Symmetry Degree Measurement and Its Applications to Anomaly Detection

Tao Qin, *Member, IEEE,* Zhaoli Liu, Pinghui Wang, Shancang Li, Xiaohong Guan, *Fellow, IEEE,* and Lixin Gao, *Fellow, IEEE*

***Abstract*—Anomaly detection is an important technique to identify network unusual behaviour patterns and keep the network under control. The network attacks are increasing in both number and sophistication. To avoid causing significant traffic patterns and being detected by existing techniques, many newly attacks tend to gradually adjust their behaviors, which always generate incomplete sessions due to their running mechanisms. In this work, we employ the behavior symmetry degree to profile the anomalies and further identify and detect unusual behaviors. By identifying the incomplete sessions generated by unusual behaviors that only contain forward or backward packets, we first proposed a behaviour symmetry degree to capture the features of these unusual behaviors; then, we employ the sketch to calculate the symmetry degree of an unusual behaviour to improve the identification efficient for online application. To reduce the memory cost and probability of collision, we divide the IP addresses into four segments that can be used as keys of the hash functions. To further improve the detection accuracy, a threshold selection method is proposed for the dynamic traffic pattern analysis. Then, the hash functions in the sketch are designed using Chinese remainder theory, which can trace the IP addresses associated with the anomalies analytically. We tested the proposed techniques based on the traffic data collected from northwest center of CERNET (China Education and Research Network) and the results show that the proposed methods can effectively detect anomalies in complex networks.**

***Index Terms*—Smart attacks, Behavior patterns, Symmetry degree, Degree sketch, Anomaly tracing.**

I. Introduction

ANOMALY detection aims at identifying the presence of unusually network traffic patterns, which has become an increasingly critical challenge for network management and security. In recent, many different kinds of threats continually appear and the new slowly and continually attacks are becoming more intelligent than we expected [1], [2]. These nearly attacks tend to gradually change their behaviors to reduce the change of network traffic patter, such as reduce the total bytes, number of packets and flows, etc., to avoid being detected by existing techniques. In this paper, we name those attacks designed with specific strategies for avoiding detection as smart attacks, such as APT (Advanced Persistent Threat) attack [1], [2], Botnet [3], Stuxnet [4], etc. Most traditional anomaly detection methods mainly examine the statistical patterns extracted from the entire raw traffic volumes will lose their efficiency in detecting those smart attacks [5], [6], [7], [8], [9]. Furthermore, the traffic statistical features have been changed with the increasing number of new applications [10], [11]. As a result, to extract stable and efficient traffic patterns from the massive raw traffic data and improve the ability of detecting smart attacks is a key challenge. To address these limitations, we propose the behavior symmetry degree to characterize abnormal host behaviors. We then combine this with sketch to detect the abnormal hosts with high efficiency and accuracy.

In this work, we use directed graph $G(V, E)$ to model a network as shown in Fig 1, in which vertices V denote the hosts and edges E represent the communication between hosts [12]. In general, the NIDS (network based intrusion detection system) collects traffic data from the egress routers, and the hosts set V can be categorized into two groups: *internal hosts* and *external hosts*. *Internal hosts* contains all hosts inside the monitored network and *external hosts* contains all hosts outside that can communicate with internal hosts. In G, each edge $e \in E$ represents a communication session between the internal and external hosts, which is a set of aggregated packets with the same internal and external addresses. It should include both the forward and backward packets under normal circumstances, we use bidirectional relationship denoted by the black edges to represent it as show in Fig.1.

From our previous works [13], we noted that most of those abnormal behaviors generate many incomplete flows, which only contain the forward or backward packets. As shown in Fig.1, the red links between two hosts denotes incomplete sessions. In this work, we employ the connection degrees to capture this kind of characteristics, whose efficiency has been verified in [14], [15]. To detect whether the internal hosts are abnormal or not, we mainly focus on analyzing the characteristics of the internal host. For the internal hosts profiling, we define two connection degrees: (1) *In Connection*

The research presented in this paper is supported in part by the National Natural Science Foundation of China (61772411, 61672026, 61602370, U1736205), Project JCYJ20170816100819428 supported by SZSTI and China Scholarship Council (201706285018).

T. Qin and P. Wang are with MOE Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an, China. E-mail: (qin.tao, phwang)@mail.xjtu.edu.cn.

Zhaoli Liu is with MOE Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an, China; e-mail: lzl.liu@stu.xjtu.edu.cn.

S. Li is with the Department of Computer Science and Creative Technologies, University of the West of England, Bristol, U.K; e-mail: shancang.li@uwe.ac.uk.

X. Guan is with the Shenzhen Research School, Xi'an Jiaotong University Shenzhen Guangdong, China, 518057; He is also with MOE Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University Xi'an, China. e-mail: xhguan@mail.xjtu.edu.cn.

L. Gao is with the Multimedia Networking & Internet Lab, department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, USA01002; email: lgao@ecs.umass.edu.

Corresponding author: T. Qin, P. Wang and X. Guan.

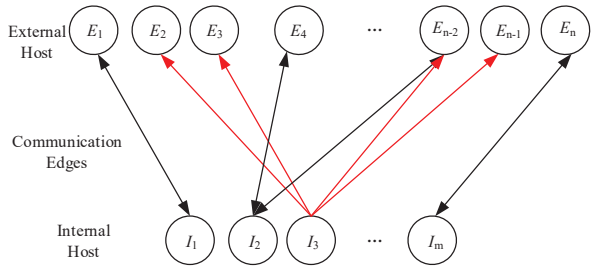


Fig. 1. Communication model based on directed graph

Degree of specific Internal host (*ICDI*), which represents the number of the unique external IP addresses which send packets to the specific internal IP address during a time window T; (2) Out Connection Degree of specific Internal host (*OCDI*), which is the number of the unique external IP addresses which receive packets from the specific internal IP address during a time window T. Both the *ICDI* and *OCDI* can effectively profile the abnormal hosts with incomplete sessions, e.g., in Fig.1 I_3 is attacking other hosts by sending scanning packets to explore vulnerable targets, so it holds a bigger *OCDI* and a smaller *ICDI* because most of the scanning requests do not receive responses. In this work, we propose the Symmetry Degree of specific Internal host (*SDI*), which can be defined using the maximum ratio between the *OCDI* and *ICDI* to characterize this kind of abnormal host behaviors effectively.

To improve the efficiency and scalability of anomaly detection, we employ sketch for *SDI* calculation and anomaly tracing. We divide the internal IP addresses into four segments based on the structure of IPv4 addresses. Each segment is selected as the key of the hash functions in the sketch to minimize the memory cost and the probability of collision. Meanwhile, we use the Chinese remainder theorem to design hash functions, and make it to be a reversible sketch. We can only use the information from hash functions to reconstruct the keys without using any keys' information. In other words, we can obtain the anomaly-related IP addresses efficiently with analytical calculation, and the computational time is constant without considering the size of traffic volume. We also design a threshold selection method based on the dynamic traffic patterns to select suitable thresholds and improve the detection accuracy.

Finally, to evaluate proposed methods in real traffic scenario, a real dataset collected from northwest center of CERNET is used to test the proposed method and results are compared state-of-the-art algorithms. Experimental results demonstrate that the proposed method outperforms existing methods, it can detect anomalies with high accuracy, low computation and memory cost.

The main contributions of this paper can be summarized as follows:

1) A graph based network model is proposed to describe the communication patterns between end hosts, in which a symmetry degree id defined that can be used to characterize the anomalies effectively. The proposed model can aggregate packets with the same internal and external IP addresses into one session, which can significantly reduce the record amount and computational complexity.
2) IP segments are used as the keys of the hash functions in sketch, and a reversible sketch is designed using the Chinese remainder theorem. By doing this, the model can minimize the memory cost and collision probability. Furthermore, the proposed method can trace the anomaly-related IP addresses for efficient security management with analytic calculation and constant computational time.
3) The proposed methods are tested using large-scale, real-world traffic datasets. Compare with existing methods, the proposed method can achieve a better performance with lower computation overhead.

II. Related Works

Anomaly detection has become an important issue for the network management and attracted significant attentions from researchers. The main idea is to build normal behavior model using historical data, and then detect the behaviors deviated from the models.

In network management and security, anomaly detection can be roughly classified as HIDS (host based intrusion detection) and NIDS (network based intrusion detection) according to the data sources used. HIDS usually uses data sources like keystroke biometrics [16], mouse dynamics [17], system calls [18], *etc*. The HIDS methods are effective in host level anomaly detection, but the scalability of HIDS is limited by the efficiency of data collection. The most popular data used for NIDS is the traffic data. Meanwhile the statistics-based models are the most popular and widely used methods. The basic idea of statistics-based models is to analyze the statistical characteristics of the traffic patterns (total number of bytes, total number of packets, *etc.*), and then detect the anomalies depends the significant pattern changes caused by attacks such as DDOS [6], [7]. The source and destination IP addresses in the packets can be treated as different hosts or users, and we can perform anomaly detection by analyzing the communication patterns among those hosts [19], [20], [21]. There are a number of statistical methods can be used for this kind of pattern analysis, such as the *Markov chain* and binary composite hypothesis testing [22], [23], [24]. However, with the increasing number of network users and bandwidth, especially sophisticated hackers, they tend to gradually change their behaviors. Those kind of methods will reduce their efficiency since anomaly behaviors never cause significant changes in traffic volumes again.

To investigate the traffic patterns more efficiently, CISCO proposed the netflow model [25], which aggregates is consisted by a group of packets with the same source and destination IP addresses, ports, *etc.* It is a logic links between hosts and provides a way to profile the patterns by hosts. Based on this model, a number methods based on machine learning and data mining are proposed to perform anomaly detection [26], [27], [28]. However, the *netflow* model is one kind of non-interactive models, it divides the packets in the same session into two or more flows, which is ineffectual for

behavior profiling. Another issue for *netflow* is that the number of flow records could be huge and the tasks of monitoring and analyzing may encounter serious storage and computation difficulties.

