Notice of Retraction

After careful consideration, this paper has been retracted due to an authorship issue.
Reasonable effort should be made to avoid citing this paper.

We regret any inconvenience.

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 05/10 pp561–573
DOI: 10.26599/TST.2018.9010021
Volume 23, Number 5, October 2018

Efficient Feature Extraction Using Apache Spark for Network Behavior Anomaly Detection

Xiaoming Ye, Xingshu Chen*, Dunhu Liu, Wenxian Wang, Li Yang, Gang Liang, and Guolin Shao

Abstract: Extracting and analyzing network traffic feature is fundamental in the design and implementation of network behavior anomaly detection methods. The traditional network traffic feature method focuses on the statistical features of traffic volume. However, this approach is not sufficient to reflect the communication pattern features. A different approach is required to detect anomalous behaviors that do not exhibit traffic volume changes, such as low-intensity anomalous behaviors caused by Denial of Service/Distributed Denial of Service (DoS/DDoS) attacks, Internet worms and scanning, and BotNets. We propose an efficient traffic feature extraction architecture based on our proposed approach, which combines the benefit of traffic volume features and network communication pattern features. This method can detect low-intensity anomalous network behaviors and conventional traffic volume anomalies. We implemented our approach on Spark Streaming and validated our feature set using labelled real-world dataset collected from the Sichuan University campus network. Our results demonstrate that the traffic feature extraction approach is efficient in detecting both traffic variations and communication structure changes. Based on our evaluation of the MIT-DRAPA dataset, the same detection approach utilizes traffic volume features with detection precision of 82.3% and communication pattern features with detection precision of 89.9%. Our proposed feature set improves precision by 94%.

Key words: feature extraction; graph theory; network behavior; anomaly detection; Apache Spark

1 Introduction

Over the last decade, people have become increasingly reliant on computer networks in their work and daily life. As a result, network traffic has experienced tremendous growth in both volume and variety. Many studies have used the technique of network traffic analysis to improve the performance of networks, optimize network structures, and strengthen network security. Anomalous network behaviors are the common representation of network faults, cyber-attacks or other abnormal events within a network. Thus, discovering such anomalous network behaviors has become an important problem for researchers in this field. Today, efficient network behavior anomaly detection faces certain challenges due to the large complex nature of network traffic.

• Xiaoming Ye is with the School of Cybersecurity, Chengdu University of Information Technology, Chengdu 610225, and the College of Computer Science, Sichuan University, Chengdu 610065, China. E-mail: yexm.edu@gmail.com.
• Xingshu Chen, Wenxian Wang, and Gang Liang are with the College of Cybersecurity, Sichuan University, Chengdu 610065, China. E-mail: chenxsh@scu.edu.cn; catean@scu.edu.cn; lianggang@scu.edu.cn.
• Dunhu Liu and Li Yang are with the School of Management, Chengdu University of Information Technology, Chengdu 610103, China. E-mail: 264885613@qq.com; edta_edu@126.com.
• Guolin Shao is with the College of Compute Science, Sichuan University, Chengdu 610065, China. E-mail: Shaoguolin716@qq.com.
* To whom correspondence should be addressed.
Manuscript received: 2017-09-24; accepted: 2017-09-29

The goal of network behavior anomaly detection is to find unexpected or irregular behaviors in the data. Many studies have investigated network behavior anomaly detection in the traffic volume time-series data. These studies focus on the features of traffic volume, which are treated as a time series, such as Internet Protocol (IP) address counts, open port counts, durations, flow counts, traffic volume, packet counts, and so on. Using the features of traffic volume statistics for anomaly detection, however, cannot efficiently detect low-intensity anomalous behaviors. Meanwhile, other researchers focus on the features of communication patterns to detect anomalous behaviors that do not exhibit traffic volume change, such as degree, in-degree, out-degree, and graph edit distance, to name a few. These studies can extract the network communication structure feature from traffic and discover low-intensity anomalous network behaviors, such as botnet command and control communications, which cannot be detected with conventional traffic volume feature anomaly detection.

Different network anomalies show the varying traffic features' changes[1]. Both traffic volume features or communication pattern features can be used to detect attacks, but they cannot comprehensively detect the overall network behavior. Hence, in the current study, we explore the benefit of combining the traffic volume feature and the communication pattern feature obtained by network traffic data. We propose a novel dynamic metrics approach for traffic feature extraction, which serves as a profiling network behavior for our anomaly detection.

1.1 Related works

Feature extraction from network traffic, which is fundamental to the implementation of network behavior anomaly detection, has received significant research attention. We summarise the related work and pose several limitations.

The deep packet inspection technique is used to analyze network traffic data. Given that network traffic volume has increased and the network speed has improved. The packet sampling technology and network flows (e.g., NetFlow and NetStream) have been proposed. Many researchers have adopted flow characteristics to find malicious behaviors in detection algorithms. In Ref. [2], the authors proposed a Hidden Markov model to realize the attack detection of Secure Shell brute force based on time series. In Ref. [3], utilizing the Gaussian mixture distribution model, the authors profiled the baseline of normal network behavior. The work of Ref. [4] proposed the Autoregressive Fractionally Integrated Moving Average (ARFIMA) model, which compared forecast values with real values, and identified the deviation degree to determine whether the current value is abnormal or normal. The previous research in Refs. [5, 6] showed that the observed traffic had self-similar and long-range dependent characteristics over time[7, 8]. Meanwhile, the traffic analysis approaches based on the time-scale can identify abrupt changes in network traffic volume[9].

Despite recent network traffic analyzes in anomalous behavior detection research, past studies have only focused on traffic volume. In practice, the detection approaches based on the packet feature extraction are extremely difficult to apply in a high-speed network environment[10]. The authors in Ref. [10] also pointed out that the flow-level feature cannot effectively reflect the whole information of the network traffic, characterized by information loss with respect to the network traffic characteristics. Particularly, attackers can evade the traffic volume detection approach, and low-intensity anomalous behaviors do not manifest in terms of traffic volume but change in terms of communication patterns.

Subsequently, several researchers presented graph theory to analyze network communication patterns and determine network attacks by identifying the anomalous graph structure dealing with these issues[11–16], including the analysis of network structure, dependence, and correlation. The authors in Ref. [17] used the network flow construct traffic graph to analyze the network communication pattern and employed the characteristic values of authority and hub in the cluster algorithm for the botnet detection. The authors in Ref. [18] proposed a graph-based analysis of the network communication pattern to identify malicious network sources. In Ref. [19], the authors proposed the use of the community discovery method to identify the anomalous community, and found that members of the community were under similar attacks with low intensity against multiple hosts. In Ref. [20], the authors described the network communication pattern based on time series using a graph model to achieve anomaly detection. The authors first defined the process of calculating the similarity of the two graphs and detected the anomaly graph to identify low-intensity attacks. The analysis of network data

based on the graph theory has received wide attention from researchers.

Despite these breakthroughs, existing research on communication pattern has focused on specific problems or aspects, such as detecting BotNets, identifying Peer-to-Peer (P2P) applications, and generating malware signatures, which deal with the problem of revealing anomalous communication structures based on fewer graph features instead of traffic volume anomaly. In contrast, we aim to provide an efficient feature extraction approach, which is built upon real-time network traffic data, to combine the benefit of traffic volume features and network communication pattern features for real-world network behavior anomaly detection. In our work, a novel dynamic metrics approach for traffic feature extraction is proposed to describe the change of network traffic situation across the two adjacent time windows. Our experiment results demonstrate that the proposed approach does not only increase and complement flow-level feature data, but also improve detection accuracy.

