Combining Supervised and Unsupervised Learning for Automatic Attack Signature Generation System

Lili Yang, Jie Wang*, and Ping Zhong

School of Information Science and Engineering, Central South University, China
{liliyang,jwang}@mail.csu.edu.cn

Abstract. Signature-based intrusion detection system is currently used widely, but it is dependent on high quality and complete attack signature database. Despite a great number of automatic attack feature extraction system has been proposed, however, with the progress of attack technology, automatic attack signature generation system research is still an open problem. This paper presents a novel combining supervised and unsupervised learning for automatic attack signature generation system based on the transport layer and the network layer statistics feature, and the system outputs the signature sets in feedback way. Finally we demonstrate the effectiveness of the model by using network data from the laboratory and Darpa2000 datasets.

Keywords: IDS, attack signature, automatic extraction, abnormal flow.

1 Introduction

In recent years, the Internet information security consciousness is gradually improved and the popularity of security software reduces the occurrence probability of the same security events, but the emerging of all kinds of malware variants or new malicious programs makes the occurrence probability of security event still high [1][2]. In such a serious information security environment, intrusion detection technology attracts more and more people's attention. Since the 1990 s, research and development of intrusion detection system presents a prosperous situation. Because of the characteristics of simple, efficient and accurate, IDS based on the signature is widely used. It depends on high quality and complete attack signature database, so the fast and accurate automatic attack signature generation system is still an important research direction in the field of network security.

The goal of automatic attack signature generation system is to find automatically new attack and extract characteristics of the new attack which can be used in IDS [3][4]. In 2003 Kreibich put forward the first automatic attack signature generation system Honeycomb [5], after that many systems have been proposed and implemented.

Since different automatic extraction technologies are used, automatic attack signature generation systems include mainly the network-based signature generation

* Corresponding author.

X.-h. Sun et al. (Eds.): ICA3PP 2014, Part I, LNCS 8630, pp. 607–618, 2014.

systems (NSG) and the host-based signature generation systems (HSG). The NSG system is deployed on the Internet, and it extracts signatures by analyzing the network data. Here, the signature points to a binary string that describes the attack with composition, distribution, or frequency. The typical NSG systems are as follows: Honeycomb, PAYL, Autograph, EarlyBird check-in, Pol, ygraph, Nemean, PADS, Hamsa, SRE, etc. The HSG system usually is deployed on a host computer, and detects the abnormality of the host and uses the collection of information on a host computer to extract the signature of the attack. The typical HSG systems are as follows: FLIPS, TaintCheck, Viglante, ARBOR, ADRMCV, COVERS, HACQIT, Packet vaccine etc [6].

Our goal is to find attack by detecting abnormal data flow in network, and extract the accurate attack signature from abnormal data. Therefore, we develop an automatic and based-network attack signature generation system model. Traditional automatic attack signature generation model adopt honeypot or classifier constructed by DPI technology or based on payload technology to identify abnormal traffic [7][8]. But the classifier cannot well identify abnormal traffic with encryption and variant. Meanwhile, honeypot need a long time to respond to worm outbreak and it may contain noise in the captured sample. In our model, we use decision tree algorithm, a supervised learning method, to construct a classifier based on network layer and transport layer statistic characteristics of network flow. Many Experiments [9][10] show that the classifier can identify abnormal data stream very well. After getting abnormal data flow, we use the unsupervised machine learning method to cluster abnormal data stream into more classes, and no similarities between each cluster. We will extract the set that can describe attack from each cluster. Here, we extract the bidirectional data flow character feature set. First, we need to extract the public substring which length is greater than 3, and use the subset of a certain frequency range to test the sample. When the sum of the false positive rate and the false negative rate is the lowest, we choose the frequency range of subset as the output from the sample. To sum up, the main contribution of our works are as follows:

1) We put forward a novel attack signature automatic generation system by combining supervised learning and unsupervised learning, which is different from tradition methods in abnormal flow identification.

2) We use feedback mechanism to confirm a frequency range of subset of public substring.

2 Challenges and Motivation

The current automatic attack signature generation system is poor in facing plenty of deformation or encryption attack. Due to this challenge, this paper seeks to develop an automatic attack signature generation model to overcome the problem.

