Anomaly Detection for DDoS Attacks Based on Gini Coefficient

Yun Liu[1] Siyu Jiang[1] Jiuming Huang[1]

[1] The post-doctoral workstation of PLA Unit 73111, China

Abstract

Distributed Denial-of-Service (DDoS) attacks present a very serious threat to the stability of the Internet. In this paper, an anomaly detection method for DDoS attacks based on Gini coefficient is proposed. First, Gini coefficient is introduced to measure the inequalities of packet attribution (IP addresses and ports) distributions during attacks. Then, an improved TCM-KNN algorithm is applied to identify attacks by classifying the Gini coefficient samples extracted from real-time network traffic. The experimental results demonstrate that the proposed method can effectively distinguish DDoS attacks from normal traffic, and has higher detection ratio and lower false alarm ratio than similar detection methods.

Keywords: anomaly detection, Gini coefficient, TCM-KNN algorithm

1. Introduction

As one of the main security threats that the Internet is facing , Distributed Denial-of-Service(DDoS) attacks exploit lots of compromised hosts to send useless packets to overwhelm a victim in a short time, which purports to consume the victim's resource, such as bandwidth, memory etc, and make its service unavailable. Compared to other attacks, DDoS attacks are easier to launch, more harmful, harder to prevent, so the detection of DDoS attacks has become an active research topic in the field of Intrusion Detection.

According to statistics, the most of DDoS attacks are highly distributed or highly random spoofed [1,2]. The reasons come from the two aspects: (1) with the spread of Botnet, attackers are easily to launch large-scale attacks by controlling highly distributed zombies. (2)To conceal their location or exhaust rapidly the resources of victim, attackers typically randomly make the IP source address or destination port of each packet they send. So, many researchers try to detect attacks by identifying their distributed structure.

Peng [3] detected attacks by monitoring the number of new source IP addresses in a given time period based on the observation [2] that the majority of the source IP addresses of incoming packets are new to the victim during attacks but appeared before in flash crowds. Sun [4] proposed a detection method based on FCD (Flow Connection Density). In [4], a flow means a set of packets with same source IP address, destination IP address, destination port, then FCD is defined as the number of flows in a given time period. In general, legitimate users usually make use of fixed source IP addresses to access a small quantity of services, so the FCD in DDoS attacks are different from that in flash crowds. Lakhina[5] presented a detection method based on TFDE (Traffic Feature Distribution Entropy). It is discovered that most of network anomalies, including DDoS attacks, can be taken apart via the distributions of four

packet attributions, which are source and destination addresses and ports. For example, a DDOS attack will cause the distribution of traffic by source addresses to be dispersed and by destination addresses to be concentrated. Further, information entropy is used as a summarization tool to quantify the inequality of the four attribution distributions, and the results are called as Traffic Feature Distribution Entropy. Though these methods can identity DDoS attacks to a certain extent, a common drawback lies in that the correlations of attributions, which conduce to improve detection quality, are not be considered by them.

To relief this problem, we proposed a new anomaly detection method based on Gini coefficient. We adopt Gini coefficient to measure the inequalities of source IP addresses and destination ports distributions on destination IP addresses respectively, then a novel machine-learning algorithm (Transductive Confidence Machines for K-Nearest Neighbors, TCM-KNN) [6,7] is applied to identify attacks by classifying the Gini coefficient samples. The experimental results on the well-known MIT dataset demonstrate our method can effectively detect DDoS attacks with high detection ratio and low false alarm ratio.

2. Gini Coefficient of Network Traffic Distribution

When a DDoS attack occurs, a majority of source IP addresses and destination ports appeared in network traffic will concentrate on single destination IP address, which causes their distributions on destination IP address unequal. To measure these inequalities, the Gini coefficient is introduced as a summarization tool in this paper.

Gini coefficient[8] is a measure of statistical dispersion developed by the Italian economist Corrado Gini and commonly used to measure the inequality of income or wealth distribution among individuals or households. From the geometric viewpoint, if the cumulative percentages of total income received against the cumulative number of recipients, starting with the poorest individual or household, are plotted by a Lorenz curve, the Gini coefficient represents the area between the Lorenz curve and a hypothetical line of absolute equality, expressed as a percentage of the maximum area under the line.