1.2 Key contributions

The paper proposes a fast and efficient feature extraction approach using Apache Spark for network behavior anomaly detection. The main contributions of our approach are presented below.

(1) We propose a real-time feature extraction architecture to profile traffic volume and the communication pattern simultaneously, instead of just one of them. We define four types of flow-level feature set to profile the network behavior changes across time windows, including traffic volume static feature, traffic volume dynamic feature, communication pattern static feature, and communication pattern dynamic feature.

(2) We make several key attempts to explore the regular feature fluctuations and the correlations between the proposed four types of feature set in anomalous time windows.

(3) We establish a real-time anomaly detection approach called Evidence Accumulation Deviation Degree (EADD), which provides the following advantages: (i) quantification of the absolute change to discover burstiness in network behaviors, (ii) quantification of the relative changes to reduce false alarms due to the timely occurrence of centralised, periodical network behaviors, and (iii) the quantification of the changes in trends, to discover low-frequency and low-intensity attacks even though the features of network behavior do not change suddenly.

(4) We demonstrate the applications of combining traffic volume and the communication pattern feature set in detecting conventional traffic volume and low-intensity anomalous behaviors using a real-world dataset and validation dataset. Based on the real-world labelled dataset traffic volume and the communication pattern feature, we make the different contributions for detected instances. Based on our evaluation of the MIT-DRAPA dataset, the same detection approach utilizes traffic volume and communication pattern features with detection precision rates of 82.3% and 89.9%, respectively. Our proposed feature set also showed improved precision by 94%. We also compare the changes of the time series of the traffic volume, which shows no sudden change. We visualize four graphs to profile the changes in communication patterns, successively normal graph structures before attacks, anomalous graph structures, and normal graph structures after attacks. We also compare with feature set extraction works in Ref. [21] depending on whether the same test dataset is used. Our detection results demonstrate that our extraction feature set has a high detection rate.

2 Feature Extraction Architecture

In this section, a real-time feature extraction architecture is proposed to profile the traffic volume and the communication pattern. The system consists of four key steps, as shown in Fig. 1.

Step 1 is the procedure of stream data collection. Observed packets are collected to form the session flow records in real time. At the same time, session flow records are transformed into a graph using Spark Streaming for constructing the graph model.

Step 2 calculates the statistical features. The system establishes profiles for each time interval by means of the feature extraction. Then, the static metrics and dynamic metrics approaches are applied to calculate the features set. Once the behavior profiles of current time are established, the system switches to the detection step.

Step 3 is a time series anomaly detection approach. Our proposed approach analyzes time series by a single data point and the subsequent time series. More specifically, neither the entire history data nor only one

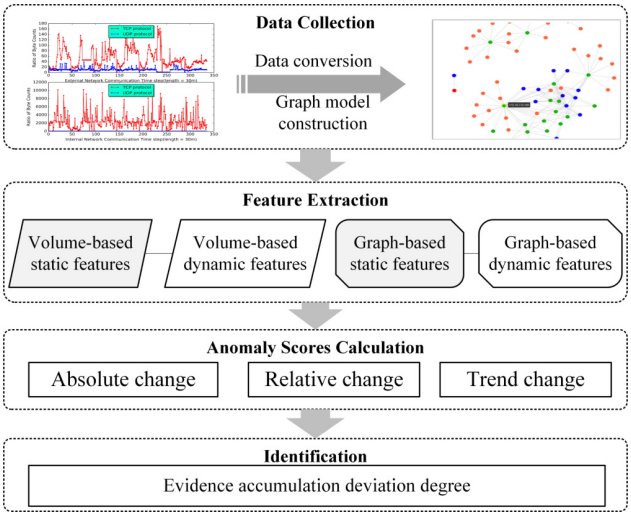


Fig. 1 Feature extraction architecture.

nearest data is considered for the comparison of the feature values. We also determine how many horizontal adjacent points and vertical adjacent points are needed to work well. This is planned for future works.

Step 4 is an anomaly decision phase. Based on data from the last step of the analysis, evidence of accumulation deviation degree (i.e., anomaly scores) is calculated, and alert thresholds are derived from years of manual analysis experiences in the campus observation networks.

3 Feature Extraction

In this section, we propose our feature set across each time window; here, we combine the traffic volume features and communication structure features. In the meantime, we cover the shortage of the lack of information on flow-level features. The definition of flow-level traffic data (i.e., session flow) is the same with our previous work[22], namely, session flow data. Two measurement methods are directly related to our work: static metrics and dynamic metrics.

We utilize the static metrics method to extract the static feature, which we calculate on one dataset of the given time interval. The static feature can describe network traffic condition in the current time window. It is a conventional feature extracted method applied on time series data.

We utilize the proposed dynamic metrics method to extract the dynamic feature and then calculate the two datasets related to two adjacent time intervals. The dynamic feature can describe the change of network traffic situation across the two adjacent time windows.

Table 1 List of network traffic features.

Feature category	Detail
Traffic volume static feature	Byte counts
	Packet counts
	IP connection counts
	Port connection counts
Traffic volume dynamic feature	Ratio of protocol
	Ratio of connection counts
	Ratio of byte counts
	Ratio of packet counts
Communication pattern static feature	Degree
	In-degree
	Out-degree
	Entropy of the degree
	In-degree
	Out-degree maximum degree
	Only in-degree node counts
	Only out-degree node counts
	Both in-degree
	Out-degree node counts
Communication pattern dynamic feature	Graph edit distance
	Overlay edge counts
	Overlay node counts

The key features are presented in Table 1.

The traffic volume feature set is inspired by previous research[23, 24]. The mean, entropy, and maximum and standard deviation values of the features are considered. The mean packet counts per time window reflect the mice flow proportion of all flows, wherein more mice flows mean a larger attack. The communication pattern feature set is also the graph feature set used in related works[25–27]. In addition, graph edit distance is suitable for measuring changes in graph topology[28]. Graph edit distance can be used to evaluate the changes between two successive graphs, which can reflect the insertion and deletion of the edges and vertices. Graph patterns can also be used to identify complex behavior relationships.

3.1 Traffic volume dynamic feature

In this section, we explore the proposed dynamic metrics methods to calculate the traffic volume dynamic features. The first step is obtaining the two data sets, namely, the continuous connection dataset and active connection dataset. The definitions are presented here. The second step is to calculate the dynamic features of these datasets.

For example, the dynamic feature value of packet counts is the ratio of the number of packet from continuous connection dataset to the number of packet

from the active connection dataset.