3 Proposed Framewoek

Model of automatic attack signature generation system is designed based on supervised and unsupervised learning. Figure 1 shows the proposed system model. We use

the decision tree classifier which is constructed by supervised learning to identify abnormal data flow. And unsupervised learning is used to cluster abnormal flow. The system contains the following modules: (i)basic packet information processing module, (ii) flow generation module, (iii) flow-level statistical feature extraction module, (iv) abnormal flow identification module, (v) abnormal flow clustering module, (vi) abnormal packet payload extraction module, (vii) character feature extraction and selection module.

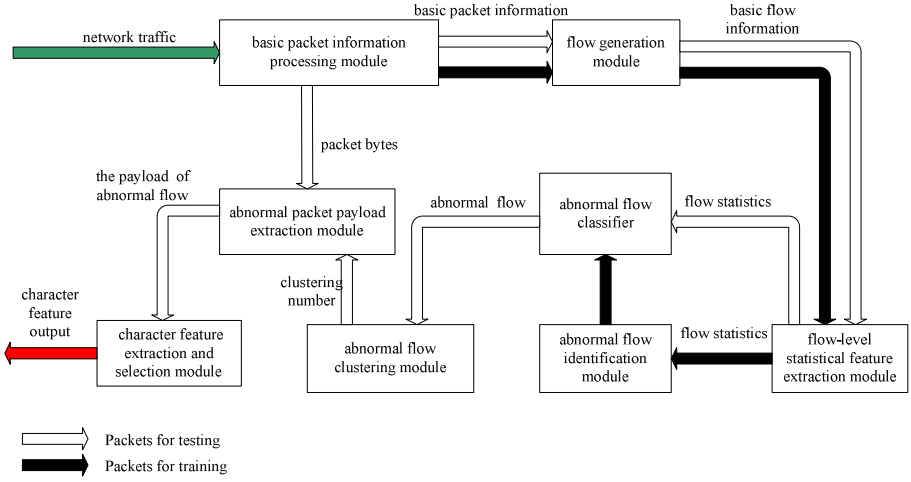

Fig. 1. Automatic attack signature generation system model

First, we need train the abnormal flow classifier. Specific process is as follows: the basic packet information processing module extracts basic information from packets for training. The flow generation module use basic packet information to generate flow, and output the basic flow information. The flow-level statistical feature extraction module obtains basic flow information to extract the statistical information and send it to abnormal flow identification module. Specially, the flow statistics come from transport layer and network layer of the TCP flow, such as the total number of packets, maximum packet size and the total number of SYN (see Table 1). Then the abnormal flow identification module trains the decision tree classifier with decision tree algorithm.

Second, we use trained abnormal flow classifier to classify packets for testing, and extract the character feature. Packets are sent to the basic information module to extract basic packet attribute and packet bytes. The flow generation module will use the basic packet information generate flow and output basic flow information. The flow-level statistical feature extraction module extracts flow statistics by using the basic flow information. At this point, the tested flow statistics information is passed to the abnormal flow classifier, rather than the abnormal flow identification module. The abnormal flow classifier will classify flow, and send the abnormal flow to abnormal flow clustering module. The abnormal flow clustering module will gather similar

degree of flow and output results. Abnormal packet payload extraction module extracts the C2S or S2C sequence in cluster. Specially, C2S means the information transmission direction is client to server, S2C means the information transmission direction of server to client. In the end, character feature extraction and selection module extract character feature from information sequence. The concrete construction of model will be explained in the later.

Table 1. Flow-level Feature Generated by the Flow of Statistical Feature Extraction Module

features Name	features Description
pkts_c2s、pkts_s2c	total number of packets
pkt_noPayload_c2s、pkt_noPayload_s2c	total number of packets without payload
bytes_c2s、bytes_s2c	total number of bytes transferred
pay_bytes_c2s、pay_bytes_s2c	total number bytes from all payloads
duration_c2s、duration_s2c	flow duration
maxsz_c2s、maxsz_s2c	maximum packet size
minsz_c2s、minsz_s2c	minimum packet size
avfsz_c2s、avfsz_s2c	average packet size
stdsz_c2s、stdsz_s2c	standard deviation of packet size
IAT_c2s、IAT_s2c	average inter-arrival time
maxpy_c2s、maxpy_s2c	maximum payload size
minpy_c2s、minpy_s2c	minimum payload size
avgpy_c2s、avgpy_s2c	average payload size
stdpy_c2s、stdpy_s2c	standard deviation of payload size
synflag_c2s、synflag_s2c	total number of SYN
rstfalg_c2s、rstfalg_s2c	total number of RST
pushflag_c2s、pushflag_s2c	total number of PSH
finflag_c2s、finflag_s2c	total number of FIN

