Ann. Data. Sci. (2014) 1(3–4):311–323
DOI 10.1007/s40745-014-0023-7

Data Mining-based DNS Log Analysis

Hongyuan Cui · Jiajun Yang · Ying Liu · Zheng Zheng · Kaichao Wu

Received: 15 October 2014 / Revised: 20 November 2014 / Accepted: 20 December 2014 /
Published online: 17 January 2015
© Springer-Verlag Berlin Heidelberg 2015

Abstract Domain name system (DNS) provides a critical function in directing Internet traffic. Defending DNS servers from bandwidth attacks is a significant task of DNS service providers. Traditional rule-based anomaly or intrusion detection methods are not able to update the rules dynamically. Data mining based approaches are able to find various patterns in the massive dynamic query traffic data. The patterns may assist the DNS service providers to detect anomalies in real time. In this paper, a novel frequent episode mining algorithm is proposed, as well as a volume trend prediction method which allows anomalies to be detected in real time. Density-based clustering approach is adopted to partition numerous domain names into different groups based on the characteristics of their query volume time series. Consistent episode mining method is proposed to find how the query traffic 'propagate' at different time between different domain names. Experiments are performed on a real-word DNS log data

H. Cui · J. Yang · Y. Liu (✉)
School of Computer and Control, University of Chinese Academy of Sciences, Beijing, China
e-mail: yingliu@ucas.ac.cn

H. Cui
e-mail: hongyuancui@163.com

J. Yang
e-mail: 398966925@qq.com

Y. Liu
Fictitious Economy and Data Science Research Center,
Chinese Academy of Sciences, Beijing, China

Z. Zheng · K. Wu
Computer Network Information Center, Chinese Academy of Sciences, Beijing, China
e-mail: zhengzheng@cnic.cn

K. Wu
e-mail: kaichao@cnic.cn

set. Interesting patterns are presented, indicating data mining based approaches are suitable and promising in the domain of DNS service.

Keywords Data mining · Clustering · Frequent pattern mining · DNS · Anomaly detection

1 Introduction

Domain name system (DNS) is a hierarchical distributed naming system for computers, services, or any resource connected to the Internet. A DNS resolves queries for URLs into IP addresses for the purpose of locating computer services and devices worldwide. By providing a worldwide, distributed keyword-based redirection service, DNS is an essential component of the functionality of the Internet.

With the ever increasing network flow and complexity of network topology, problems often happen in DNS service. For example, a large-scale network break-down happened in 6 provinces in China in 2009 due to an attack to the DNS server by a hacker. China has been one of the countries suffering from enormous network attacks in the world. Security has become a crucial problem in DNS service. Thus, it is an important task for DNS service providers to detect and report anomalies or exceptions as early as possible, and reduce the loss resulted from the unexpected events. Another important task is to provide high quality service to Web users.

Traditional methods in DNS security are rule-based methods. DNS experts have to identify the characteristics or features of any abnormal behavior from historical data offline, and then explicitly provide them to the monitoring system in the form of rules. However, such rule-based methods have two serious weaknesses: (1) the rules are not easy to update since efforts of domain experts are required. However, the patterns of abnormal behaviors in the network are evolving dramatically, and thereby the effectiveness of the detection system will be significantly reduced; (2) the size of historical data set collected by DNS system is so huge that beyond the ability of human being to analyze. For example, the number of query records captured by DNS log at a top level domain server is over 40 billion in a single month. Automatic quantitative analysis techniques on massive DNS data are in real demand.

Data mining is a kind of technique that can discover interesting, meaningful and understandable patterns hidden in massive datasets. The patterns discovered by data mining can be utilized in decision-making in many domains. To our best knowledge, data mining has not been widely used in DNS query traffic analysis yet. Therefore, in this paper, we explore to solve problems in DNS service by applying various data mining methods. Our contributions are listed as follows:

(1) In order to predict the traffic volume at a domain name and prevent attacks by hackers, we propose a volume prediction method. It discovers the frequently occurred query volume trend patterns from the most recent DNS log. If the current query volume at a domain name does not match with the predicted trend, an anomaly alarm will be delivered to the system instantly.
(2) In order to have a deep understanding of the features of the query traffic of different domain names, we partition the query traffic time series from all the domain

names into distinct clusters by adopting a density-based clustering algorithm. The representative query traffic series of each cluster is referred as the query traffic pattern. Such results provide us a chance to further investigate the browsing patterns of the Web users or identify the common features of various anomalous queries.