The Continuous Connection Set (CCS) is across the two adjacent time windows. CCS contains all records of the history window dataset, $\mathrm{CS_{WH}}$, which also belongs to the observation window dataset, $\mathrm{CS_{WA}}$, defined below.

$$\mathrm{CCS} = \mathrm{CS_{WA}} \cap \mathrm{CS_{WH}} \tag{1}$$

Meanwhile, the Active Connection Set (ACS) is also across the two adjacent time windows. The elements of ACS are in observation window dataset, $\mathrm{CS_{WH}}$, but not in the history window dataset, $\mathrm{CS_{WA}}$, which is defined below.

$$\mathrm{ACS} = \mathrm{CS_{WA}} - \mathrm{CS_{WH}} \tag{2}$$

In the equation above, WH denotes the history window (or the nearest-neighbour time window), $\mathrm{CS_{WH}}$ denotes the dataset of WH, WA denotes the current time window (or the observation window), and $\mathrm{CS_{WA}}$ denotes the dataset of WA.

3.2 Communication pattern dynamic feature

In this work, session flow data are represented as graphs. The graph nodes represent the source or destination IP addresses, and graph edges represent the network connection between nodes. We extract the graph-based structural features to profile the communication patterns for the network traffic analysis. The critical steps involved in the extraction of communication pattern features are described below.

Step 1: The session flow data are converted into the graph data to construct a directed graph $G = (V, E)$ in each time window, where V denotes the nodes in the graph and IP is the address set of the session flow data. The $(u, v) \in E$ denotes the edges in the graph, indicating that the network communication behavior exists between the IP address u and IP address v .

Step 2: We apply the static metrics definition and dataset, $\mathrm{CS_{WA}}$ to calculate the graph-based static features. This is shown in Fig. 2, where $T = \{t-n, \ldots, t-m, \ldots, t-1, t\}$ denotes a set of time intervals, $G(t)$ denotes a graph snapshot in time t, and a series of graph snapshot forms a time series graph given below.

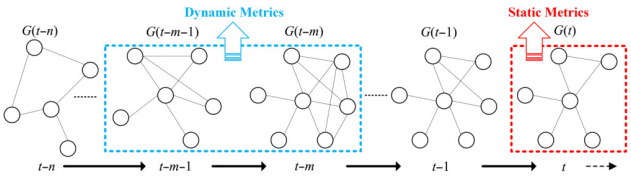


Fig. 2 Time series of graphs.

$G = \{G(t-n), \ldots, G(t-m), \ldots, G(t-1), G(t)\}$.

Step 3: We apply the dynamic metrics definition to calculate the features set of the dynamic graph, for example, the dataset related to the two graphs snapshot, namely, $G(t-m-1)$ and $G(t-m)$, over two adjacent time intervals at time $t-m-1$ and $t-m$. For example, the dynamic feature value of graph edit distance is calculated by $|V_{t-1}| + |V_t| + 2|V_{t-1} \cap V_t| + |E_{t-1}| + |E_t| + 2|E_{t-1} \cap E_t|$, where $|V_{t-1}|$ represents the number of nodes in graph $G(t-1)$, $|E_t|$ represents the number of edges in graph $G(t)$, $G(t-1)$ and $G(t)$ represent the graphs of the two adjacent time intervals.

4 Computation of Anomaly Scores

According to the long-term, we observe the regular feature fluctuations and the correlations among them from network traffic data on campus data centers. Then, we establish the anomaly detection approach, namely, the evidence accumulation deviation degree, which consists of the absolute change, the relative change, and the trend change. Next, we perform the following: (1) we quantify the absolute change based on our proposed method to discover the abrupt change, (2) we quantify the relative change to reduce false alarms due to the centralised, periodical network behaviors that occur, and (3) we quantify the change of the trend change to determine whether the network behavior will not change suddenly. The following parts introduce the process of calculating three change values.

4.1 Absolute change

The comparative quantification of the absolute change aims to solve the problem of burstiness network behaviors, which focus on the change of feature cross adjacent times.

The value of the absolute change is given by the formula $|f^i(t) - f^i(t')|$, where f^i denotes the i-th feature, t is current time and t' is the adjacent time. In each time interval, we compute $\mathrm{EA}^{(1)}$. As shown in Eq. (3), $\mathrm{EA}^{(1)}$ represents the value of evidence accumulative for the absolute change, w_i represents the weight of the i-th feature, m represents the number of features (and is equal to 1) and $\mathbb{R}$ represents the feature set with reference to the horizontal adjacent points at time t.

$$\mathrm{EA}^{(1)}(t) = \sum_{i=1}^{m} \left(w_i \cdot \max\{|\ f^i(t) - f^i(x)\ |, x \in \mathbb{R}\} \right) \tag{3}$$

4.2 Relative change

In this section, we define the evidence accumulation absolute change. Relative change aims to describe the network behavior with the property of periodicity and regularity in the time series data. This approach tries to solve the problem wherein centralised, periodical network behaviors appear on time can lead to more false alarms.

The value of the evidence accumulation relative change is given by the formula $f_i^j(t)/\max(f^i(t), f^i(t-1), \ldots, f^i(t-N_2))$, where N_2 denotes the number of data points that are vertically adjacent to time t, and the other variables have the same meaning as those described in Eq. (3). In each time interval, we compute $\mathrm{EA}^{(2)}$, which is given by Eq. (4), where $\mathrm{EA}^{(2)}$ represents the value of evidence accumulative relative change, $\mathbb{Z}$ denotes the feature set of the data points that are vertically adjacent to time t with reference to previous k weeks at the time t and $|\mathbb{Z}|$ denotes the size of the set $\mathbb{Z}$. $\mathbb{Z}$ is given by the formula $|\mathbb{Z}| = N_2 \times (k+1)$. We define the evidence accumulation relative change measure below.

$$\mathrm{EA}^{(2)}(t) = \sum_{i=1}^{m}\left(\frac{w_i}{k}\sum_{j=1}^{k}\left(\frac{f_i^j(t)}{\max\{f_i^j(x)\}}, x \in \mathbb{Z}\right)\right) \tag{4}$$

4.3 Trend change

The majority of past studies have focused on the individual unusual changes from the time series data, like our absolute change and relative change. We also consider the shapes of several data points of the time series with respect to the other extracted data points. By quantifying the fluctuations in trend changes, we can profile the sharp feature values on specific time series. When low-frequency and low-intensity attacks occur in a network environment, the features of network behavior will not change suddenly.

In each time interval, we calculate $\mathrm{EA}^{(3)}$, $\mathrm{EA}^{(3)}$ is given by Eq. (5). We also apply Symbolic Aggregate Approximation (SAX) to calculate degree of deviation between the subsequences. SAX[29] is proposed based on Piecewise Aggregate Approximation (PAA)[30]. Then, we calculate the minimum distance, $mdist(\hat{C}, \hat{Q})$, where $\hat{C}$ represents the subsequence of current time and $\hat{Q}$ represents any subsequence of history data.

$$\mathrm{EA}^{(3)} = \sum_{i=1}^{m}\left(\sum_{j=1}^{k} w_i\varphi_j \cdot mdist(\hat{C}_i, \hat{Q}_i^j)\right) + \varepsilon \tag{5}$$

where φ_j denotes the weight of previous j-th week and $\sum_{j=1}^{k}\varphi_j = 1$.