In fact, Gini coefficient is suit not just to income but rather to any random variable, so we extend it to measure the inequalities of source IP address and destination port distributions on destination IP address in network traffic. Besides, we calculate the Gini coefficient according to a more convenient formula [9] which approximatively regards the definite integral on Lorenz curve as the sum of a series of trapezoids with equal height. Supposing that a sample set S of size N is divided into n subset $\{S_1,S_2,\cdots,S_n\}$ which are indexed in non-decreasing order in respect of their cardinalities, meaning that $|S_i| \leqslant |S_{i+1}|$, $i=1,2,\cdots,n-1$, the Gini coefficient of the division can be computed in the following formula:

$$Gini = \frac{1}{n}(2\sum_{i=1}^{n-1} w_i + 1) \qquad (1)$$

where $w_i = \sum_{j=1}^{i} p_j = \frac{1}{N}\sum_{j=1}^{i}|S_j|$ and p_j denotes the ratio of $|S_j|$ to $|S|$.

Let *sip* as the source IP address, *dip* as the destination IP address and *dport* as the destination port, we calculate the Gini coefficient of *sip* and *dport* concerning *dip* respectively and denote them as *Gini*(*sip*) and *Gini*(*dport*).

Taking *Gini*(*sip*) as an example for illustration, it is calculated in the following way. Assuming that sample the

network flows with time interval Δt and the set of distinct *dip* of the packets incoming in Δt is $\{dip_i | i=1,2,...,n\}$, defining an *n*-dimension vector *A*[*n*], in which *A*[*i*] represents the number of different *sip* of packets with the destination IP address dip_i, according to Equation(1), we have:

$$\begin{aligned} Gini(sip) &= \frac{1}{n}(2\sum_{i=1}^{n-1} w_i + 1) \\ &= \frac{2}{n}\sum_{i=1}^{n-1}(p_1 + p_2 + ... + p_i) + \frac{1}{n} \\ &= \frac{2}{nN}\sum_{i=1}^{n-1}(n-i)A[i] + \frac{1}{n} \end{aligned} \quad (2)$$

where $N = \sum_{j=1}^{n} A[j]$ is the total number of different *sip* observed by all *dip* and p_i represents the ratio of the number of different *sip* observed by dip_i to *N*.

Gini(*sip*) can theoretically range from 0 to 1. A lower *Gini*(*sip*) indicates a more equal distribution, implying lower probability of that an attack occur, while higher *Gini*(*sip*) indicate more unequal distribution meaning lower probability of there exist an attack. The metric takes on the value 0 when each *dip* sees equal number of different *sip*, i.e., *A*[1] =*A*[2] = ... = *A*[*n*] and 1 when all of *sip* focus on one *dip*, i.e., *A*[*i*]=*N*.

Gini(*dport*) can be calculated in the similar way. The combination of *Gini*(*sip*) and *Gini*(*dport*) forms a two-dimensional feature vector which is used as input of the following detection algorithm.

3. Anomaly Detection Based on Gini Coefficient

In this section, we illustrate our anomaly detection method based on Gini coefficient in detail. In our method, the TCM-KNN algorithm, a machine learning algorithm, is introduced to model the distribution of normal Gini coefficient samples, and then judge the legality of a Gini coefficient sample to test by checking how far the distances from it to normal samples is.

3.1. TCM-KNN Algorithm

TCM-KNN algorithm[6] is a machine learning method based on algorithmic randomness theory. Unlike traditional methods in machine learning, TCM-KNN can offer measures of reliability to individual points, and uses very broad assumptions except for the iid assumption (the training dataset and the testing dataset are independently and identically distributed).Besides, TCM-KNN is immune to the effect of "noisy" data in training dataset. These properties make it better detection performance than the traditional anomaly detection methods in practice.

Now, we will give the formal description of TCM-KNN algorithm for the application field of network anomaly detection. Suppose we have a train sample set $X=\{x_i \in normal,\ i=1,2,\ldots,N\}$ and a test sample set $Y=\{y_j,\ j=1,2,\ldots,M\}$, where *N* and *M* is the number of samples in *X* and *Y* respectively, and our goal is to determine every sample to test whether normal or attack according to the train set.

In the process of TCM-KNN algorithm, the sorted sequence (in ascending order) of the distances (Euclidean distance is used in this paper) of each sample x_i from the train samples are computed and denoted as D_i. Also, D_{ij} will stand for the *j*-th shortest distance in this sequence. Then, x_i is assigned a measure α_i which is called the strangeness measure [6] and defined as

$$\alpha_i = \sum_{j=1}^{k} D_{ij} \quad (3)$$

where *k* is the number of neighbors used. This measure represent the sum of the *k* shortest distances x_i from the train sample set.