(3) A consistent pattern based traffic volume monitoring and anomaly prediction method is proposed. If a frequent episode *fe* happens on a large portion of days at a given DNS server at a certain time, it is called a consistent pattern. All the DNS servers that have a common *fe* are clustered into a same group. Once an abnormal query volume is observed at a DNS server, a warning message will be sent out to the other members in the cluster. This method provides us a chance to predict the abnormal volume before it really takes place.

(4) The effectiveness of our proposed methods is examined by a real-world DNS log dataset and the experimental results are presented.

The rest of this paper is organized as follows. Section 2 overviews some related work. In sect. 3, we present our query volume prediction method. In sect. 4, we briefly introduce DBSCAN clustering algorithm and present the clustering results. Section 5 presents our consistent pattern based volume monitoring and anomaly prediction method. Section 6 summaries our current work.

2 Related Work

Since query traffic flow is an accurate reflection of DNS service, anomaly detection in query traffic has been paid more and more attention. For example, Jung et al. [1] proposed a novel method to detect anomaly in SMTP Client by DNS query traffic. Ishibashi et al. [2] proposed a method to discover junk mail senders by studying ISP DNS. But in some circumstance, DNS itself can be part of the attack in Internet like DDoS [3] and DNS cache poisoning [4].

Ji et al. [5] proposed a K-means clustering based algorithm to cluster the temporal behaviors of IP addresses and domain names. It partitions the domain names into four clusters. Rather than comparing the traffic volume happened at different domain names, the method in [5] performs clustering on a set of derived variables, such as the total number of DNS requests, the number of distinct IPs, the average time interval between two DNS requests, etc. Although it produces interesting results, it requires prior knowledge of the number of clusters, k, which is not easily obtained beforehand.

Wang et al. [6] proposed a mathematical method to detect nation-wide large-scale attacks on the Internet. A covariance matrix is built to record the covariance between the query volume happened at two different provinces at different time stamps. Average covariance matrix indicates a normal situation. If the current covariance matrix deviates from the average covariance significantly, an abnormal event may be going on. This method is suitable for nation-wide attacks but fails in the detection of attacks towards a specific domain name.

Xu et al. [7] improved RIPPER algorithm to detect Botnet, which is often used for malicious activities (e.g., DDos, spam, phishing etc.). It outperforms the traditional

algorithms, such as features matching or statistical methods, in discovering more less-visited domain names.

3 Query Volume Prediction

Volume prediction and anomaly detection in DNS query traffic is significant for DNS service providers. Abnormal query volume, as well as abnormal behaviors or illegal behaviors of Web users may result in malfunction of DNS server or network. As far as our knowledge goes, most existing algorithms [1,6,7] are rule-based or black-list based approaches, where the rules are not able to be adjusted dynamically.

In this section, we propose a novel algorithm, frequent episode mining which discovers all the consecutive frequent patterns in a sequence database. Frequent episode mining is different with frequent itemset mining. An episode is a sequence of consecutive events while an itemset is a set of items. We assume that if an episode occurs frequently in the recent history, it may occur repeatedly with high probability unless an abnormal event happens. Since patterns with time intervals tend to be haphazard and unreliable, we are only interested in patterns with no interval.

We can predict the query volume at a given domain name at the next moment by referring its recent frequently occurred query volume patterns. The frequent episodes of query can help DNS experts answer questions such as how the episodes evolve over time by sorting the interesting patterns by the timing stamps, which domain names have the same traffic volume trend in a period time, etc.

