_k\}.$$

The datagram length is the size of the packet that breaks the alarm rules of Snort. We search the distinct datagram length of all alerts, and generate the datagram length set

$$DGMLEN = \{dgmlen_1, dgmlen_2, \ldots, dgmlen_x\},$$

where x is the number of the distinct datagram length of all alerts. Suppose the number of alerts whose datagram length equal to $dgmlen_i$ is $dgmNum_i$. The alert number with different datagram length can be calculated:

$$DGMNUM = \{dgmNum_1, dgmNum_2, \ldots, dgmNum_x\}.$$

Above 5 features ($SNUM, DNUM, STHREAT, DTHREAT, DGMNUM$) are selected to evaluate the alerts and detect attacks.

2.2 Shannon Entropy-based Feature Analysis

Shannon entropy is used as measures of information and uncertainty [13]. For a dataset $X = \{x_1, x_2, x_3, \ldots, x_n\}$, each data item x belongs to a class $x \in C_x$. The entropy of X relative to C_x is defined as

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

where p_i is the probability of x_i in X.

The distribution characteristics of five features are analyzed using Shannon entropy. The entropies of $SNUM$ and $DNUM$ in time interval t can be calculated

$$H(Sip_t) = -\sum_{i=1}^{m} (snum_i/n) \log(snum_i/n) \tag{2}$$

$$H(Dip_t) = -\sum_{i=1}^{k} (dnum_i/n) \log(dnum_i/n) \tag{3}$$

The entropy of $STHREAT$ and $DTHREAT$ can be calculated:

$$H(Sthreat_t) = -\sum_{i=1}^{m} \frac{threat\_of\_sip(i)}{sum\_threat} \cdot \log\left(\frac{threat\_of\_sip(i)}{sum\_threat}\right) \tag{4}$$

$$H(Dthreat_t) = -\sum_{i=1}^{k} \frac{threat\_of\_dip(i)}{sum\_threat} \cdot \log\left(\frac{threat\_of\_dip(i)}{sum\_threat}\right) \tag{5}$$

where $threat\_of\_sip(i)$ is the threat sum of the alerts from sip_i, $threat\_of\_dip(i)$ is the threat sum of the alerts to dip_i, and sum_threat is the threat sum of all the alerts in ALERTS which can be calculated using

$$sum\_threat = \sum_{i=1}^{n} threat_i \tag{6}$$

The entropy of datagram length is

$$H(Dgmlen_t) = -\sum_{i=1}^{x} (dgmNum_i/n) \cdot \log(dgmNum_i/n) \tag{7}$$

After calculating the entropies of above features, we can use an entropy vector $V(t) = [H(Sip_t), H(Dip_t), H(Sthreat_t), H(Dthreat_t), H(Dgmlen_t)]$ to represent the network status of time interval t.

2.3 Renyi Cross Entropy-based Attack Detection

The Renyi entropy, a generalization of Shannon entropy, is a measure for quantifying the diversity, uncertainty or randomness of a system. The Renyi entropy of order α is defined as

$$H_\alpha(P) = \frac{1}{1-\alpha} \log_2 \sum_r p_r^\alpha \tag{8}$$

where $0 < \alpha < 1$, P is a discrete stochastic variable, and p_r is the distribution function of P [14]. Higher values of α, approaching 1, giving a Renyi entropy which is increasingly determined by consideration of only the highest probability events. Lower values of α, approaching zero, giving a Renyi entropy which increasingly weights all possible events more equally, regardless of their probabilities. The special case $\alpha \to 1$ gives the Shannon entropy. The Renyi cross entropy of order α is derived as

$$I_\alpha(p, q) = \frac{1}{1-\alpha} \log_2 \sum_r \frac{p_r^\alpha}{q_r^{\alpha-1}} \tag{9}$$

where p and q are two discrete variables, p_r and q_r are their distribution functions [14]. If $\alpha = 0.5$, the Renyi cross entropy is symmetric, which means $I_\alpha(p, q) = I_\alpha(q, p)$. In the rest of the paper, when referring to the cross entropy we mean the symmetric case