There are a number of methods have been developed for reducing the number of data records to be processed for real-time monitoring. Sampling is one of the most simple and efficacious ways [29], [30], [31]. However, sampling may cause missing important flow fingerprints especially when a large number of flows mixed with only a small number of flows generated by anomaly behaviors [32], [33]. Another one is to extend the aggregated scale and extract the main traffic matrix characters. ODflow model [34] and regional flow model [35] are developed for large-scale traffic monitoring. The experimental results using the actual traces in 10Gbps network environment also verify their efficiency in traffic profiling. But those kind of models have low efficiency in detailed traffic patterns profiling and smart attack detection. Furthermore, different aggregated scales may generate different results. It is difficult to select the suitable scales for traffic aggregation.

Sketch based methods are widely used in dealing with massive packets in real-time network monitoring. Sketch is a kind of probabilistic dimension reduction techniques, which "sketches" a huge number of flows into a probabilistic summary. It is clear that the goal of traffic measurement and monitoring based on the traffic flow information approximately reconstructed using the information of the sketch. Several traffic measurement methods using sketch are proposed in recent, such as finding heavy hitters, heavy changes and estimating flow size distribution [36], [37], [38], [39]. To further improve the processing efficiency and perform scalable traffic anomaly detection, sketch can also be implemented in hardware using FPGA [40], [41]. The sketch can also be used to detect anomalies by combining with other statistical methods, such as the CUSUM algorithm [42], *etc*. We can also extract suitable features and combine them with sketch for detecting specific anomalies with high accuracy, such as the *LD-Sketch* and *SkyShield sketch* [43], [44]. Generally there are mainly two issues in the sketch related methods: (1) designing suitable hash function for the sketch, appropriate hash function can reduce memory used and probability of collision, which are key for the goal of anomaly related IP address tracing; (2) selecting suitable features as keys, such as the source IP address or destination IP address, or their combinations, and different keys are suitable for detecting different anomalies.

Additionally, attacks in the network today become more and more intelligent, they can gradually change their behaviors to avoid causing significant pattern changes, such as the changing trend of total bytes and the total number of flows. It is a difficult task to find those slight changes from the massive traffic patterns. Researchers focus on extracting new features from different views for detailed behavior profiling. Zhang *et al.* in [45] employ the underlying triggering relations of network events to detect stealthy malware activities. Zhang *et al.* in [46] used the dependency of network requests to detect stealthy malware activities. In our previous works, we developed a reversible user-embedding framework for this kind of behavior detection [47], extracting measurable and intelligible features and design efficient computing method are becoming more and more important in detecting those smart attacks.

Focus on the above problems and enlightened by the related works, in this work we proposed the symmetry degree sketch to perform degree calculation, anomaly detection and anomaly related IP address tracing.

III. Design Goals and Threat Model

The works in [10], [11] have shown that the statistical network traffic patterns have been changed and it is very difficult to identify the slight abnormal patterns from the massive traffic patterns. In this section, we mainly focus on designing a lightweight and efficiency framework that can match the needs of anomaly detection in the network today. We also give the assumptions about the adversary and threat model to help the readers to catch the key points of our work easily [48].

A. Design goals

Taking both lightweight and efficiency into account, in this paper we design an online anomaly detection method that meets the following design goals:

1. Simple and usable: The method should be deployable on most enterprise network without requiring installation of new hardware components. The feature used should be easily obtained and can be used to capture the pattern difference between the normal operations and attacks.

2. Lightweight: The method should be capable of calculating the features with little computational resources. Its computational complexity should change slightly with the change of the network scale.

3. Efficiency: The method designed should be able to detect the abnormal behaviors appear recently, such as attacks intendedly control their behavior to avoid cause obvious changes in the traffic patterns.

B. Assumptions about the adversary and threat model

In a network, the attackers can explore the vulnerabilities and then control weak hosts with vulnerabilities for unpermitted behaviours, such as information theft, service monitoring *etc.* To detect those ongoing attacks and develop reasonable model, in the following sections we will elaborate on the assumptions about the adversary and threat model.

Assume that the adversary could be someone who has limited knowledge of the monitored network but somehow want to access to the network. In other words, the adversary does not have the physical accessible power but can only explore the vulnerabilities using attack technologies. He does know the detailed configurations and the defense policies of the target network. In particular, our method is designed to secure against the following common types of attacks:

1. Scanning Attack: Including the network and port scan, the attacker tries to explore the vulnerabilities of the hosts inside the target network by chance without any prior knowledge.

2. Stealthy Attack with Heartbeats: Including the Trojan Horse, Botnet and other similar virus. The virus has successfully infected the host. And the attacker utilizes the heartbeats to master the running status of the viruses in the host system.

3. Worm and similar attacks: Those attacks can explore the potential targets and propagate automatically. During the propagation phase, the potential targets exploring by chance will result in a large *SDI*.

4. DDoS Attack: The attacker employs many bots to occupy most the computing resource of the specific server and make it unavailable to normal users.

IV. Feature Extraction and Framework Designed

To achieve the goal of design a lightweight and efficient anomaly detection method, we employ the model in Fig 1 to design a network model to reduce the number of flow records. We also design easily extracted and understand features for measuring the difference between the attacks and the normal operations.

A. Network model and behavior feature extraction

Assuming that the monitored network has *m* hosts and they can interact with *n* external hosts. Let $I_i(1 \leq i \leq m)$ denote the *i-th* internal host, and $E_j(1 \leq j \leq n)$ denote the *j-th* external host, respectively. Assume s_{ij} equals to 1 when $(1 \leq i \leq m, 1 \leq j \leq n)$ and there are packets transferred from the internal host I_i to the external host E_j. Otherwise s_{ij} equals to 0. Let d_{ji} equal to $1(1 \leq i \leq m, 1 \leq j \leq n)$ when there are packets transferred from the external host E_j to the internal host I_i, otherwise d_{ji} equals to 0. The communication patterns in the specific time window *T* between the internal and external hosts can be represented by the matrix *M(t)* in Eq.(1). This model can be used to capture the dynamic and exchange behavior patterns at the host level, its efficiency has been verified in our previous work for application classification [49].

$$M(t) = \begin{bmatrix} (s_{11}, d_{11}) & (s_{12}, d_{21}) & ... & (s_{1n}, d_{n1}) \\ (s_{21}, d_{12}) & (s_{22}, d_{22}) & ... & (s_{2n}, d_{n2}) \\ ... & ... & ... & ... \\ (s_{m1}, d_{1m}) & (s_{m2}, d_{2m}) & ... & (s_{mn}, d_{nm}) \end{bmatrix} \quad (1)$$

The proposed model aggregates the packets with the same internal and external addresses into one session, which can significantly reduce the number of records and further reduce the computational complexity. Based on the model proposed, we define and employ the following features to profile the communication behavior patterns.

Feature1: The In Connection Degree of specific Internal host *(ICDI)*: The number of unique external hosts which send packets to a specific internal host during the time window *T*, which is defined as

$$ICDI_i = \sum_{j=1}^{n} d_{ji}, 1 \leq i \leq m \quad (2)$$

Feature2: The Out Connection Degree of specific Internal host *(OCDI)*: The number of unique external hosts which receive packets from the specific internal host during the time window *T*, which is

$$OCDI_i = \sum_{j=1}^{n} s_{ij}, 1 \leq i \leq m \quad (3)$$

Feature3: The Symmetric Degree of specific Internal host *(SDI)*:The symmetric degree of the specific internal host is defined using Equation 4, a constant 1 is added to the denominator and numerator to avoid zero. Usually there are two ways to define the symmetric degree: the first one can be obtained using the Out connection degree divided by the In connection degree; the other one can be obtained using the In connection degree divided by the Out connection degree. In this paper, we select the bigger value of them as the symmetric degree to mine the anomalies.

$$SDI_i = \max\{\frac{OCDI_i + 1}{ICDI_i + 1}, \frac{ICDI_i + 1}{OCDI_i + 1}\}, 1 \leq i \leq m \quad (4)$$

B. Framework of the methods designed

To achieve real-time anomaly detection in large scale networks, we propose an efficient framework as shown in Fig 2, which contains the following four steps: **step 1**: Traffic collection. The network traffic is collected from the northwest center of CERNET, the offline traffic used for performance evaluation is collected using the Coral Reef developed by CAIDA (Cooperative Association for Internet Data Analysis) [50].

step 2: Behavior symmetry degree extraction. To characterize the anomalies, we extract the behavior symmetry degrees by analyzing the packet set with the same internal and external IP addresses in a specific time window *T*. We design the degree sketch for efficient calculation. To reduce the memory cost and collision probability, we select the IP segments as the keys for hash functions.

step 3: Anomaly detection. We perform anomaly detection mainly based on the analysis of *SDI*. If *SDI* at a host is bigger than a selected threshold, we claim that we detected an anomaly. To adapt with the dynamic changing trends of traffic patterns, a threshold selection method is proposed.

step 4: Anomaly related IP address tracing. We design a reversible sketch by using the Chinese remainder theorem, so that we can trace the anomaly-related IP addresses efficiently. And then we can control the anomalies with suitable policies to keep the network under control.

V. Data Acquisition

A. Anomalies Data Acquisition

In this work, we collected more than 500 attacks, including the Worm, botnet, Scanning and DDOS attack, which can be categorized into three groups: (1) The users report, the users report their attacks they suffered to the administrator and ask for help for solution; (2) Botnet detection system, we established a honeynet in our LAB that includes more than 40 hosts (http://botwarden.xjtu.edu.cn). The anomalies captured by the honeynet are reported to the botnet detection system; (3) Monitoring system set in the campus network center, in

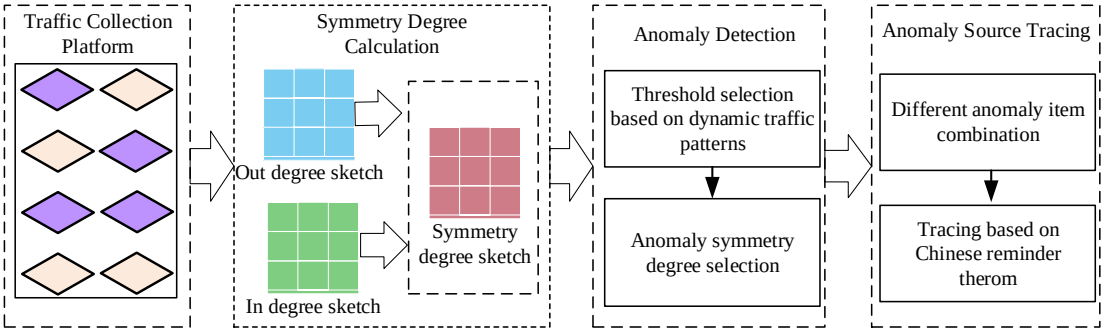


Fig. 2. overall flow of the algorithm

which each attack contains all the raw packets of one specific victimized host or hacker. The detailed information of these attacks is shown in Table **??**. We also verified that the behavior patterns reflected by the collected data are different with that of the normal operations.