3.1 Problem Statement

- $I = \{i_1, i_2, \ldots, i_m\}$ is a set of items;
- $T = \{t_1, t_2, \ldots, t_p\}$ is a set of timing stamps;
- An event, $E = \{t_k i_j\}$, where item i_j $(1 \leqslant j \leqslant m)$ happened at t_k $(1 \leqslant k \leqslant p)$;
- $D = \{S_1, S_2, \ldots, S_n\}$ is a sequence database where each sequence $S_i \in D$ consists of a sequence of events;
- An event set ES is a frequent episode if $sup(ES) \geqslant min\_sup$, and the timing stamps in ES are consecutive, where min_sup is the user pre-specified minimum support threshold;
- The task of frequent episode mining is to find the complete set of episodes, $U = \{ES | sup(ES) \geqslant min\_sup\}$.

Let's use the query volume database in Table 1 as an example. Let $I = \{u, d, s\}$, where u denotes '*rise*' in volume, d denotes '*decline*' and s denotes '*steady*'. $T = \{00, 01, 02 \ldots, 23\}$, denotes 24 hours of a day. Assume the minimum support threshold is 2, then $(00s, 01u, 02d)$ is a frequent episode because it occurred in two sequences, S_4, S_7, consecutively occurred from 0 o'clock to 2 o'clock. Although $(00s, 03u, 04d)$ occurred in S_1 and S_7 respectively, it is not a frequent episode because it violates the definition that the timing stamps must be consecutive.

Table 1 Query volume sequence database of bbs.kepu.net.cn

ID	Query volume sequence (u: up, d: down, s: steady) (24 h)						
S_1	*00s*	*01s*	*02u*	*03u*	*04d*	…	*23u*
S_2	*00s*	*01u*	*02u*	*03d*	*04s*	…	*23s*
S_3	*00s*	*01d*	*02u*	*03s*	*04s*	…	*23d*
S_4	*00s*	*01u*	*02d*	*03s*	*04u*	…	*23u*
…							
S_7	*00s*	*01u*	*02d*	*03u*	*04d*	…	*23d*

3.2 Frequent Episode Mining

Figure 1 shows the pseudo code of our proposed frequent episode mining algorithm. U denotes the collection of all the frequent episodes. It follows the *generation-and-test* methodology in Apriori [8]. In the main loop (from line 2 to 9), it generates the k-episode candidates and then finds out the real frequent k-episodes by scanning the database. With the support of *Downward Closure Property*, procedure *candidate_gen* () generates a k-episode candidate by joining two frequent (k-1)-episodes which share the common k-2 prefix. Note that the timing stamps of all the events in each episode must be consecutively increasing.

3.3 Volume Prediction

We predict the query volume of a given domain name at incoming moment by referring its recent patterns. The prediction method can be roughly outlined in three steps:

(1) Mine the frequent episodes from the recent query volume sequence database of a given domain name;
(2) Obtain strong patterns from the frequent episodes. Here we adopt the concept, confidence, from association rules mining model [8]. Assuming $ES = A \cup B$ is a frequent episode, the confidence of ES is defined as the conditional probability of B given A, that is, confidence(ES) = confidence$(A \cup B) = p(B|A)$. If a frequent episode ES satisfies the minimum confidence threshold, *min_conf*, we call ES a strong pattern.

Predict the incoming query traffic volume by referring the strong patterns. If the actual traffic volume deviates from the prediction significantly, an anomaly warning will be delivered to the central monitoring system immediately.

3.4 Experimental Results

We use a real-world DNS log dataset in our experiments. A DNS log record is generated whenever a DNS request is issued by a client. For example, when a user browses a domain name by its URL, a DNS log record will be created and recorded in the log. Attributes created by DNS log include requesting time, source IP, destination URL,

```
Input: Sequence database D, minimum support threshold, min_sup.
Output: U, frequent episodes in D.

1     U1 = find_frequent 1-episodes (D);
2     for (k = 2; Uk-1 ≠ ∅; k++) {
3         Ck = candidate_gen (Uk-1, min_sup);
4         for each sequence S ∈ D
5             for each c ∈ Ck
6                 if  c ⊆ S then
7                     c.count++;
8         Uk = {c ∈ Ck | c.count≥ min_sup}
9     }
10    return U = ∪k Uk;

procedure candidate_gen (Uk-1: frequent (k-1)-episodes; min_sup: minimum support
threshold)
1     for each episode p1 ∈ Uk-1
2         for each episode p2 ∈ Uk-1
3             if (p1[1] = p2[1]) ∧ (p1[2] = p2[2]) ∧ … ∧ (p1[k-2] = p2[k-2])) then
{
4                 c = p1 ⊳⊲ p2 in increasing order by time;
5                 if timing points of c[k-1] and c[k] are not consecutive then
6                     delete c;
7                 else if has_infrequent_subsets (c, Uk-1) then
8                         delete c;
9                     else add c to Ck;
10    }
11     return Ck;

procedure has_infrequent_subsets (c: candidate k-episodes; Uk-1: frequent (k-1)-
episodes)
1   for each (k-1)-episode of c
2      if s ∉ Uk-1 then
3          return TURE;
4   return FALSE;
```