$$I_{0.5}(p, q) = 2 \log_2 \sum_r \sqrt{p_r q_r} \tag{10}$$

The Renyi cross entropy is used to fuse the values of different features. As mentioned above, we use an entropy vector $V(t) = [H(Sipt), H(Dipt), H(Sthreatt), H(Dthreatt), H(Dgmlent)]$ to represent the network status of time t, thus the network status can be viewed as a time series of entropy vector $V(1), V(2), \ldots, V(t)$. Before calculating Renyi cross entropy, $V(t)$ is unitized to

$$\bar{V}(t) = [\bar{H}(Sip_t), \bar{H}(Dip_t), \bar{H}(Sthreat_t), \bar{H}(Dthreat_t), \bar{H}(Dgmlen_t)] \tag{11}$$

where

$$\begin{aligned}
\bar{H}(Sip_t) &= H(Sip_t)/H_{sum} \\
\bar{H}(Sthreat_t) &= H(Sthreat_t)/H_{sum} \\
\bar{H}(Dip_t) &= H(Dip_t)/H_{sum} \\
\bar{H}(Dthreat_t) &= H(Dthreat_t)/H_{sum} \\
\bar{H}(Dgmlen_t) &= H(Dgmlen_t)/H_{sum}
\end{aligned} \tag{12}$$

and $Hsum = H(Sip_t) + H(Dip_t) + H(Sthreat_t) + H(Dthreat_t) + H(Dgmlen_t)$.

To determine if there is any change in the network at time t compare with previous time $t-1$, we use the following equation to calculate the Renyi cross entropy of $\bar{V}(t)$ and $\bar{V}(t-1)$

$$I_{0.5}(\bar{V}(t), \bar{V}(t-1)) = 2\log_2 \sum_r \sqrt{p_r(t-1)p_r(t)} \tag{13}$$

We set η as the threshold of $\left|I_{0.5}(\bar{V}(t-1), \bar{V}(t))\right|$ to test whether there is a change. The choice of threshold η is network dependent and it can be set as experience. Since our purpose is to detect network attack, it is not enough to compare network status of time t to its previous time $t-1$, unless we make sure that no attack occurs in time $t-1$. Thus, the average of the latest n normalized Shannon Entropies is employed to replace the $t-1$, called $\bar{V}(t,n)$

$$\bar{V}(t,n) = \frac{1}{n}\sum_{i=1}^{n} \bar{V}(t-i) \tag{14}$$

Then, we calculate the Renyi cross entropy of $\bar{V}(t)$ and $\bar{V}(t,n)$, and network attack is detected if its absolute is greater than η.

$$I_{0.5}(\bar{V}(t,n), \bar{V}(t)) = 2\log_2 \sum_r \sqrt{p_r(t,n)p_r(t)} \tag{15}$$

3 Experiment Results

3.1 Data Collection

In the research, we have used Snort to monitor 32 C-class subnets in the Xi'an Jiaotong University campus network for two weeks, which include more than 4,000 users. In this paper, we select the alerts gathered in 2010-12-6. There are 862,284 alerts with 65 signatures, which come from 42,473 distinct source IP addresses and send to 11,790 distinct destination IP addresses.

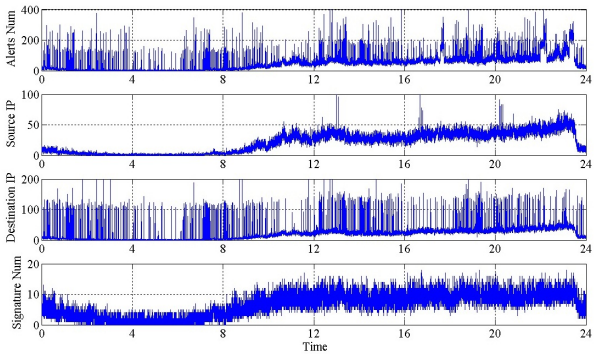


Figure 1: The statistical results of alerts (2010-12-6).