B. Traffic Data Collected in Our LAB

We collected one actual traffic trace from the egress router of our LAB with 500 users. These users are from different labs in the departments of school of electrical and information engineering. Totally there are about 500 hosts with global IP addresses, including two servers, one is a FTP data server with an address as `ftp://202.117.54.250:4021`. It is mainly used for video and software sharing in the LAB. Another is a WEB server with URL `http://nskeylab.xjtu.edu.cn`, the IP address is `202.117.54.254`. It is mainly used to provide news and mail services to the students and faculties. Fig **??** shows the network topology. We filter the traffic collected using the label methods used in section V.D to filter the anomalies.

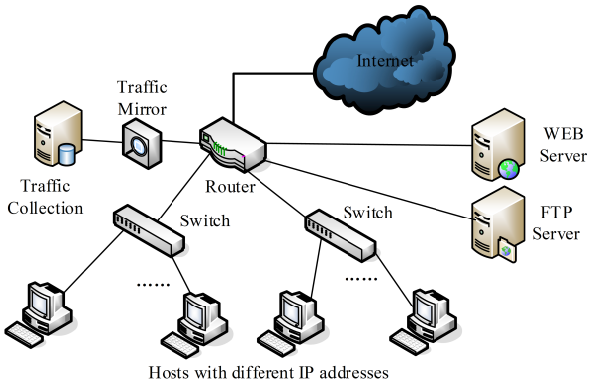


Fig. 3. Network topology of our LAB

C. Large scale data collected from our campus network

The large scale data used is collected from the Northwest Regional Center of CERNET, its topology is shown in Fig 4. The network being monitored is the campus network which contains more than 30,000 end users with self-governed IP addresses, including students, faculty members and contract personnel from service providing companies. All of the traces used in this paper are collected at an egress router (B2) with a bandwidth of 10Gbps using the traffic collection tool Coral Reef [50] for the time horizon of more than 20 hours ranging from 2017 to 2018. We choose traffic traces that cover three different time periods to collect traffic traces with different kind of behaviors. The first one is the time period from 2:00 AM to 6:00 AM, when most of the students are asleep. The time period from 8:00 AM to 12:00 AM, when most of the students are having classes. We also select a time period from 8:00 PM to 12:00 PM, when most of the students are using the Internet, the basic information of the traces collected is shown in TABLE 2. Where # of PKT denotes the total number of packets, # of IH denotes the number of internal hosts and # of VA denotes the number of verified anomalies in the trace. In this way, we try to make the data used contains more kinds of behavior profiles and representative. Compare with the public datasets, the collected data set has the following advantages: 1) the detailed network configuration, we have the detailed listed of IP addresses inside the monitored network, which is employed for connection degrees and *SDI* calculation of the internal hosts. 2) the data is collected from a large scale network, which is useful for verify the efficiency of our methods.

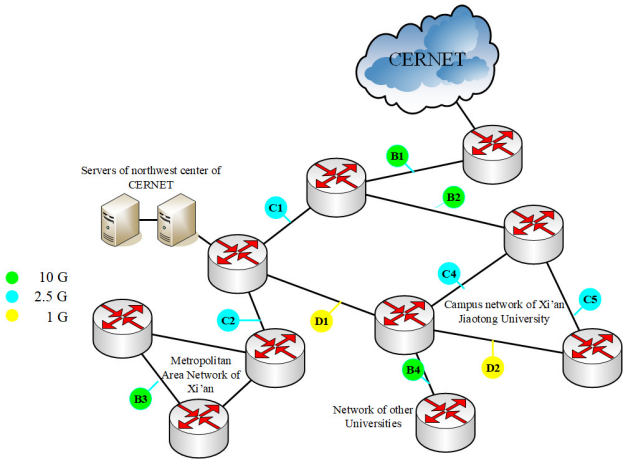


Fig. 4. Topology of the northwest of CERNET

TABLE 1
The attacks collected and their influences on *SDI*

Type	Number of Attacks	Simple attack descriptions and their influences on *SDI*
Worm	230	Hosts infect the worms search other targets automatically by sending many scanning packets, but there is no response to most of the scanning packets. The infected host holds larger *OCDI* than *ICDI*, which results in a larger *SDI*. In this paper the worm used include the Worm.WhBoy.cw, Ransomware, etc.
Botnet	60	Bot masters control many hosts that are named as Bots, and the master can launch attacks to others using those Bots. The Bot masters receive many heart beating packets from different Bots but without response to those packets, thus the *ICDI* of the master is larger than the *OCDI*, which results in a larger *SDI*.
Scanning	210	Scanning is the first step for hackers to search targets. The scanning attacks used in this paper are the internal hosts explore the targets from the external hosts. Both the fast and slow scanning attacks send many scanning packets to find the potential targets and most of them do not receive responses. The *OCDI* of the scanner is larger than *ICDI*, which generates a larger *SDI*.
DDoS	20	The DDoS attack used in this paper is the DDoS attacks to the bbs server (bbs.xjtu.edu.cn) and the mail server (mail.xjtu.edu.cn). These attacks send many packets from different attackers to one specific target and the target does not generate response packets, the *ICDI* of the target is larger than the *OCDI*, which generate a larger *SDI*.

D. Anomaly identification from the trace

We integrated three different ways to identify the anomalies from the raw packets to construct the benchmark to evaluate our methods. Firstly, we build the black IP address list by collecting the security reports from different third parties and the DNS traffic collected from our campus network. The reports collected include the security report published by the Computer Emergency Response Team of Northeastern University (NEUCERT). NEUCERT publishes the anomaly IP addresses they detected and updates the list every five minutes, which is publicly available at http://antivirus.neu.edu.cn/scan/ssh.php. We also collect the abnormal urls from the security company 360. The report's address is (https://webscan.360.cn/url), which is updated every day. We also use the detection results of the Botnet detection system set in our LAB to monitor whether there are Bots in our campus network. If there are Bots, we can obtain the domains they used for communication with their masters. Combined with the DNS traffic data collected from our campus network and the abnormal domains obtained, we can obtain the abnormal IP addresses who query these abnormal domains.

Except the black IP list, we also use the Netflow Analyzer (https://www.manageengine.cn/products/netflow/) as a tool to mine the anomalies in the data trace. We also manually analyze the statistical traffic characteristics, such as the flow size, flow life time, to determine whether there are anomalies. The methods used include methods developed in our previous works [35], [15]. By combine those technologies, we try to identify the anomalies. We employ those verified anomalies to evaluate the methods developed. The detailed information about the anomalies identified is shown in TABLE 3.

TABLE 2
Simple description on dataset

Trace	Duration	Begin Time	# of PKT	# of IH	# of VA
One	5.5	2017.10.15 20:10	378451461	4421	19
Two	5.5	2017.12.27 08:30	305452461	3620	28
Three	5.5	2018.03.12 02:08	264513582	3608	15

TABLE 3
Simple statistics on dataset

Trace	Scan	DDoS	DoS	Worm	Botnet
One	12	2	1	3	1
Two	17	4	2	1	4
Three	10	3	0	1	1

VI. Feature Analysis

A. SDI measurement using attack and normal behavior trace

We analysis the *SDI* of the anomalies and that of the data collected from our LAB. The analysis results are shown in Fig 5. As the figure shows, the *SDI* values of more than 85% of the abnormal hosts are bigger than 40. We also analyzed the abnormal hosts whose *SDI* values are less than 40, and we find those hosts are the Bots which are controlled by the Bot master outside the campus network. They send heartbeat packets to the masters slowly. In this situation, the Bot masters hold larger In connection degree than the Out connection degree. We can simply include the *SDI* of the external address in the features to detect this kind of attacks. Furthermore, the *SDI* of the normal behaviors is around one. There are obvious differences between the *SDI* of anomaly and normal behaviors, which verify the efficiency of *SDI* in anomaly detection.

B. SDI measurement using large scale traces

We also verify the efficiency of *SDI* in profiling anomalies by analyzing the traffic traces collected from the CERNET. We firstly analyze the distribution of the *SDI*, including the CDF (Cumulative Distribution Function) and PDF (Probability Distribution Function), and the analysis results are shown in Fig 6(a). It can be seen that most of the hosts in the network being monitored approximately have the same OCDI and ICDI, and the *SDI* approximately equals to one. This analysis results versified that most of the users use the network for information exchange, they send request packets to the servers and receive the response packets from the servers. It can also be found that some hosts hold either large OCDI or ICDI and the *SDI* is far away from one. This is caused by the incomplete sessions, which means the host sends the request packets but it does not receive the response data or it receives

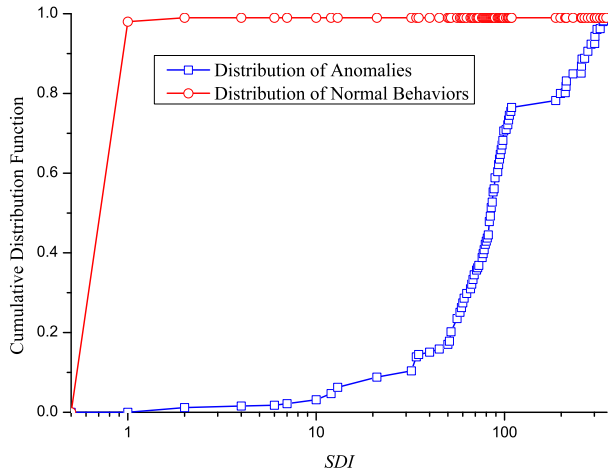


Fig. 5. *SDI* distribution

many connection requests from many hosts but it does not response them. These characteristics have been explored in our prior works [43] and we claim that they are useful for detecting anomalies.