Fig. 1 Pseudo code of frequent episode mining algorithm

query type, etc. The size of a DNS log is huge. For example, billions of records are captured at a top level domain server every month. It is even larger at a root domain server. The DNS log dataset we used here is captured by a local DNS server (Computer Network Information Center, Chinese Academy of Science) between 08/28/2012 and 09/03/2012.

Since an alert of an anomalous volume fluctuation is more valuable than the raw number of queries at a given domain name, we pre-process the log data as follows:

(1) Find all the distinct domain names occurred in the DNS log;
(2) Count the number of queries at each domain name by hour by day;
(3) Create a separate dataset for each domain name, each row containing a sequence ID and the number of queries happened hourly;

Table 2 Frequent episodes and strong patterns discovered at bbs.kepu.net.cn

Pattern length	Episodes	Support	Confidence (%)
3	*18u19s20s*	2/7	66.7
3	*03s04s05d*	2/7	66.7
4	*14s15s16u17d*	2/7	100
…	…	…	…

(4) Transform the number of queries into variation. '*d*' denotes decline in query traffic volume from the last hour, '*u*' denotes rise, and '*s*' denotes steady.

We select the relatively popular domain names whose average query frequency per day is over 1,000. 147 data sets are created, one set per domain name. Table 1 shows part of the dataset for bbs.kepu.net.cn, where each row represents the query traffic variations by hour in a day.

Experiments are performed on some of the domain names and numerous strong episode patterns are mined. In order to avoid too many useless episodes or redundant frequent episodes, we only keep the strong episodes equal or longer than three events. We present the results from bbs.kepu.net.cn as an example. Eight strong episode patterns are mined from the query volume sequence database for bbs.kepu.net.cn when *min_sup* and *min_conf* are set at 25 and 65 %, respectively. Table 2 presents some of the strong episode patterns. From Table 2, we can make predictions with high confidence in some cases. For example, if the query volume rose from 14:00 to 16:00, the probability the volume will decline from 17:00 to 18:00 is 100 %. So, if the query traffic volume rises at some time between 17:00 and 18:00, we will have high confidence to believe that an anomalous event is happening at bbs.kepu.net.cn.

4 Query Traffic Characteristic Analysis

The number of domain names on the Internet is so enormous that beyond the ability of human being to analyze their query traffic characteristics one by one. Thus, it is necessary to partition the domain names into groups where domain names with similar query traffic are in the same group. Experts can thereby perform analysis in each group, such as extracting the common characteristics of the Web users, or identifying the common characteristics of anomalous behaviors, which is useful for DNS service providers to prevent attacks.

In this section, we cluster the domain names according to their query traffic volume trends along a period of time. We assume that if some domain names have a similar trend, there is a high possibility that they have some interior relation.

4.1 DBSCAN

Clustering is one of the most widely used data mining techniques. It partitions the data into clusters so that objects within a cluster have high similarity in compari-

son to one another but are dissimilar to objects in other clusters. Dissimilarities are assessed based on the attribute/dimension values describing the objects. Distance between each pair of objects is often used as the similarity measure. K-means is the most popular clustering algorithm with no doubt. Although K-means clustering has been studied in DNS query traffic in [5] and interesting results were produced, it requires prior knowledge of the number of clusters, k, which is not easily obtained beforehand.