In the paper, our alphabet of cardinality is set to 10. In accordance with Eqs. (3)–(5), the value of evidence accumulation deviation degree can be calculated, as shown in Eq. (6).

$$\begin{aligned}\mathrm{EA} = {} & \theta_1\sum_{i=1}^{m}\left(w_i \cdot \max\{|\ f^i(t) - f^i(x)\ |\}\right) + \\ & \theta_2\sum_{i=1}^{m}\left(\frac{w_i}{k}\sum_{j=1}^{k}\left(\frac{f_i^j(t)}{\max\{f_i^j(t)\}}\right)\right) + \\ & \theta_3\sum_{i=1}^{m}\left(\sum_{j=1}^{k} w_i\varphi_j \cdot mdist(\hat{C}_i, \hat{Q}_i^j)\right) + \varepsilon\end{aligned} \tag{6}$$

Algorithm 1 outlines the major steps of the network behavior anomaly detection algorithm. We develop a scalable algorithm using a proposed traffic feature extraction approach and the time series anomaly detection technique, to detect the two possible types of traffic volume and communication pattern anomalies.

5 Experimental Evaluation

In this work, Apache Kafka, Apache Flume, and Apache Storm were combined for the big data analysis. However, using multidimensional data can be problematic due to computing expenses, making such data inappropriate for real-time calculations to be used in actual practice. Multiple dimension features are necessary for the proper profiling time series network behaviors in the real world. Traditional ways of massive data processing, analyzing, and storage for network traffic may not work effectively.

5.1 Experimental system setup

(1) Network architecture for data collection. The data source used in this work is captured from the campus network and we only process the Internet Protocol Version 4 (IPV4) packets. The system adopts the port mirroring technique to accurately capture network traffic.

The runtime environment is located in a campus network, as illustrated in Fig. 3. Though port mirroring, one switch port sends a copy of data packets to another switch port, so all of the packets are redirected to

Algorithm 1 Anomaly identification algorithm

Input: network traffic session data
Output: alarm

get stream data and set time window ;
for *each rdd in DStream* **do**
  $slice_{his}$ = get previous time-window slice data;
  $slice_{now}$ = get current time-window slice data ;
  parse session data $rddslice_{his}$;
  parse session data $rddslice_{now}$;
  if $rddslice_{his} \neq null$ **then**
    ftv_s = caculate static traffic volume features;
    build edge data $edge$RDD;
    create new graph $rddgraph$;
    caculate static traffic activity graph features $ftag_s$;
  end
  if $rddslice_{his} \neq null$ *and* $rddslice_{now} \neq null$ **then**
    caculate common data between previous and current $rddcommon$;
    caculate active data this time window $rddactive$;
    caculate dynamic traffic volume features ftv_d;
    caculate dynamic traffic activity graph features tdg_d ;
  end
  $f = ftv_s \cup ftag_s \cup ftv_d \cup tdg_d$;
  the feature vector f is as input data of anomaly detection; // or save to database
  get current time type tc workday or weekend;
  get history data $\mathbb{R}_{tc}$ for absolute changes ;
  get history data $\mathbb{Z}_{tc}$ for relative changes ;
  get history data Q_{tc} for trend changes ;
  filter anomalous times on feature set $\mathbb{R}$, $\mathbb{Z}$, and Q;
  get current feature vector $singledata(t) = f(t)$;
  get subsequence data $seqdata$;
  for *each* f_i *in* $singledata(t)$ **do**
    $EA^{(1)}(t)$ =caculate the absolute change;
    $EA^{(2)}(t)$ =caculate the relative change ;
  end
  for *each* f_i *in* $seqdata(t)$ **do**
    get i-th feature subsequence history data $\hat{Q}_i$;
    get i-th feature subsequence current data $\hat{C}_i$;
    $EA^{(3)}(t)$ =caculate the trend change;
  end
  $EA(t)$ =caculate evidence accumulative degree;
  if $EA > \varphi$ **then**
    alarm and save EA to $MySQL$;
  end
end

the collection router. Then, in our big data cluster (computing cluster and storage cluster), we can capture network traffic from the collection router. We use PF_RING technology to capture the packets from our server with the 10 Gbps optical network card. Next, we employ Apache Flume to obtain the packets from one network card and then Apache Kafka technology is employed to transmit the packets to another network card. Finally, the function of step 1 is implemented, as shown in Fig. 3.

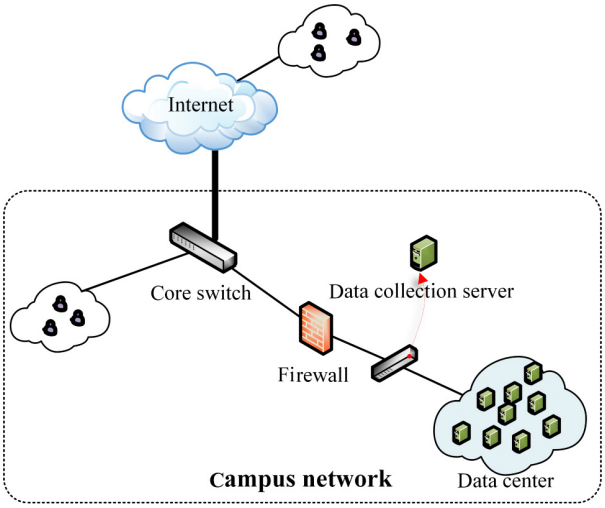


Fig. 3 Network architecture.

(2) Big data platform for data processing. The proposed algorithms are developed on Spark 1.6.0 using Scala 2.10.4, continuously executes for real-time anomaly detection at campus data center. To launch a Spark application in cluster mode, we need to set the proper configuration to obtain the better performance. Our Spark application properties in runtime, shown in Table 2, demonstrate good performance.

Our big data cluster includes several nodes using Tecal RH2288H V2. All the nodes have 10 cores of two Intel(R) Xeon(R) Processor E5-2658 v2 2.40 GHz with 256 GB of Random Access Memory (RAM). The connection between each node is 10 Gbps. The big data platform is set up with Hadoop YARN 2.6.0 on CentOS 6.5 as a single cluster, which is configured with a single NameNode and several DataNodes. Each node has the same application deployment. Spark is deployed in the Hadoop cluster managed by YARN.

5.2 Feature analysis

We make several key attempts to explore the

Table 2 Spark application properties.

Property	Value
Deploy mode	Cluster
Driver cores	2
Driver memory	2 GB
Executor counts	16
Executor memory	1 GB

regular feature fluctuations and correlations among the proposed four types of feature sets. We make the following observations: (i) these features indicate that the daily and weekly patterns of network behaviors have regular periodicity, (ii) several features are almost unchanged, except for several anomalous fluctuations, (iii) communication pattern features show pronounced increasing peaks and decreasing valleys at regular intervals, and (iv) when anomalies occur, different types of features show anomalous fluctuations at varying time slots and the same type feature may show different degree changes.

(1) Traffic volume feature. Figure 4 shows the traffic volume static feature. We observe a similarity between the traffic volume in Fig. 4a and the data packet counts in Fig. 4b. The red and blue lines represent the protocol of Transmission Control Protrol (TCP) data and the User Datagram Protocol (UDP) data, respectively. The feature values of network behavior variability or burstiness are similar, especially with the long-range time scale. Evidently, these features indicate the daily and weekly patterns. Even with the same traffic volume static features, Figs. 4a and 4b show anomalous fluctuations at different time slots. Furthermore, as shown in Fig. 4, we can conclude that the anomalies occurring at time slot a1 exhibit the change of feature values on packet counts that are greater than the volume.