3.1 Abnormal Flow Identification Module

This module uses the transport layer and network layer flow-level statistical features to construct classifier with the supervised learning which is decision tree algorithm. The advantage of this method is to identify the encryption and deformation flow. The decision tree is selected because the decision tree classification rules of has the characteristics of high accuracy and easily understand.

3.2 Abnormal Flow Clustering Module

In this section, we use K-Means algorithm to cluster abnormal flow. The K-Means algorithm process is as follows:

If the dataset D contains n objects in Euclid space, the division method distribute the objects in D to the k clusters C_1, …, C_k, and let $C_j \subset D$ and $C_i \cap C_j = \Phi$, where $i, j >= 1$ and $i, j <= k$. Let c_i stand for the cluster C_i, the distance between object $p \in C_i$ with center c_i is denoted by $dist(p, c_i)$, see in equation (1).

$$dist(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (1)$$

The K-Means algorithm initially chooses k objects randomly, and the selected object represents the center of cluster. For the rest of objects, according to the Euclidean distance of each cluster center, they will be assigned to the most similar cluster, the similarity is measured by $dist(p, c_i)$. Then, the algorithm iteratively changes the cluster. It uses the object which is assigned to the cluster in the last iteration to calculate a new mean for the cluster, and the new mean will be the new cluster center. When the cluster is the same as the previous, the algorithm is over. The detail of K-Means algorithm is shown in figure 2.

Algorithm: K- Means.
Input:
1) K: the number of clusters;
2) D: the dataset containing n objects.
Output: K clusters
Methods:
1) Select the initial cluster center from k objects;
2) Repeat
3) According to the mean of objects, let remainder object is assigned to the most similar cluster;
4) Recalculate the mean of objects in each cluster and update it;
5) Until no longer change.

Fig. 2. K-Means algorithm

3.3 Character Feature Extraction and Selection Module

Character feature extraction and selection module is divided into two phases. In the first phase, module extracts the direction of C2S or S2C public substring which is longer than 3 characters. In the second stage, the module detects attack sample by using the subset of public substring with a certain frequency range. When the sum of the false positives rate and the false negative rate is lowest, module outputs the range of frequency substring as the attack signature. The Detail of character feature extraction and selection algorithm is shown in figure 3.

Algorithm: Character feature extraction and selection module algorithm.

Input:

1) $Cluster_i$, the i-th cluster. Where $Cluster_i = \{C2S_{Cluster_i}, S2C_{Cluster_i}\}$, and $C2S_{Cluster_i}$ denoted the information of C2S direction, $S2C_{Cluster_i}$ denoted the information of S2C direction.

2) $PublicSequence_{len\geq 3}(x, x')$, it extracts public substring which is longer than 3 characters between x and x'.

Output: $Features_{Cluster_i}|(k_{\min}, k_{\max})$, where $(k_{\min}, k_{\max})$ is the frequency range of substring.

Methods:

1) If $Sequence_m \in C2S_{Cluster_i}$, $Sequence_n \in C2S_{Cluster_i}$ $m \neq n$ and then

2) $Features \leftarrow PublicSequence_{len\geq 3}(Sequence_m, Sequence_n)$

3) Delete the repeat sequences of the $Features$;

4)Calculate the frequency of public substring in $Features$;

5) Let the substring of the range of $(k_{\min}, k_{\max})$detects related sample. When the sum of the false negative rate and the false positive is lowest, the module outputs the substrings $Features_{Cluster_i}|(k_{\min}, k_{\max})$;

6) Repeat aforementioned process for the $S2C_{Cluster_i}$ in $Cluster_i$.