In this paper, we are interested in finding out the characteristics of query volume time series at different domain names, whose dimensionality is high in nature. Therefore, we adopt a more recent clustering algorithm, density-based spatial clustering of applications with noise (DBSCAN) [9]. It defines a cluster as a maximal set of density-connected objects. It grows regions with sufficiently high density into clusters. It shows strong capability to discover arbitrary shaped clusters, and leaves the user with the responsibility to select the values of two crucial parameters, ε-neighborhood and *MinPts*. With different input parameters, we can get clustering results with different resolutions.

4.2 Data Preprocessing

We use the DNS log captured by a local DNS server (Computer Network Information Center, Chinese Academy of Science) between 08/28/2012 and 09/02/2012. We used 147 most frequently queried domain names. A vector consisting of 144 numbers is created for each domain name, where each number denotes the query traffic volume per hour in the above six days. The volume is normalized between 0 and 1.

4.3 Experimental Results

We implemented DBSCAN exactly the same as in [9]. Three clusters were discovered and the corresponding clustering representatives are presented in Fig. 2, where each curve reflects the hourly query volume. 32 out of 147 domain names are discarded as noises. In Fig. 2, we can see that the difference between the query trends of different clusters is evident, indicating that the behaviors of the users visiting the domain names in different clusters have quite different characteristics. In each figure, query volume time series of 4 representative servers were selected to plot with different colors. Although our current clustering result is preliminary, it is promising to segment abnormal temporal behaviors from others by DBSCAN clustering, and thereby extract the features of anomalous DNS queries.

5 Query Traffic Volume Monitoring and Anomaly Prediction

In this section, we propose a consistent episode mining method. A consistent episode is defined as a frequent episode which occurs frequently at a given DNS server at a given time. The consistent episodes can help DNS administrators to answer questions such as how the episodes evolve over time by sorting the consistent episodes by

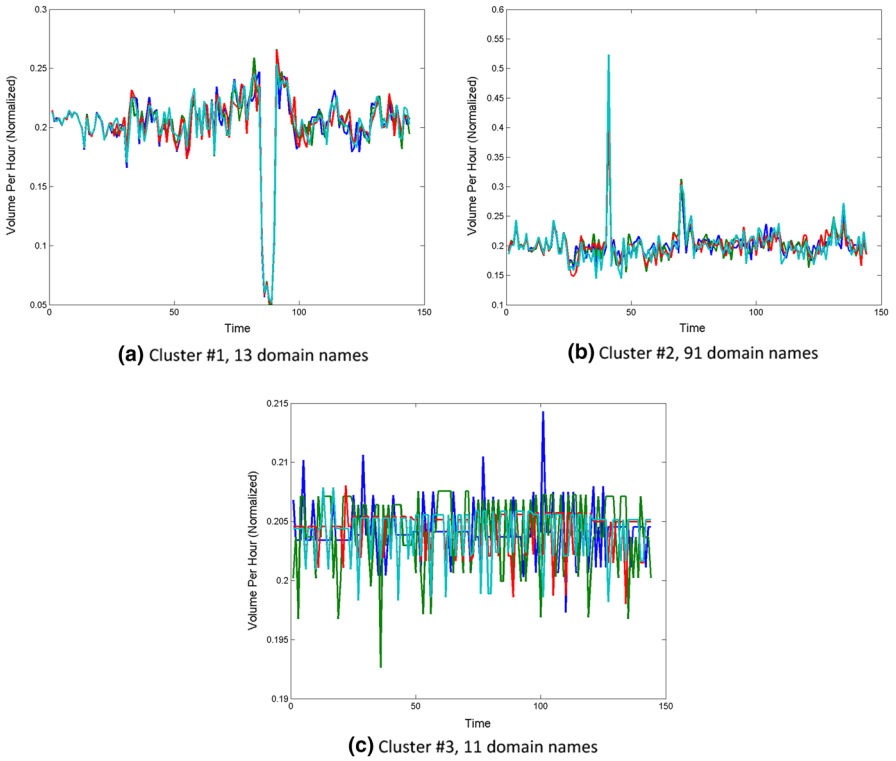

Fig. 2 Cluster representatives of DNS query volume

the timing stamps, or how a network traffic congestion propagates spatially (what is the origin of the congestion and by what routes the congestion propagates) by combining the knowledge of the spatial relations (distance, direction, inter-connection, etc.) between the corresponding DNS servers, etc. Spatial effects should be taken into account because we do observe unusual effects of volume in a server to another server at a 'seemingly unrelated location'.