We find that the daily and weekly patterns of network communication behaviors in the data center have regular periodicity, which is in accordance with the findings of past works[31, 32]. Moreover, some servers only provide services (e.g., database, backup, cluster, etc.) and cannot be directly accessed by external users. These network communication behaviors of the data center reflect the provision mode for daily service and the network behaviors of external users[33, 34].

Figure 5 shows the feature values of the ratio of connection counts related to traffic volume dynamic features. As can be seen in the figure, the trends are almost unchanged, except for several anomalous fluctuations within the time series data.

(2) Communication structure feature. Figure 6 plots the same time range on the static graph features calculated over 30-minute intervals. Figures 6a–6c represent the number of nodes that only have incoming edges, outgoing edges, and both incoming and outgoing edges, respectively, as observed in the snapshot graph. Notice that the number of nodes rises significantly in Fig. 6a but not in Fig. 4. Figure 4 reflects the traffic volume features and Fig. 6 reflects the communication pattern features at the time-scales of interest. Notably, the variabilities of network behavior features appear in different time slots in Figs. 6a–6c.

Figures 7a–7c represent the graph edit distance and the number of edges overlap or nodes overlap that appear in adjacent dynamic graphs, respectively. We find that significant stability and time-series data values over adjacent time slots do not change abruptly during the 7-day period. Figure 7 shows pronounced increasing peaks and decreasing valleys at regular intervals.

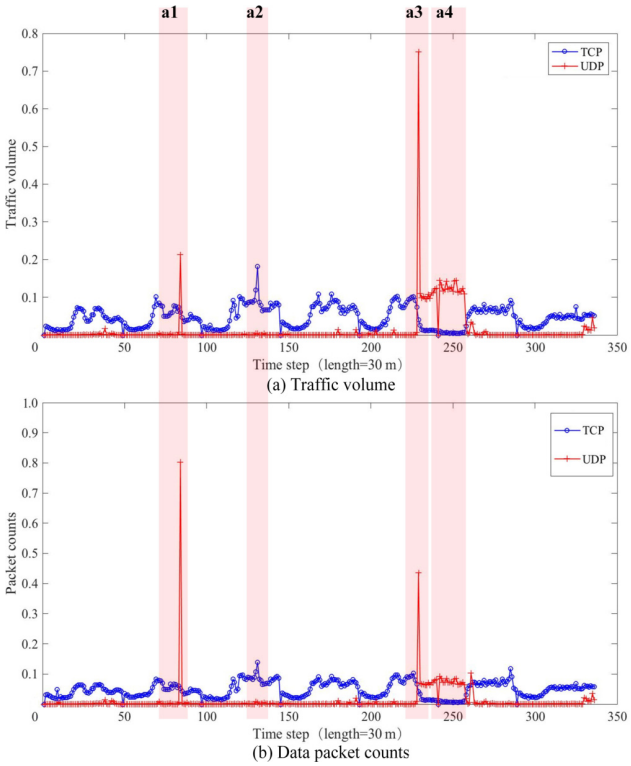


Fig. 4 Static features of session flow.

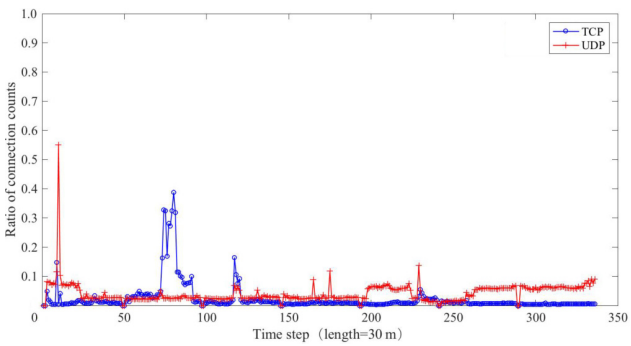
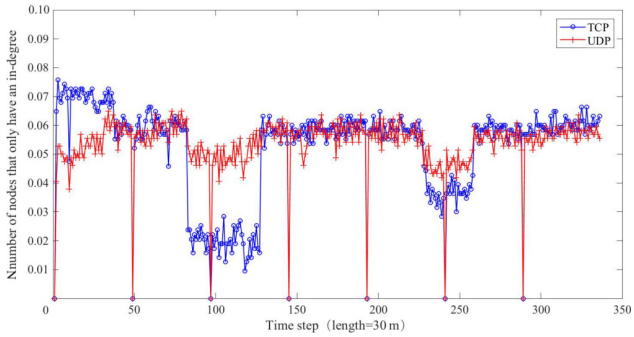
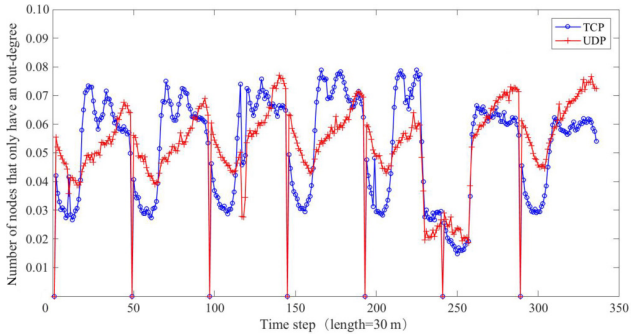


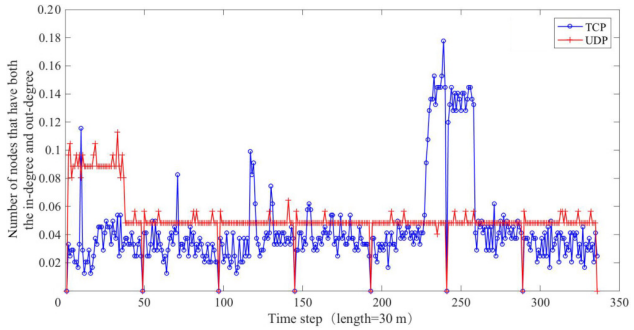
Fig. 5 Dynamic features ratio of connection counts.



(a) Number of nodes that only have an in-degree



(b) Number of nodes that only have an out-degree



(c) Number of nodes that have both the in-degree and out-degree

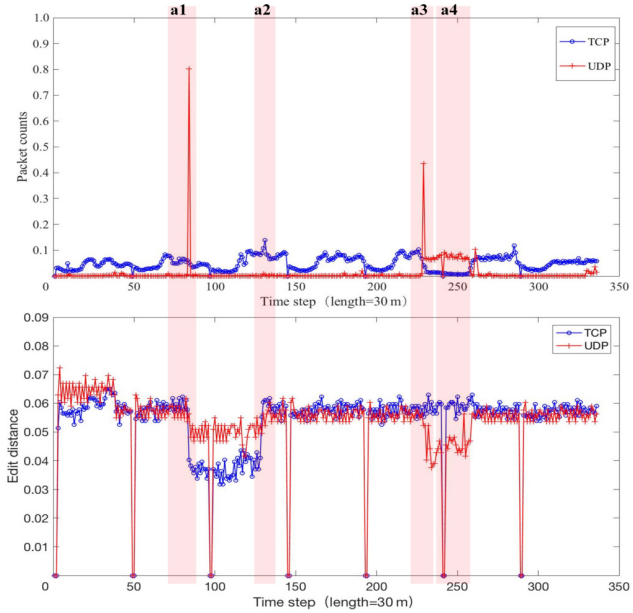
Fig. 6 Static features of the graph.