As shown in Fig.1, four statistical features of alerts display the trend as the people living customs and habits (the time interval set as 5 seconds). Few alerts are generated in the middle night; then, more alerts are detected from 8:00 to 10:00 when students get up successively; the alerts keep the same trend from 10:00 to 23:30; the alerts collapse at last 30 minutes, since network constraint due to the dormitory administrating rules.

At the same time, the statistical features change abruptly in some time intervals. In general, these abnormal upheavals are the sign of the faults or network attacks.

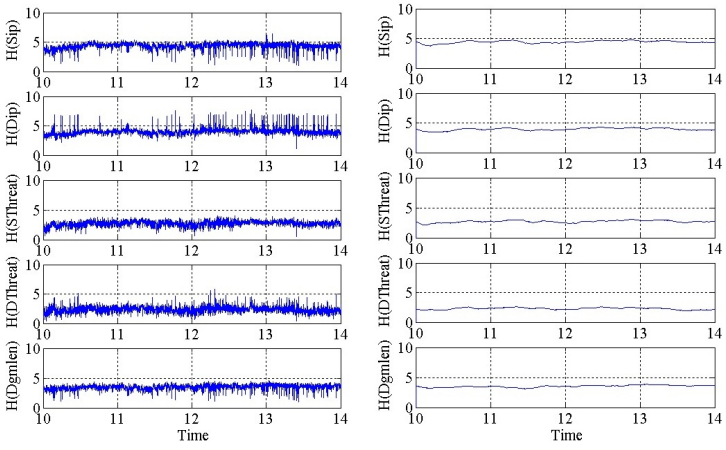
We select two alerts sets in different time period as training and test data set:

Training data set includes 170,516 alerts generated from 10:00 to 14:00. These alerts come from 13,148 IP addresses and send to 7,570 IP addresses. By analyzing these alerts manually, we identify 87 host scan attacks, 5 port scan attacks, 1 DoS attack and 1 host intrusion.

Test data set includes 578,389 alerts generated from 14:00 to 23:30. These alerts come from 29,327 IP addresses and send to 10,590 IP addresses. By analyzing these alerts manually, we identify 203 host scan attacks, 7 port scan attacks, 6 DoS attack, 3 host intrusion and 1 worm attack.

3.2 Entropy-based Attack Detection

The training data is evaluated by Shannon entropy, as shown in Fig. 2 (a). We remove the alerts associated to true attacks, which called as *Attack Alert*. The remainders are called as *Flase Alert*. We re-evaluate the Noise Alert in the training data set, as shown in Fig. 2 (b). The Shannon entropies are relatively smooth when no attack occurs; otherwise, one or some of the values would change abruptly.



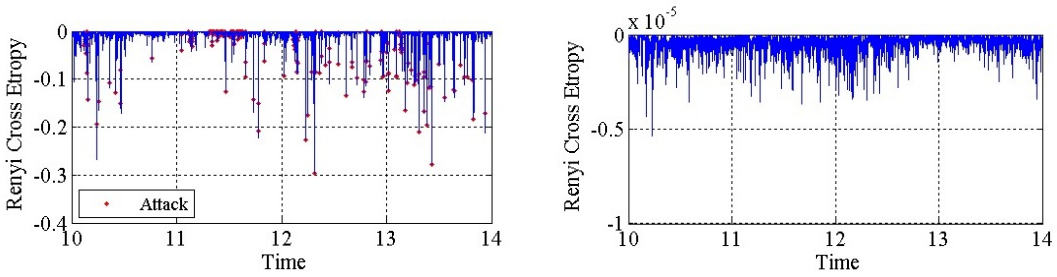
(a) All alerts (b) False alerts

Figure 2: Shannon entropy.