On the basis of the strangeness, the *p*-value of a test sample y_j can be defined as:

$$p(\alpha_{new}) = \frac{\#\{i : \alpha_i \geq \alpha_{new}\}}{N+1} \qquad (4)$$

In equation (4), # denotes the number of elements in finite set. α_{new} is the strangeness value of y_j.

Assuming that $\{\alpha_1, \alpha_2, ..., \alpha_N\}$ are the strangeness measures for the training samples, *p*-value denote the ratio of the sum of the training samples whose strangeness values is not less than α_{new} to the sum of samples, so a lower *p*-value indicates a higher probability y_j belonging to an attack.

3.2. Anomaly Detection Algorithm Based on Gini Coefficient

Our anomaly detection algorithm based on Gini coefficient including two parts: training and testing. In training part, we sample a raw normal traffic trace with time interval Δt , and calculate the Gini coefficient of each sample, then we get a train set $\{x_i, i=1,2,\ldots,N\}$,where $x_i=[Gini(sip),Gini(dport)]_i$. Next, we calculate and store α_i for each x_i. In the original TCM-KNN algorithm, all distances of x_i from the other train samples are stored for calculatingα_i, so the space complexity is $O(N^2)$. However, it is wasting because that only the *k* shortest distances is required to do it. So we define a matrix *Dist*[*N***k*] to save the *k* shortest distances of each train sample, and once the distance of x_i from another train sample, say x_j, is calculated, update the *k* shortest distances of x_i and x_j synchronously. Thus we can obtain an improved TCM-KNN algorithm with space complexity $O(Nk)$, where $k \ll N$.

In testing part, we monitor the real-time traffic and compute the Gini coefficient with time interval Δt, thus acquire a test sample *r*. Then, we calculate the *p*-value of *r* according to the equation (4). If the *p*-value is less than the predefined confidence level δ , we can declare *r* an attack with confidence 1-δ , else we regard *r* as normal. The detail algorithm is given in Fig. 1.

Let *k* as the number of nearest neighbors; *N* as the number of train samples; δ as the prefined detection threshhold; *Dist*[*N***k*] as the matrix for save the *k* shortest distances of each train sample

***** Training Part *****

1. for *i*=1 to *N*
2. for *j*=*i*+1 to *N*{
3. compute the distance between x_i and x_j, noted as *d*;
4. find the maximum one d_i in the *k* shortest distances of x_i;
5. if (d_i>*d*) replace d_i with *d*;
6. find the maximum one d_j in the *k* shortest distances of x_j;
7. if (d_j>*d*) replace d_j with *d*; }
8. for *i*=1 to *N* { $a_i = \sum_{j=1}^{k} Dist[i][j]$;}

***** Testing Part *****

1. Sample real-time traffic and get a Gini coefficient sample *r* to test;
2. compute the α value of *r* according to equation (3);
3. compute the *p*-value of *r* according to equation (4);
4. if ($p \leqslant \delta$) declare *r* an attack with confidence 1- δ;
   else declare that *r* is normal;

Fig. 1: Anomaly detection algorithm of DDoS attacks based on Gini coefficient

4. Experiments

4.1. Dataset

We validate our claims on the DDoS attacks dataset (LLDoS 2.0.2) from MIT Lincoln Laboratory[10], which is a TCP flooding flow with randomly generated source IP address and destination port in each packet and a round time span of 7 seconds. Without loss of generality, we

add the normal background flow which is from normal dataset of the Lincoln Lab.

Besides, the Detection Ratio (DR), the False alarm Ratio (FR) and the Error Ratio (ER) are used to evaluate the detection results in this paper.

DR represents the ratio of attack samples detected correctly, FR represents the ratio of normal samples mistaken as attack ones, and ER represents the ratio of the total samples judged wrongly.

To evaluate the performance of our method, the detection results are compared with that of the two detection methods: TFDE[5] and FCD[4].

4.2. Detection Results

Sample the background flow and attack flow together and calculate the Gini coefficient of each sample to obtain both the positive and negative feature sample set. In order to test the robustness of our method against the disturbance of background flow, we increase the scale of background flow by protracting the sampling period. The sampling period of the background flow (noted as T) increase gradually from 1s to 5s by 1s each time and the sampling period of the attack flow (noted as t) is fixed to 0.01s. Thus we obtained five group data each one containing both attack samples and normal samples. Then we randomly extract 5600 normal samples and 600 attack samples from every group data where 5000 normal samples are used to train the TCM-KNN algorithm and the rest are used to test. Fig. 2 presents the distribution of two classes of samples (normal and attack) in feature space when T=5s.