To further mine the characteristics of the incomplete sessions, we analyse the distribution of the incomplete session size and the distribution of the number of incomplete sessions that each host holds. The analysis results are shown in Fig 6(b), where the outer is the results of the flow size and the inner is that of the number of incomplete sessions each host holds. As the results show, most of the incomplete sessions have less than three packets, which means if the flow is a TCP flow, it does not complete the three-way handshake process, which is the necessary beginning process for a TCP connection. If the flow is a UDP flow, it means there is only small pieces of data exchanged, as the UDP mainly be used for files or multimedia transfer between hosts and usually there are thousands of packets in this kind of applications. Thus we claim most of the incomplete sessions are generated by attacks. From the inner figure we can find that most of the users only hold less than 15 incomplete sessions during the monitoring time period, this may be caused by the random normal behaviors, for example you input a wrong domain in the browser and you will receive no response. However, we also notice that there are several hosts hold hundreds or thousands of incomplete sessions, which is obvious different from normal operations. Those results verify that the larger *SDI* is mainly caused by abnormal behaviors and its efficiency in anomaly detection.

C. SDI robustness analysis

1) Sensitivity to different adversary's policies: Assume that the adversary does not have the physical accessible power and he can only explore the vulnerabilities using attack technologies. The most common policy used is scanning the network to explore potential targets by chance. Those attacks will generate many incomplete sessions and can result in a large *SDI*. To avoid cause obvious changes in *SDI*, the adversary can slow down the attempt frequency or adopt some special policies. If he slow down the attempt frequency, the exploring time will increase dramatically. Such as he attempt 20 times per minute originally, if there are 2,000 hosts in 200 minutes. If the target network contains more hosts, the exploring process will be longer. And this is not acceptable for the hacker. As he has no knowledge about the target network, many attack policies is not suitable for this case. Thus the *SDI* is sensitivity to the adversary's exploring patterns.

2) Sensitivity to different attack phases: Although *SDI* is extracted based on the pattern difference between the attack and the normal operations, it is only sensitive to attacks patterns of different phase. Some virus, such as Trojan Horse and some Botnets, may infect the host by Email, USB desk, or Web script. The *SDI* is not sensitivity to those infection behaviors. But for the virus explores the targets by scanning, such as worm, the *SDI* can detect those propagation behaviors. During the attack phase, such as the DDoS attack, there will be many incomplete connections, and the *SDI* can be employed to detect those attacks. In other words, *SDI* cannot detect the abnormal behaviors with both req-ack packets or behaviors do not cause obvious changes in *SDI*. Behaviors with both req-ack packets hold the forward and backward packets at the same time and the *SDI* is similar that of the normal behaviors. It is difficult to detect those anomalies in the network today. We can detect those anomalies by combining with other methods, such as mining the co-occurrence or periodic behaviors from a long time period traffic monitoring.

VII. Feature Calculation and Anomaly Detection

A. Symmetry degree sketch design

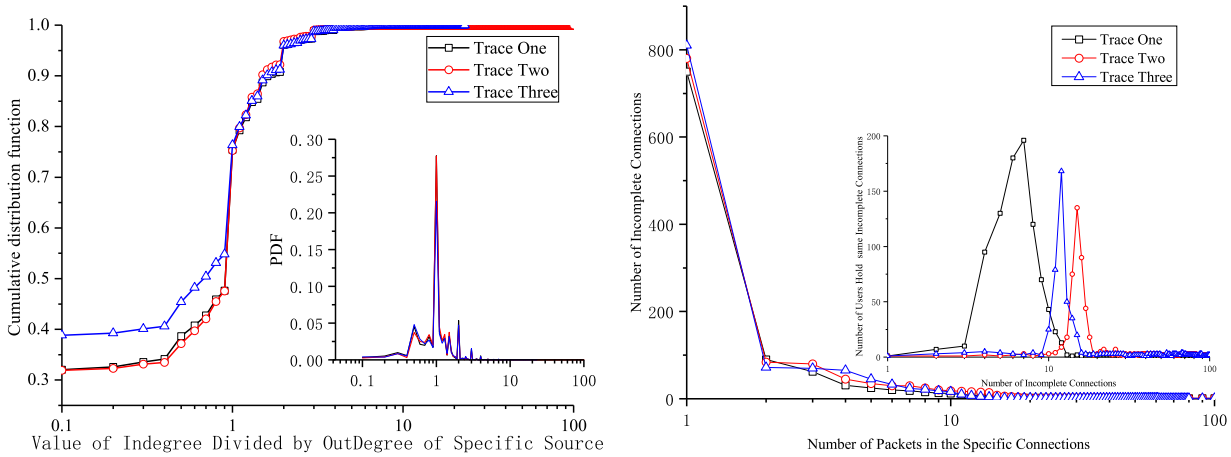
Sketch methods have been studied for several years and are widely used in traffic measurement and anomaly detection. Here we employ the sketch for symmetry degree online calculation.

The Out and In connection degree sketches are denoted as B^{out} and B^{in}. We have $B = (B_1, ..., B_H)$, in which $B_i (1 \leq i \leq H)$ is a $v \times m_i$ bit arrays and H is the number of data arrays. The columns of $B_i[j][k] (0 \leq j \leq v, 0 \leq k \leq m_i)$ are associated with a hash function $h_i : \{0, 1, ..., n-1\} \rightarrow \{0, 1, ..., m_i - 1\}$, where *n* is the size of the space of source IP addresses. All rows in B_i share a hash function $f : \{0, 1, ..., E-1\} \rightarrow \{0, 1, ..., v-1\}$, where E is the size of the destination IP addresses. The update process is shown in Fig 7. Initially, the bits in each $B_i (1 \leq i \leq H)$ are all set to zero, when a packet $p_i = (s_i, d_i)$ arrives, each B_i is updated by setting the bit in its row $f(d_i)$ and column $h_k(s_i)$ equal to 1, where the label d_i is the destination IP address and s_i is the source IP address, as in Eq.(5).

$$\begin{cases} B_i^{out}[f(d_i)][h_i(s_i)] = 1, \text{ if } p_i \text{ is forward packet} \\ B_i^{in}[f(s_i)][h_i(d_i)] = 1, \text{ if } p_i \text{ is backward packet} \\ 1 \leq k \leq H \end{cases} \quad (5)$$

For one specific source IP address *i*, there is no any other hosts mapped to the same items of $B_1(i)$,$B_2(i)$,...,$B_H(i)$ at the same time. Thus we can obtain the B_i of specific user by calculating the $B_1(i)$,$B_2(i)$,...,$B_H(i)$ using equation shown in equation 6, and then we can obtain its *OCDI(i)* and *ICDI(i)*.

$$B(i) = B_1(i) \otimes ... \otimes B_H(i) \quad (6)$$



(a) The distribution of the *SDI*

(b) The distribution of the flow size

Fig. 6. Different feature distributions

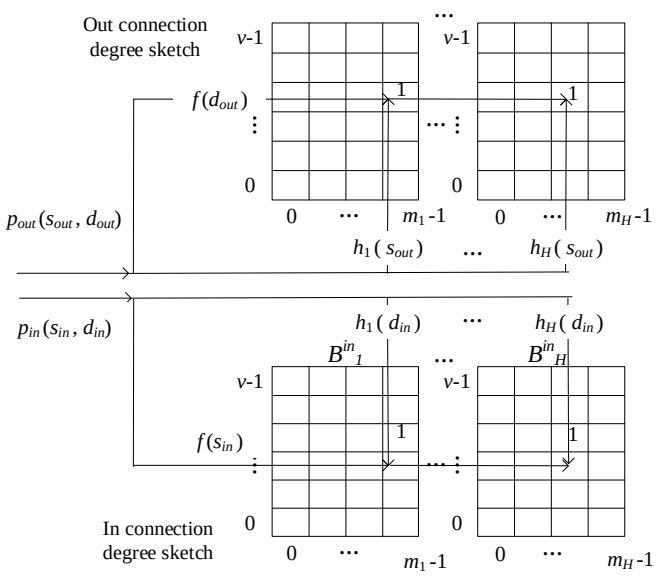


Fig. 7. Update procedure of the connection degree sketch

To minimize the memory space used and reduce the probability of collision of degree sketch, we divided the IP address into 4 segments according to the IPv4 structure. Each segment is one byte length and selected as the key of the hash function in the sketch. Each IP segment only contains 256 different situations, only and only if the four segments are conflicted at the same time, the whole IP address is conflicted. Compared with choosing the whole IP address as the key, our method can reduce the length of hash table as well as the probability of collision.

We employ the sketch with same structures and hash function groups to calculate the *OCDI* and *ICDI*, thus we can obtain the *SDI* directly based on the *OCDI* and *ICDI*.

B. Hash function design

In this works, we employ the Chinese remainder theorem to design the hash functions in the sketch to make the connection degree sketch reversible.

The hash function used in this paper is defined in Eq.(7, where m_1,...,m_H are different prime numbers.

$$h_j(x) \equiv xmodm_j, 1 \leq j \leq H \tag{7}$$

We select the IP segments as the keys of the hash functions of the degree sketches, and the mapping process are given in Fig 8. We firstly divide the IP address into four segments as IP_1, IP_2, IP_3 and IP_4. Using the first hash function $h_1(x) \equiv xmodm_1$, we can obtain the a_1, b_1, c_1 and d_1, then we set the item corresponding to the string $a_1.b_1.c_1.d_1$ in the hash table to 1. We process other IP segments similarly. Compare with the methods using whole IP address as the keys, our methods can reduce the length of the hash table and the probability of collision.