Fig. 3. Character feature extraction and selection module algorithm

4 Experimental Evalutation

In this section, we present the experimental results displaying the performance of the proposed model. First, we simply introduce the source of experimental data. Second, we evaluate the effectiveness of abnormal data flow classifier in dealing with indentifying abnormal network traffic. Third, we evaluate the effectiveness of abnormal data flow clustering module. In the end, we evaluate the effectiveness of the system model with character feature Extraction and selection experiment result.

4.1 Data

The proposed model is evaluated by using network traffic which is shown in table 2.

First, we adopted normal network traffic and Port_Scan in the lab. The phase-4-dump-inside and phase-5-dump-inside all come from the first attack scenarios of classical Darpa2000 intrusion detection dataset. This attack scenario includes multiple networks and audit sessions. specially, the sessions are divided into five stages: detect network, compromise hosts with Solaris sadmind, install mstream DDoS Trojan horse software, launch DDoS attack. The Phase-4-dump-inside dataset comes from the fourth stage, namely it is the phase of the installing the Trojan horse mstream DDoS software; Phrase-5-dump comes from the fifth stage which was launching DDoS attack stage.

Due to the large amounts of attack traffic is based on the TCP protocol, the network traffic in the experiment is TCP flow.

Table 2. source of dataset

The dataset	Dataset description
Normal	Adopting in Laboratory
Port_Scan	NMAP collecting in Laboratory
Phase-4-dump-inside[11]	Intranet dataset of phase 4 of scenario 1 in DARPA intrusion detection
Phase-5-dump-inside[11]	Intranet dataset of phase 5 of scenario 1 in DARPA intrusion detection

4.2 Evaluating the Effectiveness of Abnormal Flow Classifier

In this section, the dataset which mixes normal with phase-4-dump-inside is sent to the abnormal flow identification module. After that we get an abnormal classifier, and we use the way of crossing validation to evaluate the effectiveness of abnormal flow classifier. See figure 3, the figure shows abnormal data flow classifier reached more than 98% of the correctly classified instance. Therefore, the way of using decision tree classifier to construct abnormal flow identification module is feasible and effective in identifying abnormal flow.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         526               98.6867 %
Incorrectly Classified Instances         7                1.3133 %
Kappa statistic                          0.7933
Mean absolute error                      0.0191
Root mean squared error                  0.1126
Relative absolute error                 27.0512 %
Root relative squared error             60.714  %
Total Number of Instances              533

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.996     0.263      0.99      0.996     0.993      0.766    0
                 0.737     0.004      0.875     0.737     0.8        0.766    4
Weighted Avg.    0.987     0.254      0.986     0.987     0.986      0.766