In TCM-KNN algorithm, the parameters k and δ are crucial to the final detection results, so be required to identify before calculate the p-values of samples. After tested repeatedly, the better detection results of three detection methods can be obtained when the parameters are set according to Table 1. Table 2 shows the comparison of detection results of three methods.

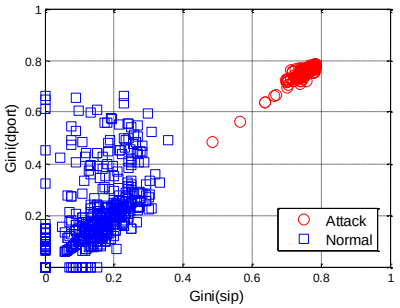

Fig.2: Distribution of normal and attack samples in feature space when T=5s

Table1: Parameters k and δ of each detection method

Detection Methods	Gini Coefficient	TFDE	FCE
***k*(number of the nearest neighbors)**	10	15	10
δ(confidence level)	0.001	0.003	0.005

We can see from Table2 that the detection results of three methods are close to each other when T is small. However, with increasement of T our method outperform the others. Specially, when T is 5s, the detection ratio of our method is still above 94% and 5.6% and 12.5% higher than TFDE as well as FCD respectivly. The reason lies in that the correlations of packet attributions are exploited in our method while the others didn‘t. To be specific, TFDE only considers the marginal probability of each attribution, leaves out the joint probability of attributions, while FCD just catched attribution flow counts rather than the shape of attribution distribution. As a result, they are easy to suffer from the interference of background flow.

Table2: Comparison of detection results of three methods

% \ T(s)		1	2	3	4	5
GINI	*DR*	99.7	98.5	97.8	95.2	94.3
	FR	0.2	0.5	1.2	1.7	3.5
	ER	0.3	1.0	1.7	3.3	4.6
TFDE	*DR*	98.3	96.7	94.5	92.3	88.7
	FR	0.0	0.7	1.8	3.2	5.2
	ER	0.8	2.0	3.7	5.4	8.3
FCD	*DR*	99.2	94.7	91.8	87.3	81.8
	FR	0.5	1.3	3.3	3.2	6.7
	ER	0.7	3.3	5.8	7.9	12.4

5. Conclusions

A Gini coefficient based DDoS anomaly detection method is proposed in this paper. This method detects DDoS attacks by measuring the inequalities of source IP address and destination port distributions on destination IP address. Experimental results show that the proposed method has satisfying detection accuracy.

In the future, we will discuss how the parameters can affect the detection results and work for an adaptive technique to set the parameters.

Acknowledgements

This work was financially supported by the National Science Foundation of China (No. 61070198 and No. 60903040)

References

[1] Houle KJ, Weaver GM, Long N, Thomas R, "Trends in denial of service attack technology", CERT and CERT Coordination Center, 2001.

[2] Jung J, Krishnamurthy B, Rabinovich M, "Flash crowds and denial of service attacks: Characterization and implications for CDNs and Web sites", Proc. the 11th World Wide Web Conference, 2002.

[3] T.Peng,C.Leckie, K. Ramamohanarao, "Proactively detecting distributed denial of service attacks using source ip address monitoring, " Proc. Of the Third International IFIP-TC6 Networking Conference, pp.771-782, 2004.

[4] Q.D. Sun, D.Y. Zhang and P. Gao, "Detecting Distributed Denial of Service Attacks Based on Time Series Analysis," Chinese Journal of Computers, 28(5), pp.767-773, 2005.

[5] A.Lakhina, M.Crovella, C.Diot, Mining Anomalies Using Traffic Feature Distributions, Proc. ACM SIGCOMM '05, pp.217-228, 2005.

[6] Y. Li, B.X. Fang, L. Guo and Y. Chen, Network Anomaly Detection Based on TCM-KNN Algorithm, Proc. ASIACCS '07, 2007, pp.13-19.

[7] Y.Li, An Effective TCM-KNN Scheme for High-Speed Network Anomaly Detection, International Journal of Advanced Science and Technology,24(2010),pp.11-16.

[8] C. Gini, Variability and Mutability, Journal of the Royal Statistical Society, 76(3), February,1913, pp. 326-327.

[9] J.H.Zhang, An Convenient Method to Calculate Gini Coefficient. Chinese Journal of Shanxi Agricultural University:Social Science Edition, 6(3),2007,pp.275-278.

[10] MIT Lincoln aboratory. http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html