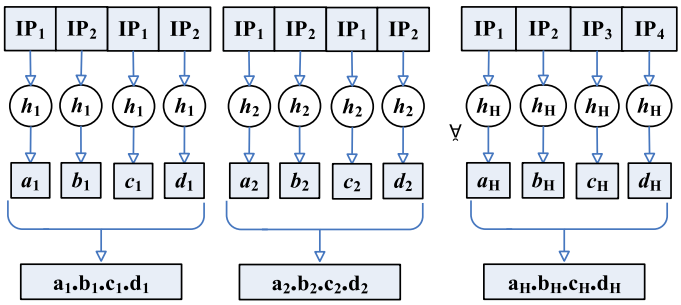


Fig. 8. IP segment mapping procedure

C. Anomaly threshold selection

For anomaly detection, we set a threshold for the feature extracted to judge whether there is an anomaly. We can use the probability of feature *Y* falls in the interval $(\mu - \varepsilon, \mu + \varepsilon)$ to check whether the item *Y* is abnormal or not, where μ is the expectation and σ is the variance of *Y*. Chebyshev's theorem is widely used in bounds selection. Based on the theorem, the probability of feature *Y* falls in the interval $(\mu - \varepsilon, \mu + \varepsilon)$ is

not less than $1-\sigma^2/\varepsilon^2$, where μ is the expectation and σ is the variance of *Y*.

Usually, the network is running in normal states and the traffic features are generated by normal operations. We can treat the *SDI* distribution as normal distribution according to its definition and the analysis results of Fig.6a. Thus we can use the Eq.(8) to the traffic calculate the upper and lower bounds. In this paper, we use the expectation μ of the *SDI* as the baseline, the difference of 2 or 3 standard variance σ from the baseline as the upper and lower bounds to construct the thresholds. If the symmetry degree at the monitored time point exceeds the range decided by the bounds, we regard it as an anomaly. To improve the detection accuracy, we select the bounds based on the dynamic changing patterns. A selecting time window is set and then the expectation and variance of the symmetry degree inside the window are selected to calculate the bounds. The size of the sliding window can be adjusted according to the detection results.

$$\begin{array}{l} P(|Y-\mu| \leq \sigma) \approx 68\% \\ P(|Y-\mu| \leq 2\sigma) \approx 95\% \\ P(|Y-\mu| \leq 3\sigma) \approx 99\% \end{array} \tag{8}$$

D. Anomaly related IP address tracing

The problem of tracing anomaly-related hosts can be described as when we detect abnormal indices, c_i in the hash tables, how to reconstruct the abnormal hash key x (the original IP addresses) correctly and quickly. In this paper, we first reconstruct the IP segments and then combine them into IP addresses. The detailed reconstruction process is described as follows:

Step1: Extracted the anomaly item combinations from each hash table and calculate their corresponding keys, if and only if there is only one item in each table, then we can combine the keys into the anomaly IP address directly; if not, turn to Step4.

Step2: Based on the extracted abnormal items, we can extract the different abnormal segments, such as the segment IP_1, from each hash table according to Eq.(9).

$$\begin{cases} IP_1 \equiv a_1 (mod m_1) \\ IP_1 \equiv a_2 (mod m_2) \\ ... \\ IP_1 \equiv a_n (mod m_n) \end{cases} \tag{9}$$

Based on the Chinese reminder theorem, Eq.(9) only has one solution as Eq.(10).

$$\begin{array}{l} IP_1 = \sum_{i=1}^{n} a_i t_i M_i \\ where \\ M = m_1 \times m_2 \times .. \times m_n = \prod_{i=1}^{n} m_i \\ M_i = M/m_i, \forall i \in \{1, 2, ..., n\} \\ t_i = M_i^{-1} \end{array} \tag{10}$$

Similar to the above process, other IP segments IP_2, IP_3 and IP_4 can be derived, and then we can obtain the abnormal IP address as IP(IP_1.IP_2.IP_3.IP_4).

Step3: If there are other abnormal item combinations which have not been processed, turn to Step4, otherwise we have extracted all the abnormal IP addresses.

Step4: Extract an abnormal item combination from the hash tables and go to Step2, then mark this abnormal item combination to be processed. If there is no new abnormal item combination that is not processed in the hash tables, the whole tracing process is ended.

E. Traced anomaly processing

After anomaly detection, to classify the anomalies detected according to their processing agencies and threat degrees is a main task. Based on the classification results, the administrator can deal with the high threats timely with his limited time and energy, in turn, keep the network under control.

There are two simply ways to classify the anomalies detected: Firstly, classify the anomaly detected based on their Connection degrees and Symmetry Degrees, we assume that the host with biggest connection degree is more dangerous to other hosts, and it need to be processed immediately; Secondly, in our previous work [51], an anomaly threat degree calculation method is developed based on the characteristics extracted from themselves. The proposed method can also be used here to qualify the threat degrees of the detected results. Those simple classifications can greatly improve the efficiency of the system security management.

As for control policy, we can employ the dynamic quarantine method to control the abnormal behaviors and reduce their influence based on the principle of "assume guilty before proven innocent", which has been widely used for control highly infectious disease. Zou *et al.* employed this method to control the propagation of internet worms [52]. We can control the internet access behavior of hosts with higher threat degrees by a soft dynamic quarantine method. Each host in the monitored network can be quarantined individually. And the quarantine on a host is released after a fixed quarantine time window, such as 120 seconds. Once the quarantine on a host is released, this host can be quarantined again if it is classified as anomalies again. This dynamic quarantine method has one obvious advantage, a false detection only lead to a quarantine on a normal host for a short time, thus its normal activities will not be interfered heavily.

VIII. Parameter Adjustment and Optimization

The proposed model consists of three important parameters, the number of hash functions in the sketch, the parameters of the hash function designed and the thresholds selected. The initial parameters can be selected based on the theoretical analysis and network management experiences, we need to adjust them to achieve better detection performance.

A. Metrics for performance evaluation

We use the precision and recall, which are widely used in the related literature to evaluate the performance of the developed method. The definitions of precision and recall are shown in Eqs. (11) and (12), in which the True positives (TP) denotes the number of the anomalies that are correctly classified as anomalies, False positives (FP) denotes the number of the

normal events that are wrongly classified as anomalies, and False negatives (FN) denotes the number of the anomalies that are wrongly classified as normal events, respectively.

$$precision = \frac{TP}{TP + FP} \tag{11}$$

$$recall = \frac{TP}{TP + FN} \tag{12}$$

B. Initial parameters selection

As the designed sketch is reversible and can be used for anomaly tracing, so the IP segments obtained from the reversible analytical process should be unique. Based on the Chinese reminder theorem, when integers $m_1, \ldots, m_H$ are co-prime numbers and big enough, there is only on solution for any arbitrary integers $a_1, a_2, ..., a_n$ in Eq.(13) under mode M, where $M = m_1 \times m_2 \times ... \times m_n$.

$$\begin{cases} x \equiv a_1(mod m_1) \\ x \equiv a_2(mod m_2) \\ ... \\ x \equiv a_n(mod m_n) \end{cases} \tag{13}$$

As each segment of the IP address is a byte length and the value is limited to 255, thus the product of all selected prime numbers should be greater than 255, as shown in Eq.(14).

$$m_1 \times m_2 \times ... \times m_n > 255 \tag{14}$$

Secondly, the sketch designed should use minimum memory for easily online application. In this work we select the IP segments as the keys of the hash functions, thus there are totally m_i^4 different item combinations after mapping based on m_i. After the selection of $m_1, m_2, \ldots, m_n$, the number of different items of the symmetry degree sketch should be $m_1^4 + m_2^4 + \ldots + m_n^4$. To minimize the memory used, $m_1^4 + m_2^4 + \ldots + m_n^4$ should be as small as possible.

Thus the original parameter selection problem can be converted to how to select the suitable $m_1, m_2, ..., m_n$, which satisfy that $m_1 \times m_2 \times ... \times m_n > 255$, $m_1, m_2, ..., m_n$ are positive prime numbers and $m_1^4 + m_2^4 + ... + m_n^4$ have the smallest summation.

C. Adjustment of the Hash Function

Based on the analysis in Section VIII.B, we find when H equals to 4 and the prime numbers as 2, 3, 5, 11, we can obtain the unique IP address with smallest memory cost. However, the experimental results show that this selection will cause high collision rate, that can result in a low recall and precision. Those parameters must be adjusted to achieve better detection performance. Firstly we give the recall and precision using a single hash table with different primes smaller than 100. And then we select the primes with high recall, precision and consume less memory to design the hash functions. The results are shown in Fig **??**. As the figure shows, when m_j equal to 2, trace 1 and trace 3 have very low recall and precision, approximately equals to 0. With the prime increases, the IP collision rate decreases and the recall and precision increase. With prime bigger than 23 the recall and precision of the three traces is basically above 90%.

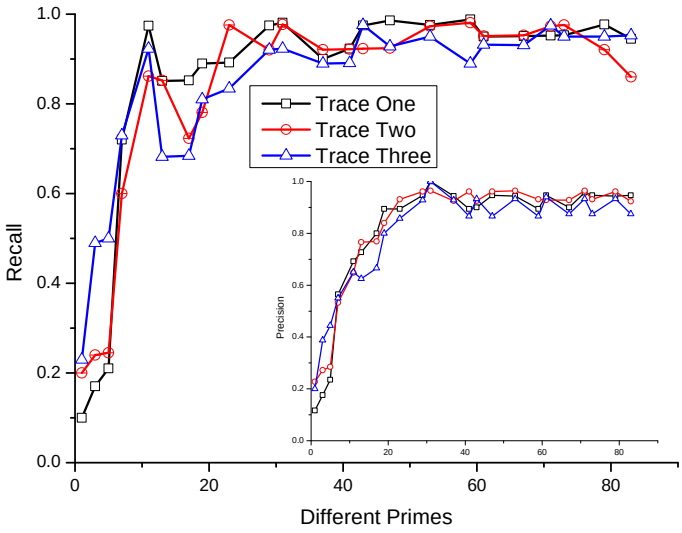


Fig. 9. Anomaly detection rate of each prime number

To ensure the symmetric degree sketch occupies the minimal memory, based on the results in Fig.9 and the selection methods in Section VIII.B, the hash function group selected is shown in Eq.(15).

$$\begin{cases} h_1(x) \equiv xmod29 \\ h_2(x) \equiv xmod31 \end{cases} \tag{15}$$

The experimental results based on the hash functions in Eq.(15) are shown in Table 4, in which the NAV denotes the Number of Anomaly Verified with manual efforts in the traces. From Table 4, it can be found that after modifying the core hash function group, the detection performance is significantly improved. All the recall and precision have reached to more than 90%. However, the precision of the trace 1 is lower than 50%, that of other traces are also lower than 70%. The results indicate that the hash functions still need to be adjusted to improve the performance.