Next, we compare the traffic volume feature of packet counts with the feature of graph edit distance. The anomalies occurring at time slot a1 and the feature of packet counts demonstrate the sudden huge growth; although the feature values are almost normal in the next time slot, the anomalies continue. The correlations reflected in the graph edit distance at that point suddenly decrease as well; the anomalies lasted for almost 50 time slots, as shown in Fig.7a.

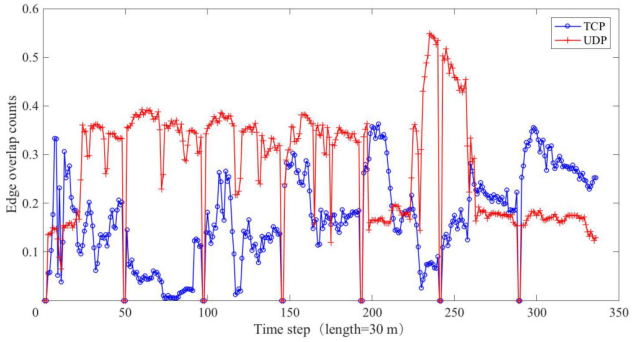
5.3 EADD method

The proposed real-time feature extraction and anomaly detection approach EADD is implemented in our campus network environment. We also apply our proposed feature set to the MIT-DARPA 1999 intrusion detection dataset.

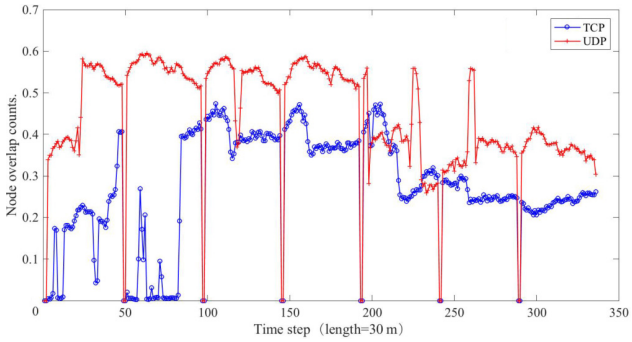
(1) Real-world dataset. By observing a total of



(a) Demonstration of the different time points by comparing graph edit distance to packet counts



(b) Edge overlap counts



(c) Node overlap counts

Fig. 7 Dynamic features of the graph.

3 814 373 408 session flow records collected over 181 days, we found that the number of daily average session flow records is 21 073 886 and the number of daily average users is 387 406. The expiration times of the TCP and UDP session flows are 5 min and 20 s, respectively. The system uses a batch interval of 60 s and a time window of 60 s in realizing the real-time

processing of the received session flow records. As for the selection of the parameters for detection algorithms, in absolute change, each feature is referenced to the adjacent 10 points at a given time, whereas in relative change, each feature is referenced to the adjacent time in the previous 3 weeks and the current week. Meanwhile, in trend change, the length of the original time series is 20, which is reduced to 10 dimensions via PAA. Moreover, considering the correlations across time, the nearer the recent data are assigned to the feature value x, the greater their higher weight.

The system detects the anomalous behaviors during network operation, which may be due to the network transformation of the data center by the network administrator, server migration (e.g., change of the server's IP address, network devices addition and deletion, and so on), the greater number of connections to the rarely accessed network servers, and other unusual behaviors of the network management. These useful and seemingly legitimate exceptions, which are often different from the usual network behaviors, can be found in time, as shown in Fig. 8.

Our approach can complement the existing traffic volume data analysis by identifying the specific phenomena we are missing because they are just not high volume enough, including BotNets, P2P, established services, and so on. Figure 8 shows the results of the real-time detection of the proposed approaches. Our proposed approach may not cause false alarms because the EADD is always below 9 on the internal network communication (and below 97 on the external network communication) in this work.

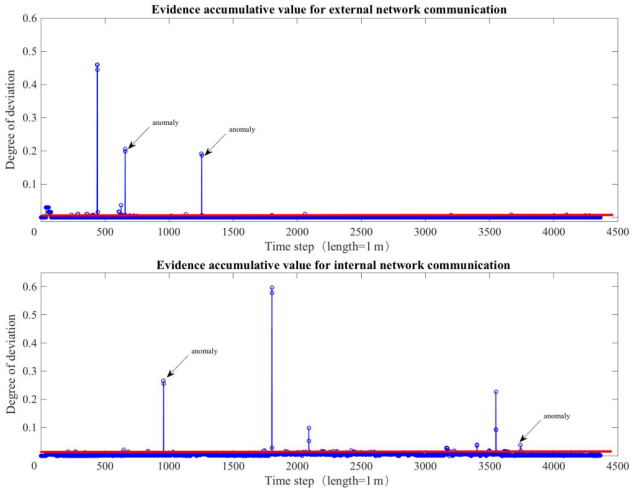


Fig. 8 Real-time anomaly detection.

To alleviate the effect of noise, the approach EADD only produces non-zero charges when the current time window EADD is larger than the normal EADD by thrice the standard deviation.

We apply the communication pattern features and the traffic volume features together to detect the anomalies, and then label the whole detected anomaly instance. Then, we compare the contributions of different feature sets with the same detected instance by conducting two experiments. The first experiment is that only traffic volume features are chosen to detect anomaly. The second experiment is that only communication pattern features are chosen. After several experiments, we find in the real-world data that the traffic volume features and communication structure features play complementary roles in the different network anomalies. The approach shows the highest accuracy with both feature sets as well as lower accuracies of 68.63% and 79.91% with just the traffic volume features and communication structure features, respectively, as shown in Table 3. As mentioned above, different network anomalies show the changes of different network features.

(2) 1999 DARPA dataset. We validate the detection precision of the proposed approaches on the MIT-DARPA 1999 dataset[35]. We consider only the outside traffic in our evaluation and analyze the traffic data on Monday, Week 5. The traffic data include not only the normal behaviors, but also the attacking activities. The numbers of each attack type can be found in Table 4.

Table 3 Accuracy result in different features.

Feature category	Accuracy (%)
Traffic volume feature	68.63
Communication structure feature	79.91
Both	100

Table 4 Numbers of each type of attack.

Attack type	Number of attacks	Attack type	Number of attacks
crashiis	2	queso	14
eject	6	secret	2
fdformat	10	selfping	2
httptunnel	2	tcpreset	3
neptune	6	teardrop	2
perl	2	xsnoop	2
pod	2	xterm	8
ppmacro	22	yaga	4
ps	6		

This work sets out to make a comparative evaluation of our feature set running on the algorithm J48 and AdaBoostM1 in WEKA. The experiments results shown in Table 5 indicate that the feature set combining traffic volume and communication structure features shows the best precision and recall. Chen et al.[21] proposed the combination of Principal Component Analysis (PCA) and wavelet analysis feature selection algorithm for better anomaly detection. Their solution was also tested using the MIT-DARPA 1999. However, their algorithm generated many false alarms (147 false alarms). In comparison, our extraction feature set, which focuses on the traffic volume and communication pattern analysis of attacks, can detect attacks that change the volume and the structure of the network overtime with a high detection rate.