Although the Shannon entropies reflect the regularity of network status, it is difficult to detect attack directly by using five fixed thresholds. Because the Shannon entropy value varies with the activities of end users even the network runs in normal way. In our experiment, the Renyi cross entropy is used to fuse the Shannon entropy of five statistical features to detect attack. As shown in Fig. 3, we calculate the Renyi cross entropy of the alerts in train data set using (13). It is clearly shown that 1) the Renyi cross entropy will change sharply when the network are attacked, see Fig. 3 (a); 2) the Renyi cross entropy will be close to 0 without the large-scale network attacks and failures, see Fig. 3 (b). Thus, it is easy to detect attack using fixed threshold.

In the experiments, when $\eta_{detect} = -0.016$, 84 attacks can be detected from 94 attacks with 11 false detections. 81 host scan attacks can be detected from 87 host scans. The missed scan attacks last for a relative long time and with small scan density. 1 port scan is detected from 5 port scans. 1 host intrusion and 1 DoS attack are detected successfully.

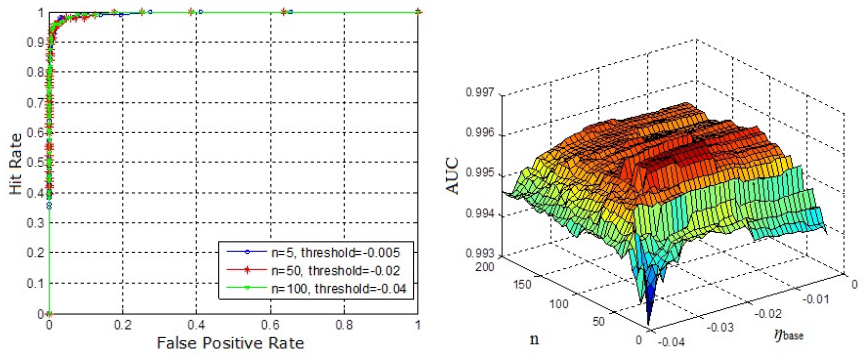
According to (14) and (15), the n and η are important for the accuracy of attack detection. In the experiments, we set $\eta_{base} = \{-0.001, -0.002, -0.003, \ldots, -0.04\}$ and $n = \{5, 10, 15, \ldots, 200\}$. For each combination of η_{base} and n, the training data is analyzed in the following method. Firstly, each $V(t)$ is unitized to $\bar{V}(t)$ using (11) and (12); Secondly, the Shannon entropy can be calculated using (14). Its unitized form is $\bar{V}(t, n)$. Finally, $\bar{V}(t)$ is compared with $\bar{V}(t, n)$ using (15) to calculate Renyi cross entropy value.



(a) All alerts (b) False alerts

Figure 3: Renyi cross entropy.

In the experiment, ROC (Receiver Operating Characteristic) is used to describe the detection results. ROC is a graphical plot of true positive rate and false positive rate [15]. Fig. 4(a) shows the ROC curve of detection results in training data, where the size of NTS n and base threshold η_{base} equals (5, 0.005), (50, 0.02) and (100, 0.04) separately. When detection threshold η_{detect} comes to 0, almost all the time intervals are detected as network attack. Thus, the detection false positive rate and hit rate are both near 100%. A detection result with high hit rate and low false rate is considered to be a good result. In this case, the ROC curve is plotted at the top left corner, and the AUC value (Area Under ROC Curve) has large value. In this paper, we use AUC value to evaluate the detection results. The best combination of n and ?base can be obtained using training data. As shown in Fig. 4(b), the AUC values of all the combinations are calculated, and the highest AUC is 0.9962 when $n = 95$ and $\eta_{base} = -0.022$.



(a) ROC (b) AUC

Figure 4: Detection result on training data set.

3.3 Testing

The test data set is analyzed to detect the attacks using entropy-based method. As shown in Fig. 5, 211 attacks can be detected from 220 attacks (detection rate is as high as 96%) with 8 false detections. 197 host scan attacks can be detected from 203 host scans. 4 port scans are detected from 7 port scans. 3 host intrusions, 1 worm attack and 6 DoS attacks are detected successfully.