TABLE 4
Algorithm results after modifying hash functions

Trace	NAV	TP	FP	FN	Precision	Recall
One	19	19	21	0	47.5%	100%
Two	28	26	18	2	59.1%	92.8%
Three	15	14	8	1	63.6%	93.3%

To further improve the detection efficiency and reduce the false detection rate, we noted that the cause of IP address collisions and its influence on the false detection results. It is noted that the lower precision is mainly caused by the collisions of IP segments between that of the normal IP addresses and the abnormal IP addresses. A simple example is shown in Fig.10. If in the raw datasets, there are two abnormal IP addresses and one normal IP address. Segments of the normal IP address conflicts with that of the abnormal IP_1 in hash table 1, while segments of the normal IP address also conflict with the abnormal IP_2 in the hash table 2. During the process of reverse solution, we obtain three different item combinations. And these two conflicted normal IP segments

are combined into an abnormal IP address, which result in a false detection. To solve this problem, we can increase the number of hash functions to reduce the probability of conflictions.

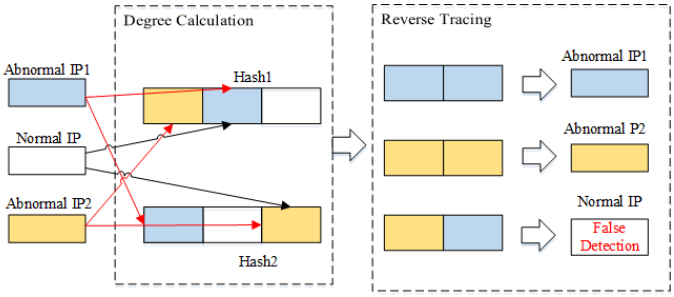


Fig. 10. Reason for false positives

In Table 5 we give the recall and precision with different number of hash functions and different set of primes. From TABLE 5 we can get that with the number of hash function increasing, the precision is greatly reduced, which means the confliction is reduced. When the H equals to 4 and m_j equals to 29, 31, 43, 47, respectively, the precision of all the traces are higher and that of recall are higher too. Based on those analysis results, the hash function group used is shown in Eq.(16).

TABLE 5
Detection results of different selection of hash functions

Trace	Hash and primes used	NAV	TP	FP	FN	Precision	Recall
One	2(29,31)	19	19	21	0	47.5%	100%
	3(29,31,43)	19	19	3	0	86.3%	100%
	4(29,31,43,47)	19	19	0	0	100%	100%
	5(29,31,43,47,53)	19	19	0	0	100%	100%
Two	2(29,31)	28	26	18	2	59.1%	92.8%
	3(29,31,43)	28	26	5	2	83.8%	92.8%
	4(29,31,43,47)	28	26	0	2	100%	92.8%
	5(29,31,43,47,53)	28	26	0	2	100%	92.8%
Three	2(29,31)	15	14	2	1	87.5%	93.3%
	3(29,31,43)	15	14	2	1	87.5%	93.3%
	4(29,31,43,47)	15	13	1	2	92.8%	86.7%
	5(29,31,43,47,53)	15	13	1	2	92.8%	86.7%

$$\begin{cases} h_1(x) \equiv x mod 19 \\ h_2(x) \equiv x mod 41 \\ h_3(x) \equiv x mod 43 \\ h_4(x) \equiv x mod 47 \end{cases} \quad (16)$$

D. Rescale the sliding time window

Threshold is another important parameter for obtaining better detection results. In this paper we selected the threshold based on the dynamic changing traffic patterns by employing sliding time window mechanism, which employ the patterns inside of the sliding time window in front of the time point being analyzed to generate the thresholds. Here we firstly adjust the size of the sliding time window, which is selected as 30 seconds based on the network management experience, thus the threshold of the current detection time point is computed based on the network traffic patterns in the past 30 seconds. Fig.11 shows the detection results of the developed algorithm with different sliding time window size.

As the Fig 11 shows, with the sliding window size increasing, the recall and precision decrease quickly. Those results verified that the users behaviors exist a kind of inertia, they won't change their behaviors sharply. Thus we can employ the thresholds extracted from historical data to perform anomaly detection. Additionally, the behavior characteristics close to the time point being analyzed have greater influence on the results, and that of the time points far from the analyzed time points may still have some influence. Thus we need select the suitable sliding time window size to achieve better results. From the experimental results shown in Fig 11, we can find the suitable sliding time window size is 30.

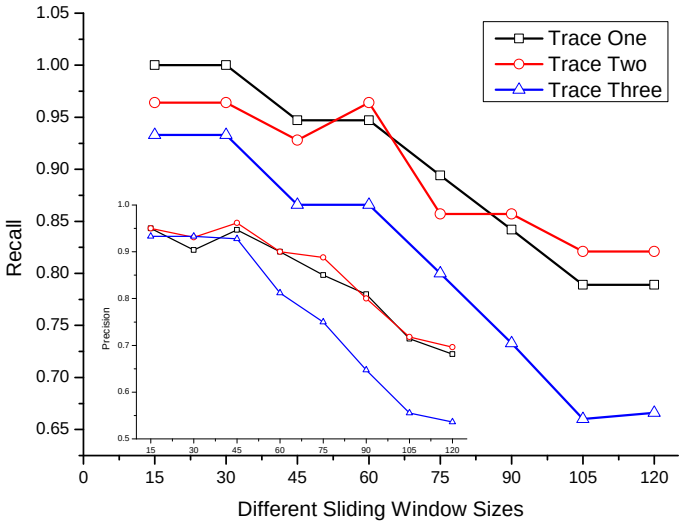


Fig. 11. Detection results with the sliding window size

E. Threshold adjustment

Another important parameter is the number of σ in Eq.(8), we use L to denote it. We selected the mean of the *SDI* in the sliding time window as the baseline, and the L variances are used to generated a certain upper and lower bounds to form the range used for anomaly detection. The experimental results with different L are shown in Table 6. As the TABLE shows, the precision increase decreases with the increase of the upper and lower range of baseline, and select L equal to 3 can obtain higher recall and precision.

TABLE 6
Detection results with different L

Trace	L	NAV	TP	FP	FN	Precision	Recall
One	2	19	19	8	0	70.3%	100%
	3	19	19	0	0	100%	100%
	4	19	19	0	0	100%	100%
Two	2	28	27	8	1	77.1%	96.4%
	3	28	26	0	2	100%	92.9%
	4	28	23	0	5	100%	92.9%
3	2	15	14	2	1	87.5%	93.3%
	3	15	13	1	2	92.8%	86.7%
	4	15	13	1	2	92.8%	86.7%

IX. Performance Evaluation

Based on the parameters selected in above Section, we evaluate proposed method with other related methods to verify its performance.

A. Compared with methods using other statistical features

We first select the methods based on the total number of flows [53] and the flow size distribution [54] to verify the efficiency of the *SDI* in detecting smart attacks. In [53], [54], the Netflow model is used to aggregate packets into flows.

In [53], the authors proposed EGADS contains three components, including the time series modeling module, the anomaly detection module and the alerting module. The anomaly detection methods are mainly based on time series methods. Their methods can detect the anomalies such as the outliers whose value is significantly different from the expected values. The change points whose valve is significantly different before and after time point t. In this paper we employ the EWMA (Exponentially Weighted Moving Average) methods to measure the significantly changes in the dynamic changing trends of the total number of flows. If we detect an abnormal change, we regard it as an anomaly. But the number of flows of all hosts are massive and the slight changes caused by anomalies usually cannot be found by the time series model, thus the recall and precision are lower.

In [54], the authors employ the entropy to measure the distribution of specific traffic feature and then perform abnormal detection. Here we employ the entropy to measure the distribution of flow size. We claim an anomaly is detected if we find an obvious changes in the distribution. But the attacks today trend to gradually change their behavior to avoid causing obvious changes, the recall and precision are lower.

The performance evaluation results are shown in TABLE 7. As the TABLE shows, methods based on those statistical features have very low recall and precision. This is because that there are thousands of flow records per second in the network today, there is no obvious changes in those patterns with some ongoing anomalies. Meanwhile, *SDI* is extracted according to the difference between the normal behaviors and anomalies, it is more efficient in identifying anomalies from the massive traffic patterns.

TABLE 7
Detection results with different features

Trace	Methods	NAV	TP	FP	FN	Precision	Recall
One	Proposed	19	19	0	0	100%	100%
	[53]	19	0	2	19	0%	0%
	[54]	19	2	4	17	33.3%	10.5%
Two	Proposed	28	26	3	2	89.6%	92.85%
	[53]	28	3	6	25	33.3%	10.7%
	[54]	28	8	9	22	47.05%	28.57%
Three	Proposed	15	14	2	1	87.5%	93.33%
	[53]	15	1	2	14	33.3%	13.3%
	[54]	15	3	4	12	42.8‘%	20%

B. Compared with methods using fixed thresholds

The traffic patterns are dynamic changing trends and have obvious routine characteristics. Generally speaking, different detection time points should use different thresholds. There are also many works using the fixed threshold for anomaly detection, we believe these methods using fixed thresholds will get higher recall and precision for short time period monitoring. Those methods will lose their efficiency for long time traffic monitoring. In this paper we employ the methods proposed in [55] to evaluate the performance of our methods.

We apply different fixed thresholds and the experimental results are shown in Fig.12. As the figure shows different traces should select different thresholds to generate better results. As analyzed in Section V.C, the traces are collected from different time periods. In different time periods, the users have different behavior characteristics and the selected thresholds should be different. If we use a fixed threshold, the recall and precision will be very low for other traces with different collection time points. The results verify that dynamic threshold selection mechanism is important for long time monitoring.