Figure 9 shows the network communication structure on the MIT-DARPA 1999 dataset in the time periods of the sudden increase of feature values to a maximum degree. We visualize the collected flow-level data as a graph at four time slots. This is very useful because it enables us to better understand the changes of communication patterns. At the time slot in which the maximum degree suddenly increases, we can observe that new small nodes of clusters appear at the beginning of the attack. Group attack behaviors are observed in three hosts (172.016.114.050, 152.169.215.104, and 206.048.044.050, respectively). By sequential observation, we find that the three clusters are larger than the prior time slot. Comparing the changes of the other features, we show a time series of the traffic volume. There is no sudden change in that time series, and such correlated anomalous behaviors cannot be detected through traffic volume features. Therefore, the proposed extraction feature set is more efficient than the other method in detecting the changes in communication patterns.

Table 5 Results in different feature sets.

Classified features	J48 (%)		AdaBoostM1 (%)	
	Precision	Recall	Precision	Recall
Traffic volume	79.60	86.10	82.30	86.10
Communication structure	79.90	73.00	89.90	94.80
Both	87.60	89.50	94.00	95.20

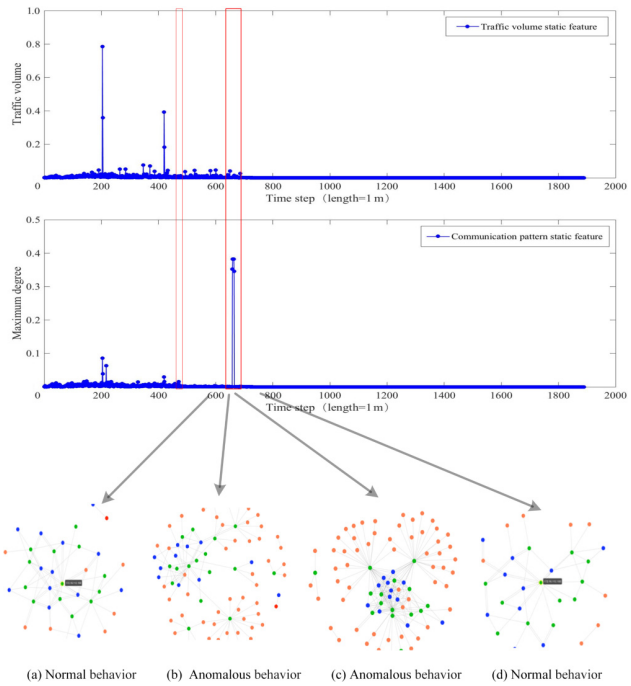


Fig. 9 Anomalous and normal periods.

Two experiments are carried out to prove the validity and feasibility of our proposed approach. The reason why a lower accuracy is obtained could be due to the different network anomalies causing different changes of feature sets on a real-world dataset. Different network anomalies show the changes of different network features. As we have seen, the traffic volume and communication pattern features can be complementary to each other. However, this does not mean that the features of the graph are better than the traffic volume features. During this detection period, network anomalies cause greater changes on the communication pattern features.

6 Conclusion and Future Works

We propose a real-time feature extraction architecture to profile traffic volume and the communication patterns for network behavior anomaly detection. We define four types of flow-level feature sets and explore the regular feature fluctuations and the correlations among them in anomalous time windows. We establish a real-time anomaly detection approach (i.e., EADD), provide the quantification of the absolute change and the fluctuations in the trend changes to discover the anomalous behaviors accurately. Next, we demonstrate the applications of combining the benefits of the traffic volume and the communication pattern feature sets in detecting conventional traffic volume and low-intensity anomalous behaviors through a real-world dataset and a validation dataset.

In the future, we plan to determine the cause of anomalous behaviors, which is our next research goal. We also plan to apply machine learning, outliers, signal

analysis, optimization algorithm, and visualization, to comprehensively understand the dynamic behavior of complex networks and improve the accuracy of anomaly identification.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 61272447), Sichuan Province Science and Technology Planning (Nos. 2016GZ0042, 16ZHSF0483, and 2017GZ0168), Key Research Project of Sichuan Provincial Department of Education (Nos. 17ZA0238 and 17ZA0200), and Scientific Research Staring Foundation for Young Teachers of Sichuan University (No. 2015SCU11079).

References

[1] K. Xu, F. Wang, and L. Gu, Behavior analysis of internet traffic via bipartite graphs and one-mode projections, *IEEE/ACM Trans. Netw*., vol. 22, no. 3, pp. 931–942, 2014.

[2] A. Sperotto, R. Sadre, P. T. Boer, and A. Pras, Hidden Markov model modeling of SSH brute-force attacks, in *Proc. 20th IFIP/IEEE Int. Workshop on Distributed Systems*: *Operations and Management*: *Integrated Management of Systems Services Processes and People in IT*, Venice, Italy, 2009, pp. 164–176.

[3] K. Huang, Z. W. Qi, and B. Liu, Network anomaly detection based on statistical approach and time series analysis, in *Proc. 23th Int. Conf. Advanced Information Networking and Applications Workshops*, Bradford, UK, 2009, pp. 205–211.

[4] T. Andrysiak, Ł Saganowski, M. Choraś, and R. Kozik, Network traffic prediction and anomaly detection based on ARFIMA model, in *Proc. Int. Joint Conf. SOCO'14-CISIS'14-ICEUTE'*14, Bilbao, Spain, 2014, pp. 545–554.

[5] M. M. Ding and H. Tian, PCA-based network traffic anomaly detection, *Tsinghua Sci. Technol.*, vol. 21, no. 5, pp. 500–509, 2016.

[6] X. M. Ye, X. S. Chen, H. Z. Wang, X. M. Zeng, G. L. Shao, X. Y. Yin, and C. Xu, An anomalous behavior detection model in Cloud Computing, *Tsinghua Sci. Technol.*, vol. 21, no. 3, pp. 322–332, 2016.

[7] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level, *IEEE/ACM Trans. Netw.*, vol. 5, no. 1, pp. 71–86, 1997.

[8] T. Babaie, S. Chawla, and S. Ardon, Network traffic decomposition for anomaly detection, *Computer Science*, vol. 96, no. 2, pp. 201–212, 2014.

[9] P. Winter, H. Lampesberger, M. Zeilinger, and E. Hermann, On detecting abrupt changes in network entropy time series, in *Proc. 12th IFIP TC 6/TC 11 Int. Conf. Communications and Multimedia Security*, Ghent, Belgium, 2011, pp. 194–205.

[10] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, On the self-similar nature of Ethernet traffic (extended version), *IEEE/ACM Trans. Netw.*, vol. 2, no. 1, pp. 1–15, 1994.

[11] M. Iliofotou, M. Faloutsos, and M. Mitzenmacher, Exploiting dynamicity in graph-based traffic analysis: Techniques and applications, in *Proc. 5th Int. Conf. Emerging Networking Experiments and Technologies*, Rome, Italy, 2009, pp. 241–252.

[12] L. Akoglu, H. H. Tong, and D. Koutra, Graph based anomaly detection and description: A survey, *Data Min. Knowl. Discov.*, vol. 29, no. 3, pp. 626–688, 2015.