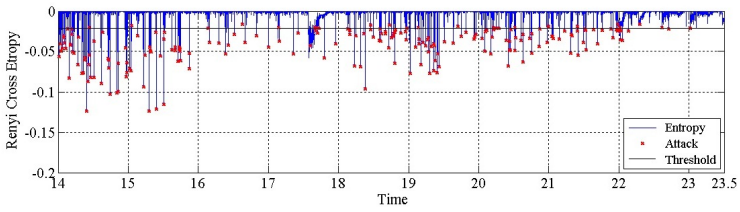


Figure 5: Attack detection results on test data set.

4 Conclusion

In this paper, a new network attack detection method based on entropy is proposed. The source IP, destination IP, alert treat and alert datagram length are selected from tens of Snort alert attributions. The Shannon entropy is used to analyze the alerts to measure the regularity of current network status. The Renyi cross entropy is employed to fuzz the Shannon entropy on different features to detect network attacks.

In the experiments, the network traffic of more than 4000 users in 32 C-class network are monitored using Snort. 748905 alerts, generated from 10:00 to 23:30 Dec. 6 2010, are selected and separated into training data set and test data set. The experiments show that the Renyi cross entropy value is near 0 when the network runs in normal, otherwise the value will change abruptly when attack occurs. The attack detection rate of entropy method is as high as 96% with only 8 false alerts.

In next step, more alerts from different time segments will be collected to test our method and an attack classification method will be considered.

Acknowledgment

This work was supported by the National Natural Science Foundation (60921003, 60970121, 91018011), National Science Fund for Distinguished Young Scholars (60825202) and the Fundamental Research Funds for the Central Universities.

Bibliography

[1] A. Gostev, "Kaspersky Security Bulletin. Malware Evolution 2010," Kaspersky, 2011.

[2] M. Fossi, G. Egan, K. Haley, E. Hohnson, T. Mack and A. Et, "Symantec Global Internet Security Threat Report Trends for 2010," Symantec, 2011.

[3] P. Cisar, S. Bosnjak and S. M. Cisar, "EWMA Algorithm in Network Practice," International Journal of Computers, Communications & Control, vol.5, pp. 160-170, 2010.

[4] G. C. Tjhai, M. Papadaki, S. M. Furnell and N. L. Clarke, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Turin, Italy, 2008, pp. 139-150.

[5] T. Pietraszek, "Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection-Recent Advances in Intrusion Detection," vol.3224, pp. 102-124, 2004.

[6] J. Mina and C. Verde, "Fault detection for large scale systems using Dynamic Principal Components Analysis with adaptation," International Journal of Computers, Communications & Control, vol.2, pp. 185-194, 2007.

[7] G. P. Spathoulas and S. K. Katsikas, in 2009 16th International Conference on Systems, Signals and Image Processing, IWSSIP 2009, Chalkida, Greece, 2009.

[8] A. Lakhina, M. Crovella and C. Diot, in Computer Communication Review, New York, United States, 2005, pp. 217-228.

[9] D. Brauckhoff, B. Tellenbach, A. Wagner, M. May and A. Lakhina, in Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC, Rio de Janeriro, Brazil, 2006, pp. 159-164.

[10] A. Wagner and B. Plattner, in Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WET ICE, Linkoeping, Sweden, 2005, pp. 172-177.

[11] R. Yan and Q. Zheng, "Using Renyi cross entropy to analyze traffic matrix and detect DDoS attacks," Information Technology Journal, vol.8, pp. 1180-1188, 2009.

[12] Y. Gu, A. McCallum and D. Towsley, "Detecting anomalies in network traffic using maximum entropy estimation," in Proc. 2005 Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement, pp. 32.

[13] C. E. Shannon, "A mathematical theory of communication," SIGMOBILE Mob. Comput. Commun. Rev., vol.5, pp. 3-55, 2001.

[14] C. E. Pfister and W. G. Sullivan, "Renyi entropy, guesswork moments, and large deviations," IEEE Transactions on Information Theory, vol.50, pp. 2794-2800, 2004.

[15] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," Pattern Recognition, vol.30, pp. 1145-1159, 1997.