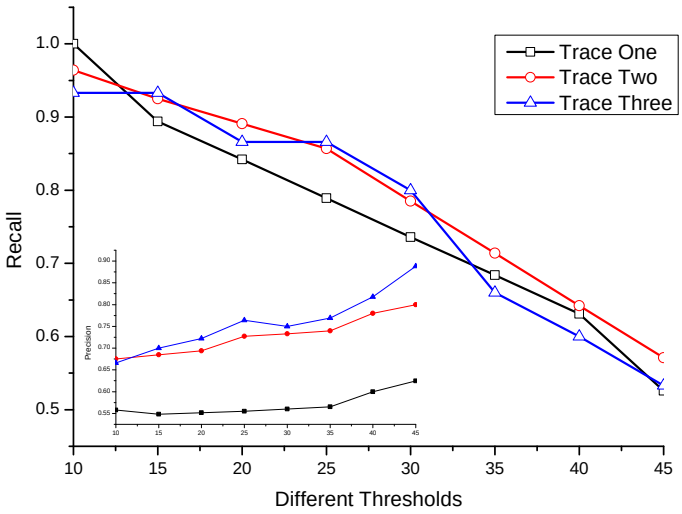


Fig. 12. Detection results with different threshold

C. Compared with sketch using whole IP mapping

The whole IP address can be selected as the key of hash functions in the sketch and perform abnormal detection [55], [56], [57]. Usually the IP address can be treated as a number with 255255255255 as the maximum value. To perform evaluation, we first selected suitable parameters for the methods based on whole IP mapping. Similar with the calculation process in Section VIII.B, we calculated and selected H equal to 4, and the prime numbers are 569, 577, 587, and 599 in Equation 16. TABLE 8 shows the experimental results of the methods based on whole IP addresses mapping. From TABLE 8, we can get that the recall is higher than the methods proposed, but the precision is lower. This is because the data obtained in the reverse process is a number instead of IP address. When the number is not big enough, we can obtain several IP addresses from the number and generate false detection results. For example, when the reversed number is 1151541228, we can obtain lots of legal IP addresses such as 115.154.12.28 or 115.154.1.228. Thus further analysis combined with the

network configuration is needed to improve the detection performance, which leads to higher time complexity.

TABLE 8
Detection results of whole IP mapping

Trace	NAV	TP	FP	FN	Precision	Recall
One	19	19	2	0	90.4%	100%
Two	28	26	3	2	89.6%	92.9%
Three	15	14	2	1	87.5%	93.3%

D. Evaluation on the computation complexity

We evaluate the computation complexity of our methods with several related methods, including the method using the fixed threshold [55], the methods using whole IP address as the key and tracing the anomaly related IP addresses by querying the table of all items and their corresponding keys [56], [57]. We selected the trace one as an example and the analysis results are shown in TABLE 9, which verified the efficiency and accuracy of our methods. The main differences among those methods and our method can be summarized as follows:

1) In [55], the threshold used is fixed and the tracing process is based on the method proposed in this paper. The results show that the recall and precision is lower than other methods, this is caused by the dynamic changing characteristic of the traffic patterns, and single fixed threshold is not suitable for long time monitoring.
2) In [56], select the whole IP address as the key and trace the anomaly based on the table queries, use the dynamic threshold selection proposed in this paper to select suitable thresholds. Traditionally the hash functions are not reversible, to trace the anomaly related IP addresses, we must store the IP addresses and their corresponding items in a table. After we detect the abnormal items from the sketch, we can query the table to get the anomaly related IP addresses. As the results show, the execution time is much longer than that of the method proposed, which is mainly caused by the time-consuming tracing process. Additionally, it costs more memory than our methods due to the key selection and the table storage.
3) In [57], it employs the IP segments as the key and dynamic threshold selection mechanism to select suitable thresholds, the tracing method is still based on the table queries. As the results, the memory cost can be reduced by selecting the IP segments as keys. The recall and precision is also increased by introducing the threshold dynamic selection mechanism. However, the tracing process is still quit time-consuming, which is not suitable for real time security monitoring in large scale network.

X. Conclusion and Future Work

Mining anomaly behavior characteristics and employ them to design the abnormal detection model is the basic way for smart anomaly detection. We find most of the attacks today generate lots of incomplete sessions, thus we propose the connection degree and symmetry degree to characterize the abnormal behaviors and use them to identify the anomalies from massive raw traffic packets. We design a symmetry degree sketch to calculate the symmetry degree quickly. To reduce the memory cost and collision probability, we select the IP segments as the keys. To make the sketch reversible and trace the anomaly-related IP addresses efficiently, we employ the Chinese remainder theorem to design the hash functions. To capture the dynamic changing trends of traffic patterns, we select the thresholds using sliding time windows mechanism. By combining the symmetry degree with the reversible sketch, we can greatly reduce the computational complexity of security monitoring while increasing the detection accuracy in high speed network. We verified the efficiency of our methods through experiments using actual network traffic data traces collected from the northwest center of CERNET. However, it cannot detect the abnormal behaviors with both req-ack packets. The *SDI* of those anomalies are similar with that of the normal behaviors. We can detect those anomalies by combining with other methods, such as mining the co-occurrence or periodic behaviors from a long time period traffic monitoring. For future work, we will focus on developing new framework based on machine learning and mining those co-occurrence behaviors for traffic security monitoring.

TABLE 9
Performance comparison of anomaly detection algorithms

Methods	Time(S)	Memory(MB)	Precision	Recall
Proposed	512	45.5	100%	100%
[55]	452	42.6	61.5%	85.5%
[56]	2594	249.4	81.8%	95.4%
[57]	2715	51.7	100%	100%

Acknowledgment

We would like give our grateful thanks to Yanyu Liu from HUAWEI Technology Co., Ltd. for her efforts in data labelling and Guodong Li from the campus network center of Xi'an Jiaotong University for his efforts in traffic trace collection

References

[1] N. Virvilis and D. Gritzalis, "The big four-what we did wrong in advanced persistent threat detection?," in *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*, pp. 248–254, IEEE, 2013.
[2] C. Tankard, "Advanced persistent threats and how to monitor and deter them," *Network security*, vol. 2011, no. 8, pp. 16–19, 2011.
[3] J. Liu, Y. Xiao, K. Ghaboosi, H. Deng, and J. Zhang, "Botnet: classification, attacks, detection, tracing, and preventive measures," *EURASIP journal on wireless communications and networking*, vol. 2009, no. 1, p. 692654, 2009.
[4] R. Masood, Z. Anwar, *et al.*, "Swam: stuxnet worm analysis in metasploit," in *Frontiers of Information Technology (FIT), 2011*, pp. 142–147, IEEE, 2011.
[5] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pp. 71–82, ACM, 2002.
[6] M. V. Mahoney, "Network traffic anomaly detection based on packet bytes," in *Proceedings of the 2003 ACM symposium on Applied computing*, pp. 346–350, ACM, 2003.
[7] G. Giorgi and C. Narduzzi, "Detection of anomalous behaviors in networks from traffic measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 12, pp. 2782–2791, 2008.

[8] S. S. Kim and A. N. Reddy, "A study of analyzing network traffic as images in real-time," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, pp. 2056–2067, IEEE, 2005.

[9] W. John and S. Tafvelin, "Analysis of internet backbone traffic and header anomalies observed," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pp. 111–116, ACM, 2007.

[10] A. N. Mahmood, C. Leckie, J. Hu, Z. Tari, and M. Atiquzzaman, "Network traffic analysis and scada security," in *Handbook of Information and Communication Security*, pp. 383–405, Springer, 2010.

[11] Z. Zheng and A. N. Reddy, "Safeguarding building automation networks: The-driven anomaly detector based on traffic analysis," in *Computer Communication and Networks (ICCCN), 2017 26th International Conference on*, pp. 1–11, IEEE, 2017.

[12] W. Aiello, C. Kalmanek, P. McDaniel, S. Sen, O. Spatscheck, and J. Van der Merwe, "Analysis of communities of interest in data networks," in *International Workshop on Passive and Active Network Measurement*, pp. 83–96, Springer, 2005.

[13] Z. Liu, X. Guan, S. Li, T. Qin, and C. He, "Behavior rhythm: A new model for behavior visualization and its application in system security management," *IEEE Access*, vol. 6, pp. 73940–73951, 2018.

[14] X. He, G. Liu, D. Wang, Y. Liu, and X. Wang, "Network anomaly behavior detection method based on out-degree and in-degree of host." https://patents.google.com/patent/CN104135474A/en/, 2014.

[15] T. Qin, X. Guan, W. Li, P. Wang, and M. Zhu, "A new connection degree calculation and measurement method for large scale network monitoring," *Journal of Network and Computer Applications*, vol. 41, pp. 15–26, 2014.

[16] N. G.-Q. L. Jia-Zhen and P. Z.-S. M. Zhi-Min, "Verification based on keystroke biologic characteristics using support vector data description," *Pattern Recognition and Artificial Intelligence*, vol. 5, 2008.

[17] Z. Cai, C. Shen, and X. Guan, "Mitigating behavioral variability for mouse dynamics: A dimensionality-reduction-based approach," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 2, pp. 244–255, 2014.

[18] M. Anandapriya and B. Lakshmanan, "Anomaly based host intrusion detection system using semantic based system call patterns," in *Intelligent systems and control (ISCO), 2015 IEEE 9th international conference on*, pp. 1–4, IEEE, 2015.

[19] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[20] M. Mardani and G. B. Giannakis, "Estimating traffic and anomaly maps via network tomography," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1533–1547, 2016.

[21] H. Kasai, W. Kellerer, and M. Kleinsteuber, "Network volume anomaly detection and identification in large-scale networks based on online time-structured traffic tensor tracking," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 636–650, 2016.

[22] I. Nevat, D. M. Divakaran, S. G. Nagarajan, P. Zhang, L. Su, L. L. Ko, and V. L. Thing, "Anomaly detection and attribution in networks with temporally correlated traffic," *IEEE/ACM Transactions on Networking (TON)*, vol. 26, no. 1, pp. 131–144, 2018.