5.1 Consistent Episode Mining

5.1.1 Consistent Episode

Assume we have already discovered all the frequent episodes in a query volume sequence database as in Sect. 3.1. And assume we have N days' data and a user-specified minimum threshold M. We proceed to find the *consistent episodes*. If a frequent episode *fe* happens on M out of the N days at a server A at a certain time t, we call (A, t, *fe*) a *consistent episode*. All the severs that have a common *fe* are clustered into a same group. Finally, we send the consistent episodes to the corresponding servers over the network. The consistent episodes and the corresponding servers are important because the patterns tend to re-occur.

5.2 Anomaly Prediction

Since the servers in the same group show similar flow trends for known or unknown reason, a server's incoming traffic could be predicted based on the most recent observations from its group members.

You may have a question that why not to let each DNS server predict its incoming query volume by using its local statistical values, such as the mean volume of the query volume observed in the past few days at a given time, or the mean congestion level in the past few days at a given moment, or the observations from the last day. The answer is as follows: by looking at local statistics, a DNS server only sees its local view with no chance to know the global view of the neighboring servers, which actually may impact its query flow. A group of servers that share a common consistent episode may have similar behaviors. Therefore, it may be more confident to predict the incoming volume at a certain server based on an event happened a moment ago. For example, consider that servers A, B, and C always experience the same congestion level sequence (*light, medium, medium, medium*) from *6*, *8*, and *10am* for subsequent 4 hours, respectively. A, B, and C may or may not be spatially connected to each other. If today, somehow, the congestion level sequence observed at A at *6am* is (*medium, heavy, heavy, heavy*), it is highly likely that this abnormal query volume will happen at B and C soon. So A will send a warning to server B and C so that they can send out "abnormal congestion" warning signals to its local monitoring devices or network administrators in advance.

5.3 Data Preprocessing

A real world DNS log dataset from a local DNS server (Computer Network Information Center, Chinese Academy of Science) is used for our experiment. The raw file contains records took place between 08/28/2012 and 09/03/2012. The number of queries to a DNS server is recorded in the log file. However, since the congestion level at each DNS server is more important than the raw visiting number, we transformed the number of visits to a server to congestion levels by Eq. 1.

$$L = \begin{cases} 1 & n < 16 \\ log_2 n - 2 & 16 \leq n \leq 128 \\ 5 & n > 128 \end{cases} \tag{1}$$

where n denotes the number of visits per hour and L denotes the corresponding congestion level. There are five different congestion levels: *non, light, medium, heavy and very heavy*, represented by *1, 2, 3, 4, 5*, respectively.

Table 3 shows an example query volume congestion level sequence at DNS server with IP address 114.112.69.61.

5.4 Experimental Results

We performed consistent episode mining on the DNS log from 08/28/2012 to 09/02/2012 and tested the capability of prediction by consistent episodes on the log

Table 3 Query volume congestion level sequence at DNS server 114.112.69.61

IP address	Total number of visits	Congestion level per hour per day
114.112.69.61	2	111111111111111111111111

Table 4 Precision when varying M

M	Precision
1	0.971178
2	0.990909
3	0.992083
4	0.996045
5	0.995374
6	0.996088

of the last day, 09/03/2012. We varied the minimum threshold, M, from 1 to 6 and the precisions of the prediction by consistent episodes is shown in Table 4. Evidently, the prediction precision is quite high by using the consistent episodes mined from the history of each server. In addition, the larger M, the more consistent the pattern, and the higher the prediction precision.