=== Confusion Matrix ===

   a   b   <-- classified as
 512   2 |   a = 0
   5  14 |   b = 4
```

Fig. 4. the result of verifying abnormal classifier with cross validation

4.3 Evaluating the Effectiveness of Abnormal Flow Clustering Module

We input the dataset which mixes phase-4-dump_inside with Phase-5-dump_inside into the abnormal flow clustering module in this section, and the result is shown in table 3.

Table 3. the result of clustering

category	Phase-4-dump-inside	Phase-5-dump-inside
Before clustering	19	32
After clustering	15	36
Error clustering	5	1

According to the anticipated target, the module can separate the two phase flow automatically. But the experimental result shows that the clustering effect is not very well. In table 3, the error clustering refers that after clustering the instance of phase-4-dump-inside is classified as the instance of phase-5-dump-inside, and the instance of phase-5-dump-inside is classified as the instance of phase-4-dump-inside.

We assume Right probability after clustering = right number after clustering / category number before clustering, the instance of phase-4-dump-inside dataset correctly clustering probability is 52.6%, but the instance of phase-5-dump-inside correctly clustering probability is 96%. In spite of K - Means algorithm is not very well in classifying two types of abnormal dataset. But it is good for extracting attack signature because the instance in each cluster has the similarity, and the similarity can help system extracts more accurate signature.

4.4 Character Feature Extraction and Selection Experiment Result

In this section, we respectively use phase-4-dump_inside and phase-5-dump_inside dataset to experiment. First, we do experiment by using phase-4-dump_insde dataset.

(1) The Experiment by Using Phase-4-Dump-Inside Dataset.
There are 19 flows after generating TCP flow in flow generation module. Table 4 shows the result of clustering by using phase-4-dump-inside dataset.

Table 4. the result of clustering by using phase-4-dump-inside dataset

No.	Label	Count
1	Cluster0	4
2	Cluster1	5
3	Cluster2	10

Due to Cluster0 without transport information, we experiment by using the C2S direction of cluster1 and cluster2. The result show in figure 5 and figure 6. We choose the subset of substring which is longer than 3 characters to detect the related samples.

See figure 5 and figure 6, the set of substring which is represented by the shortest cylindrical surface is as the attack signature set. Table 5 shows the attack signature of phase-4-dump-inside dataset. It displays the attacker was installing mstream Trojan software on the host by using remote desktop. Since the phase-4-dump-inside is the phase which was installing mstream software, the extracted set can describe it.

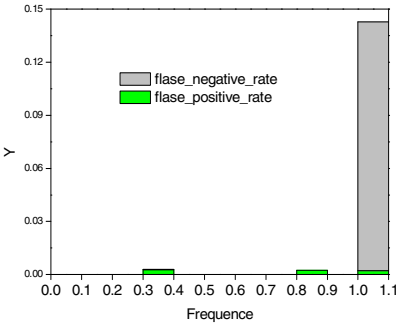

Fig. 5. The result of feature extraction experiment by using cluster1 from Phase-4-dump_inside dataset

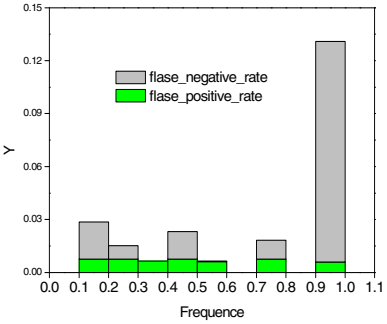

Fig. 6. The result of feature extraction experiment by using cluster2 from Phase-4-dump_inside dataset

Table 5. the attack signature of phase-4-dump-inside dataset

sample	frequency range	attack signature
cluster1	[0.3,0.4)	✓ Uroot,root,rcp -f /.sim/home/jhaines/ATTACKS/mstream/solaris/ ✓ er-sol …
cluster2	[0.5,0.6)	✓ rcp ✓ /.sim/home/jhaines/ATTACKS/mstream/solaris/ …

(2) The Experiment by Using Phase-5-Dump-Inside Dataset.

In this section, we generate TCP flow by using packets from phase-5-dump-inside dataset, then put it into the abnormal flow identification module and abnormal flow clustering module. The table 6 shows the clustering result.

Table 6. the result of clustering by using phase-5-dump-inside dataset

No.	Label	Count
1	Cluster0	4
2	Cluster1	15
3	Cluster2	13

As shown in the figure 7, figure 8 and figure 9, these figures respectively display the result of feature extraction experiment. We detect samples with a frequency range of substring which is longer than 3 characters, and when the sum of the false positive rate and the false negative rate is lowest, we will choose substrings represented by the shortest cylindrical surface as the attack signature, just like the experiment of phase-4-dump_inside dataset.