[23] J. Wang and I. C. Paschalidis, "Statistical traffic anomaly detection in time-varying communication networks," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 2, pp. 100–111, 2015.

[24] J. Zhang and I. C. Paschalidis, "Statistical anomaly detection via composite hypothesis testing for markov models," *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 589–602, 2017.

[25] CISCO NetFlow: http://www.cisco.com/en/US/products/ps6601/products_white_paper09186a00800a3db9.shtml.

[26] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *ACM SIGCOMM Computer Communication Review*, vol. 34, pp. 219–230, ACM, 2004.

[27] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *ACM SIGCOMM Computer Communication Review*, vol. 35, pp. 217–228, ACM, 2005.

[28] Y. Zhang, "An adaptive flow counting method for anomaly detection in sdn," in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, pp. 25–30, ACM, 2013.

[29] N. Duffield, C. Lund, and M. Thorup, "Estimating flow distributions from sampled flow statistics," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 325–336, ACM, 2003.

[30] N. Duffield, C. Lund, and M. Thorup, "Properties and prediction of flow statistics from sampled packet streams," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pp. 159–171, ACM, 2002.

[31] N. Duffield, C. Lund, and M. Thorup, "Flow sampling under hard resource constraints," *ACM SIGMETRICS Performance Evaluation Review*, vol. 32, no. 1, pp. 85–96, 2004.

[32] D. Brauckhoff, B. Tellenbach, A. Wagner, M. May, and A. Lakhina, "Impact of packet sampling on anomaly detection metrics," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pp. 159–164, ACM, 2006.

[33] J. Mai, C.-N. Chuah, A. Sridharan, T. Ye, and H. Zang, "Is sampled data sufficient for anomaly detection?," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pp. 165–176, ACM, 2006.

[34] A. Lakhina, M. Crovella, and C. Diot, "Characterization of network-wide anomalies in traffic flows," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pp. 201–206, ACM, 2004.

[35] X. Guan, T. Qin, W. Li, and P. Wang, "Dynamic feature analysis and measurement for large-scale network traffic monitoring," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 905–919, 2010.

[36] R. Schweller, Z. Li, Y. Chen, Y. Gao, A. Gupta, Y. Zhang, P. A. Dinda, M.-Y. Kao, and G. Memik, "Reversible sketches: enabling monitoring and analysis over high-speed data streams," *IEEE/ACM Transactions on Networking (ToN)*, vol. 15, no. 5, pp. 1059–1072, 2007.

[37] G. Cormode and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications," *Journal of Algorithms*, vol. 55, no. 1, pp. 58–75, 2005.

[38] P. Flajolet and G. N. Martin, "Probabilistic counting algorithms for data base applications," *Journal of computer and system sciences*, vol. 31, no. 2, pp. 182–209, 1985.

[39] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, "Sketch-based change detection: methods, evaluation, and applications," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pp. 234–247, ACM, 2003.

[40] T. Wellem, Y.-K. Lai, C.-Y. Huang, and W.-Y. Chung, "A hardware-accelerated infrastructure for flexible sketch-based network traffic monitoring," in *High Performance Switching and Routing (HPSR), 2016 IEEE 17th International Conference on*, pp. 162–167, IEEE, 2016.

[41] D. Tong and V. K. Prasanna, "Sketch acceleration on fpga and its applications in network anomaly detection," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 4, pp. 929–942, 2018.

[42] C. Callegari, S. Giordano, and M. Pagano, "On the combined use of sketches and cusum for anomaly detection," in *Computing and Network Communications (CoCoNet), 2015 International Conference on*, pp. 157–162, IEEE, 2015.

[43] Q. Huang and P. P. Lee, "Ld-sketch: A distributed sketching design for accurate and scalable anomaly detection in network data streams," in *INFOCOM, 2014 Proceedings IEEE*, pp. 1420–1428, IEEE, 2014.

[44] C. Wang, T. T. Miu, X. Luo, and J. Wang, "Skyshield: A sketch-based defense system against application layer ddos attacks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 559–573, 2018.

[45] H. Zhang, D. D. Yao, and N. Ramakrishnan, "Detection of stealthy malware activities with traffic causality and scalable triggering relation discovery," in *Proceedings of the 9th ACM symposium on Information, computer and communications security*, pp. 39–50, ACM, 2014.

[46] H. Zhang, D. D. Yao, and N. Ramakrishnan, "Causality-based sensemaking of network traffic for android application security," in *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*, pp. 47–58, ACM, 2016.

[47] P. Wang, P. Jia, J. Tao, and X. Guan, "Mining long-term stealthy user behaviors on high speed links," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 2051–2059, IEEE, 2018.

[48] D. Yao, X. Shu, L. Cheng, and S. Stolfo, *Anomaly Detection as a Service: Challenges, Advances, and Opportunities. In Information Security, Privacy, and Trust Series. Morgan & Claypool.* San Rafael, California: Morgan Claypool Publishers, 2017.

[49] T. Qin, L. Wang, Z. Liu, and X. Guan, "Robust application identification methods for p2p and voip traffic classification in backbone networks," *Knowledge-Based Systems*, vol. 82, pp. 152–162, 2015.

[50] K. Keys, D. Moore, R. Koga, E. Lagache, M. Tesch, and K. Claffy, "The architecture of coralreef: an internet traffic monitoring software suite," in *PAM2001, Workshop on Passive and Active Measurements, RIPE*, Citeseer, 2001.

[51] Y. Meng, T. Qin, Y. Liu, and C. He, "An effective high threating alarm mining method for cloud security management," *IEEE Access*, vol. 6, pp. 22634–22644, 2018.

[52] C. C. Zou, W. Gong, and D. Towsley, "Worm propagation modeling and analysis under dynamic quarantine defense," in *ACM Workshop on Rapid Malcode*, pp. 51–60, 2003.
[53] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1939–1947, ACM, 2015.
[54] P. Bereziński, B. Jasiul, and M. Szpyrka, "An entropy-based network anomaly detection method," *Entropy*, vol. 17, no. 4, pp. 2367–2408, 2015.
[55] X. Guan, P. Wang, and T. Qin, "A new data streaming method for locating hosts with large connection degree," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pp. 1–6, IEEE, 2009.
[56] N. LUO, A.-P. LI, Q.-Y. WU, and H.-B. LU, "Sketch-based anomalies detection with ip address traceability [j]," *Journal of Software*, vol. 10, p. 027, 2009.
[57] L. Ling, B. Yin, C. Jie, and D. O. Automation, "Anomaly analysis and identification of backbone network based on sketch and regularity distribution," *Journal of Systems Science and Mathematical Sciences*, 2015.

TAO QIN received the B.S. degree in information engineering and the Ph.D. degree in computer science and technology from Xian Jiaotong University, Xian, China, in 2004 and 2010 respectively. He is currently an Associate Professor with the Department of Computer Science and Technology and MOE KLINNS Lab, Xian Jiaotong University. Now he is a visiting scholar at the department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, USA, 01002. His research focuses on internet traffic analysis, traffic modeling, anomaly detection and online social network analysis.

Zhaoli Liu received the B.S. degree and M.S. degree in computer science and technology from Xian Jiaotong University, Xian, China, in 2007 and 2010 respectively. She is currently pursuing the Ph.D. degree in computer science and technology at Xian Jiaotong University, and she is now a visiting PhD student in Prof. Weibo Gongs group at University of Massachusetts, Amherst, USA, which is funded by China Scholarship Council. Her research interests lie primarily in network security and online social network analysis..

Pinghui Wang received the B.S. degree in information engineering and the Ph.D. degree in automatic control from Xian Jiaotong University, Xian, China, in 2006 and 2012 respectively. From 2012 to 2013, he was a Postdoctoral Fellow in the School of Computer Science at McGill University. From 2014 to 2015, he was a researcher at HUAWEI Noahs Ark Lab, Hong Kong. He is currently an associate professor in MOE KLINNS at Xian Jiaotong University. His research interests include Internet traffic measurement and modeling, abnormal detection, and online social network measurement.

Shancang Li received the B.Sc. and M.Sc. degrees in mechanics engineering and the Ph.D. degree in computer science from Xian Jiaotong University, Xian, China, in 2001, 2004, and 2008, respectively. He is currently a Senior Lecturer with the Department of Computer Science and Creative Technologies, University of the West of England, Bristol, U.K. His current research interests include digital forensics for emerging technologies, network security, cybercrimes, network attacks, wireless sensor networks, Internet of Things, and the lightweight cryptography in resource constrained devices

Xiaohong Guan received his B.S. and M.S. degrees in automatic control from Tsinghua University, Beijing, China, in 1982 and 1985, respectively, and his Ph.D. degree in electrical engineering from the University of Connecticut, Storrs, in 1993. From 1985 to 1988, he was with the Systems Engineering Institute, Xian Jiaotong University, Xian, China. From 1993 to 1995 he was a senior consulting engineer at PG&E. From January 1999 to February 2000, where he is also currently a Cheung Kong Professor of systems engineering and the Dean of School of Electronic and Information Engineering. He is also with with the Shenzhen Research School, Xian Jiaotong University Shenzhen Guangdong, China. His research interests include allocation and scheduling of complex networked resources, network security, and sensor networks.

Lixin Gao received the PhD degree in computer science from the University of Massachusetts at Amherst, in 1996. She is a professor of electrical and computer engineering with the University of Massachusetts at Amherst. Her research interests include social networks, and Internet routing, network virtualization, and cloud computing. Between May 1999 and January 2000, she was a visiting researcher at AT&T Research Labs and DIMACS. She was an Alfred P. Sloan fellow between 2003-2005 and received an NSF CAREER Award in 1999. She won the best paper award from IEEE INFOCOM 2010, and the test-of-time award in ACM SIGMETRICS 2010. Her paper in ACM Cloud Computing 2011 was honored with Paper of Distinction. She received the Chancellor's Award for Outstanding Accomplishment in Research and Creative Activity in 2010. She is a fellow of the ACM and the IEEE.