[13] D. Q. Le, T. Jeong, H. E. Roman, and J. W. K. Hong, Traffic dispersion graph based anomaly detection, in *Proc. 2nd Symp. on Information and Communication Technology*, Hanoi, Vietnam, 2011, pp. 36–41.

[14] M. S. Rahman, T. K. Huang, H. V. Madhyastha, and M. Faloutsos, Efficient and scalable socware detection in online social networks, in *Proc. 21st USENIX Conf. Security Symp.*, Bellevue, WA, USA, 2012, p. 32.

[15] U. Khurana, S. Parthasarathy, and D. Turaga, Graph–based exploration of non-graph datasets, *Proc. VLDB Endow.*, vol. 9, no. 13, pp. 1557–1560, 2016.

[16] C. R. Harshaw, R. A. Bridges, M. D. Iannacone, J. W. Reed, and J. R. Goodall, GraphPrints: Towards a graph analytic method for network anomaly detection, in *Proc. 11th Annu. Cyber and Information Security Research Conf.*, Oak Ridge, TN, USA, 2016, pp. 1–4.

[17] J. François, S. N. Wang, R. D. State, and T. Engel, BotTrack: Tracking botnets using NetFlow and PageRank, in *Proc. 10th Int. IFIP TC 6 Conf. Networking*, Valencia, Spain, 2011, pp. 1–14.

[18] Q. Ding, N. Katenka, P. Barford, E. Kolaczyk, and M. Crovella, Intrusion as (anti)social communication: Characterization and detection, in *Proc. 18th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Beijing, China, 2012, pp. 886–894.

[19] S. Weigert, M. A. Hiltunen, and C. Fetzer, Community-based analysis of netflow for early detection of security incidents, in *Proc. 25th Int. Conf. Large Installation System Administration*, Boston, MA, USA, 2011, p. 20.

[20] K. Ishibashi, T. Kondoh, S. Harada, T. Mori, R. Kawahara, and S. Asano, Detecting anomalous traffic using communication graphs, in *Telecommunications*: *The Infrastructure for the 21st Century*, Vienna, Austria, 2010, pp. 1–6.

[21] Z. M. Chen, K. Y. Chai, S. L. F. Bu, and C. T. Lau, Combining MIC feature selection and feature-based MSPCA for network traffic anomaly detection, in *Proc. 3rd Int. Conf. on Digital Information Processing, Data Mining, and Wireless Communications*, Moscow, Russia, 2016, pp. 176–181.

[22] J. Tan, X. S. Chen, M. Du, and K. Zhu, A novel internet traffic identification approach using wavelet packet decomposition and neural network, *J. Cent. South Univ.*, vol. 19, no. 8, pp. 2218–2230, 2012.

[23] S. R. Kundu, S. Pal, K. Basu, and S. K. Das, Fast classification and estimation of Internet traffic flows, in *Proc. 8th Int. Conf. Passive and Active Network Measurement*, Louvainla-Neuve, Belgium, 2007, pp. 155–164.

[24] P. Barford and D. Plonka, Characteristics of network traffic flow anomalies, in *Proc. 1st ACM SIGCOMM Workshop*

on Internet Measurement, San Francisco, CA, USA, 2001, pp. 69–73.

[25] H. Bunke, P. J. Dickinson, M. Kraetzl, and W. D. Wallis, A graph-theoretic approach to enterprise network dynamics, *Progress in Computer Science and Applied Logic*, vol. 24, pp. 63–78, 2007.

[26] M. Iliofotou, H. C. Kim, M. Faloutsos, M. Mitzenmacher, P. Pappu, and G. Varghese, Graption: A graph–based P2P traffic classification framework for the internet backbone, *Comput. Netw.*, vol. 55, no. 8, pp. 1909–1920, 2011.

[27] C. Chaparro and C. Eberle, Detecting anomalies in mobile telecommunication networks using a graph based approach, in *Proc. 28th Int. Florida Artificial Intelligence Research Society Conf.*, Hollywood, FL, USA, 2015, pp. 410–415.

[28] A. Sanfeliu and K. S. Fu, A distance measure between attributed relational graphs for pattern recognition, *IEEE Trans. Syst. Man. Cybern.*, vol. 13, no. 3, pp. 353–362, 1983.

[29] L. Mookiah, W. Eberle, and L. Holder, Detecting suspicious behavior using a graph-based approach, in *Proc IEEE Conf. Visual Analytics Science and Technology*, Paris, France, 2014, pp. 357–358.

[30] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, A symbolic representation of time series, with implications for streaming algorithms, in *Proc. 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, San Diego, CA, USA, 2003, pp. 2–11.

[31] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, Dimensionality reduction for fast similarity search in large time series databases, *Knowl. Inf. Syst.*, vol. 3, no. 3, pp. 263–286, 2001.

[32] T. Karagiannis, M. Molle, and M. Faloutsos, Longrange dependence ten years of Internet traffic modeling, *IEEE Internet Comput.*, vol. 8, no. 5, pp. 57–64, 2004.

[33] S I. Tadaki, Long-term power-law fluctuation in Internet traffic, *J. Phys. Soc. Jpn.*, vol. 76, no. 4, p. 044001, 2007.

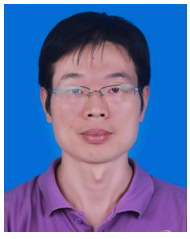
[34] G. Samorodnitsky, Long range dependence, *Found. Trends Stoch. Syst.*, vol. 1, no. 3, pp. 163–257, 2007.

[35] M. V. Mahoney and P. K. Chan, An analysis of the 1999 DARPA/Lincoln laboratory evaluation data for network anomaly detection, *Recent Advances in Intrusion Detection*, vol. 1, no. 1, pp. 220–237, 2003.

Xiaoming Ye is a lecturer in the School of Cybersecurity at Chengdu University of Information Technology. She received the PhD degree from Sichuan University in 2018. She got the BE degree from Jiangnan University in 2005. She received the MS degree from Sichuan University in 2008. Her research interests include cyber security and big data.

Xingshu Chen received the PhD degree from Sichuan University in 2004. She is now a professor in the College of Cybersecurity at Sichuan University. She is the member of China Information Security Standardization Technical Committee. Her research interests include cloud computing, cloud security, distributed file system, big data processing, network protocol analysis, and new media supervision.

Dunhu Liu received the PhD degree from Sichuan University in 2010. Currently, he is an associate professor at Chengdu University of Information Technology. His research interests include achievements transformation and data mining.



Wenxian Wang is currently a lecturer in the College of Cybersecurity at Sichuan University. He received the PhD degree in 2014 from Sichuan University. His research interests include network security and intelligence analysis.

Li Yang is a lecturer at Chengdu University of Information Technology and a PhD candidate in the School of Economics and Management at Southwest Jiaotong University. He received the MS degree from Sichuan University in 2009. His research interests include data mining and consumer behavior.

Gang Liang received the PhD degree in computer science from Sichuan University in 2007. He is currently an associate professor in the College of Cybersecurity at Sichuan University. His research interests include network security social networks and machine learning.

Guolin Shao is a PhD candidate in the College of Computer Science at Sichuan University. He got the BE degree from Sichuan University in 2013. His general research interest is cyber security.