6 Conclusion

Data mining is a kind of techniques that can discover useful and valuable patterns from huge datasets. Therefore, in this paper, we explored various existing data mining methods and proposed two novel methods to mine useful patterns from enormous and fast evolving DNS log data. Firstly, we proposed frequent episode mining algorithm, thereby predicting the incoming query volume and detecting anomaly. Secondly, we partitioned the query volume time series data into clusters by using DBSCAN clustering algorithm, and further insights of each cluster are studied. Thirdly, a novel pattern, consistent episode, is proposed as well as the method to discover consistent episodes in a sequence database. Consistent episode is useful in finding the spatial-temporal correlation among the DNS servers, as well as predicting anomalies. The effectiveness of our methods has been demonstrated by the experimental results on a real-world DNS query traffic log dataset. Based on our experimental results, it is evident to see that data mining-based approaches are valuable and promising in assisting DNS service.

Acknowledgments Project is supported by National Natural Foundation of China #61202312/70921061 and China Internet Network Information Center.

References

1. Jung J, Sit E (2004) An empirical study of spam traffic and the use of DNS black lists. In: Proceedings of the 4th ACM SIGCOMM conference on internet measurement, pp 370–375

2. Ishibashi K, Toyono T, Toyama K et al.(2005) Detecting mass-mailing worm infected hosts by mining DNS traffic data. In: Proceedings of the workshop on mining network data at ACM SIGCOMM, USA, pp 159–164
3. Klein A. BIND 9 DNS cache poisoning. http://www.securiteam.com/securitynews/5VP0L0UM0A.html
4. US-CERT. The continuing denial of service threat posed by DNS recursion. https://www.uscert.gov/sites/default/files/publications/DNS-recursion033006.pdf
5. Cheng J, Li X, Yuan J et al.(2010) K-means based analysis of DNS query patterns. J Tsinghua Univ 17:80–87
6. Wang Z, Li X, Yan B (2010) Abnormity detection of DNS query traffic at CN top level domain server, Technical Report. http://www.docin.com/p-629288242.html
7. Xu H, Li Z, Zhou L (2007) Botnet detection methods in DNS traffic. J Xiamen Univ 53:98–99
8. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceedings of the 20th international conference on very large databases, pp 487–499
9. Ester M, Kriegel HP, Sander J, Xu X (1996) In: Proceedings of 2nd international conference on knowledge discovery and data mining, Portland

Hongyuan Cui received her B.S. degree from Shandong University, China, in 2014. She is currently a graduate student in School of Computer and Control, University of Chinese Academy of Sciences. Her research interests include data mining, high performance computing, big data, etc.

Jiajun Yang received his B.S. degree from Sichuan University, China, in 2013. He is currently a graduate student in School of Computer and Control, University of Chinese Academy of Sciences. His research interests include data mining, cloud computing, big data, etc.

Ying Liu received her B.S. degree from Peking University, China, in 1999, the M.S. degree and the Ph.D. degree from Northwestern University, Evanston, IL, USA, in computer engineering in 2001 and 2005, respectively. She is currently an associate professor in School of Computer and Control, University of Chinese Academy of sciences. She also holds an adjunct appointment with Fictitious Economy and Data Science Research Center of Chinese Academy of Sciences. She served as the co-chair of National Scientific Data Conference, 2014, the workshop chair for workshop on High Performance Data Mining with 7th International Conference on Data Mining (ICDM), 2007, and the workshop chair for workshop on High Performance Data Mining with 7th International Conference on Computational Science (ICCS), 2007. Her research interests include data mining, high-performance computing, and business intelligence. She has published more than 50 research papers in international conferences and journals. Prof. Ying Liu is PI and Co-PI of 3 National Nature Science Foundation Projects of China. She is the receiver of NVidia Global Professor Partnership, Agilent Research Project Grant, and AMD Research Grant.

Zheng Zheng is currently a graduate student in University of Chinese Academy of Sciences and Computer Network Information Center, Chinese Academy of Sciences. His research interests include data-intensive computing, big data processing.

Kaichao Wu is a professor of Computer Science in Computer Network Information Center, Chinese Academy of Sciences. He received his PhD from University of Chinese Academy of Sciences in 2008. His research interests include data-centric cyberinfrastructure, spatial big data analysis.