The table 7 lists some attack signature. From this table we can find the attacker is visiting a web site. Because phase-5-dump-inside dataset belongs the phase of launching DDOS, it would bring some internet traffic when attacker access to the internet. We find the attack signatures can describe the attack, but we cannot find the order of launching attack because of the orders in UDP packet. However, it cannot hinder us to extract signatures to describe TCP traffic.

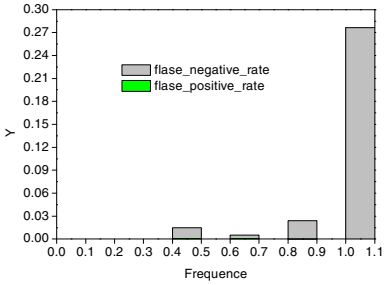

Fig. 7. The result of feature extraction experiment by using cluster0 from Phase-5-dump-inside dataset

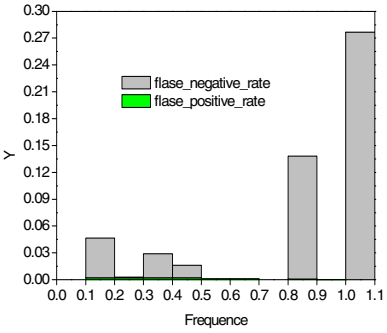

Fig. 8. The result of feature extraction experiment by using cluster1 from Phase-5-dump-inside dataset

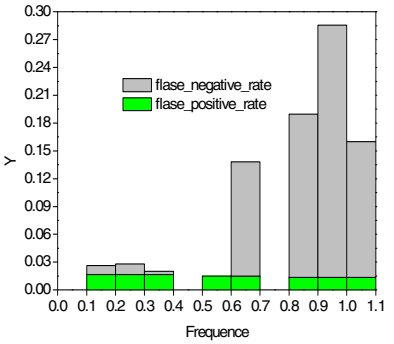

Fig. 9. The result of feature extraction experiment by using cluster2 from Phase-5-dump-inside dataset

Table 7. the attack signature of phase-4-dump-inside dataset

sample	the frequency range		attack signature
cluster0	[0.6,0.7)		✓ l/User-Agent: Mozilla/3.01 (Win95; I;)Host: www.af.milAccept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */* …
cluster1	[0.5,0.6)	[0.6,0.7)	✓ HTTP/1.0Referer: http://www.af.mil/User-Agent: Mozilla/2.0 (compatible; MSIE/3.01; Windows 95)Host: www.af.milAccept: image/gif, image/x-xbitmap,image/jpeg, image/pjpeg, */*Accept-Language: enUA-pixels: 1024x768UA-color: color32UA-OS: Windows 95UA-CPU: i686 …
The fourth stage cluster2	[0.3,0.4)		✓ jpg HTTP/1.0Referer: http://www.af.mil/User-Agent: Mozilla/2.0 (compatible; MSIE/3.01; Windows 95)Host: www.af.milAccept: image/gif, image/x-xbitmap,image/jpeg, image/pjpeg, */*Accept-Language: enUA-pixels: 1024x768UA-color: color32 UA-OS: Windows 95UA-CPU: i686 …

5 Conclusion

This paper presents a combining supervised and unsupervised learning for automatic attack signature generation system model, which based on feature from the transport and network layer. These features are more resilient to payload encryption. Our model

deals with the packet from internet and generates attack signature. Experiment results show that our work can extract the effective signature. For future work, we plan to improve the effectiveness of abnormal flow clustering module. We will extend the formulation to an online learning setting.

Acknowledgments. This work is supported by National Natural Science Foundation of China under Grant No.61202495.

References

1. China Internet Network Information Center. China Internet Development Statistics Report, http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201403/P020140305346585959798.pdf
2. Wang, X.L.: Analysis and Detection of Botnet Anomaly Traffic. Beijing University of Posts and Telecommunications. Ph D Thesis, Beijing (2011)
3. Niu, S.Z.: Introduction to Secure Information Systems, pp. 3-15. Beijing University of posts and telecommunications Press, Beijing (2004)
4. Tang, Y., Lu, X.C., Wang, Y.J.: Survey of Automatic Attack Signature Generation. Journal on Communications 30, 96–105 (2009)
5. Kreibich, C., Crowcroft, J.: Honeycomb-creating intrusion detection signatures using honeypots. In: Proceedings of the Second Workshop on Hot Topics in Networks, Boston, pp. 51–56 (2003)
6. Tang, Y.: Research on Network-based Automatic Attack Signature Generation. National University of Defence Technology. Ph D Thesis, Changsha (2008)
7. Wang, K., Cretu, G.F., Stolfo, S.J.: Anomalous payload-based worm detection and signature generation. In: Valdes, A., Zamboni, D. (eds.) RAID 2005. LNCS, vol. 3858, pp. 227–246. Springer, Heidelberg (2006)
8. Vargiya, R., Chan, P.K.: Boundary detection in tokenizing network application payload for anomaly detection. In: Proceedings of ICDM Workshop on Data Mining for Computer Security (2003)
9. Comar, P.M., Liu, L.: Combining Supervised and Unsupervised Learning for Zero-Day Malware Detection. In: Proceedings IEEE INFOCOM, pp. 2022–2030. IEEE Press (2013)
10. Han, J.W., Kamber, M.: Data Mining Concepts and Techniques, pp. 211–321. China Machine Press, Beijing (2011)
11. Lincoln Laboratory, DARPA Intrusion Detection Scenario Specific Data Sets (2000), http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/2000data.html