IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 24, 2019, accepted March 10, 2019, date of publication March 28, 2019, date of current version April 10, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2907149

Flow Context and Host Behavior Based Shadowsocks's Traffic Identification

XUEMEI ZENG[1], XINGSHU CHEN[2], GUOLIN SHAO[3], TAO HE[2], ZHENHUI HAN[2], YI WEN[2], AND QIXU WANG[2]

[1]College of Computer Science, Cybersecurity Research Institute, Sichuan University, Chengdu 610065, China
[2]College of Cybersecurity, Cybersecurity Research Institute, Sichuan University, Chengdu 610065, China
[3]College of Computer Science, Sichuan University, Chengdu 610065, China

Corresponding author: Xingshu Chen (chenxsh@scu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61802270 and Grant 61802271, in part by the Key Research and Development Project of Sichuan Province of China under Grant 2018GZ0100, and in part by the Fundamental Research Business Fee Basic Research Project of Central Universities under Grant 2017SCU11065.

ABSTRACT Cloud Virtual Private Server (VPS) services provide the chance of rapid deployment of anonymous proxy services, becoming an important part of many anonymous proxy solutions. The anonymous system represented by ShadowSocks (SS), through proxy services deployed on VPSs provided by different cloud service providers, has become an important mean for illegal network activists to engage in illegal network activities such as cyber-attacks and darknet transactions. It is difficult for local network administrators to supervise SS traffic from the cloud. While from the local network, the task faces the challenges of *Invisible Negotiation Process* and *Data Transparent Transmission*. In this paper, we present a novel SS detection method based on flow context and host behavior. The method can not only accurately identify SS flows, but also be applicable to a large-scale network environment. In this method, we extract 12-dimensional features from three aspects: the relationship between flows, hosts' flow behavior, and hosts' DNS behavior to build the detection model. Among them, the four features about flow burst and the feature of unassociated domain names' number are innovatively proposed in this paper. Moreover, the big data statistical and association techniques are used in the method. To verify the effectiveness of the method, we first built a real SS running environment based on the campus network and two VPSs on two different public cloud platforms. Moreover, we conduct a series of experiments on the NTCI-BDP data platform which is a big data platform built by our team. The experimental results show that our method achieves 93.43% accuracy on experimental data sets and can effectively identify SS traffic.

INDEX TERMS Big Data association, cloud-based anonymous proxy, flow burst, flow context, host behavior, traffic identification, shadowsocks.

I. INTRODUCTION

Network security and privacy protection have become the focus of network users' attention in recent years. Anonymous communication technologies, such as anonymous proxy [1], anonymous routing [2] and anonymous key pair [3], are frequently used to ensure the security of network terminal and personal privacy. Cloud Virtual Private Server (VPS) provides the opportunity of rapid deployment of anonymous proxy services, becoming an important part of the anonymous proxy solution and contributing to the widespread application of anonymous proxy technology. ShadowSocks (referred to as SS), a typical anonymous proxy system, can encrypt user communications and conceal the identification of users. Due to its simple deployment process, the server side of SS can be quickly deployed in a VPS [4]. However, VPS-based SS solutions are also abused while protecting the privacy of communicators. They have currently become important means for illegal network activists to engage in illegal network activities such as data theft, pornographic propagation, cyber-attacks and darknet transactions. In these kinds of solutions, virtual servers are as the scapegoats for malicious actors, which would have a negative effect on the security assessment of cloud services [5]. Hence, the work of identifying SS traffic is of great significance to network security supervision. In these scenarios, the proxy services of SS are typically deployed

The associate editor coordinating the review of this manuscript and approving it for publication was Kuan Zhang.

on VPS of cloud data centers located in different geographic locations from different cloud providers, such as Amazon AWS, Microsoft Azure, Google GCM. It is unrealistic for local network security regulators to monitor all cloud services to identify and supervisor SS traffic. This poses a challenge to cybersecurity Supervision. While there are still several difficulties for network administrators if communications are supervised from the local network. Encryption and authentication can be implemented at different layers of ISO/OSI [6]. They are implemented by information encoding and message authentication mechanisms in the physical layer and the link layer [7]–[10], by protocol encapsulation or whole packets encryption in the network layer and the transport layer, and by specific application encryption protocols in the application layer. SS traffic is only composed of encrypted traffic based on TCP. No process of the key negotiation and exchange between clients and remote servers is seen in them and the characteristics of traffic are not obvious. The task of identifying SS traffic faces the challenges of *Invisible Negotiation Process* and *Data Transparent Transmission*. Therefore, identifying SS traffic is a great challenge.

The recent encrypted and anonymous network traffic identification solutions mainly use feature-based and statistics-based methods [11], [12], which can identify the traffic of encrypted and anonymous communication systems such as Skype [13], SSH [14] and Tor [2]. Feature-based methods rely on the traffic characteristics extracted from the plaintext information of the protocol handshake between nodes in the initial stage of the establishment of encrypted communication, or from the unencrypted protocol header message, and then identify traffic by pattern or keyword matching methods. For example, we can verify whether the connection is an SSL protocol and determine the protocol version according to the SSL configuration options by the analysis of the unencrypted SSL protocol header [15]. Statistics-based methods are mainly based on the statistical or time series features which are different from other protocols in terms of length, time, direction and other characteristics in the interaction process of different encryption protocols. Heuristic rules and machine learning methods are applied. For example, we can identify cryptographic protocols such as VoIP [16], Skype [17] and Tor [2] using a higher percentage of data packets based on a specific length. Nevertheless, in SS traffic, there are no obvious distinguish features in protocol header and features about high frequency packet length like the previously mentioned methods. The use of statistical and distribution characteristics at packet level would affect the performance of methods in large-scale networks. If features being extracted with the flow granularity, it could be difficult to obtain effective features because of the small amount of information containe+d.

Considering the above issues, a novel method to identify SS traffic was proposed in this paper. The method bases on the fact that the use of SS changes the way that the host finds the target server and the pattern of the host establishes connections with the target server. It results in changes on the behavior of host domain name requests and flow initialization, and the changes on the relationship between flows on concurrency and relevance. We firstly monitor the relationship between flows in different time windows, the behavior of the SS host on flow and DNS. Then, leveraging the big data association technology, we establish the association between flows, between flow and host, and between flow and DNS. Accordingly, we extract 12 features reflecting the distinction between SS flows and non-SS flows (other flows except SS flows) as the feature representation of the flow from the association. Finally, we construct the SS identifying model using the Random Forest algorithm. Experiments on data sets collected and generated in the real-world network show that our proposed method can effectively identify SS traffic and achieves 93.43%, 93.74%, 90.66% and 92.17% in accuracy, precision, recall and respectively.

The main contributions of this paper are as follows:

- The differences between SS flows and non-SS flows were characterized. From different perspectives, several concepts were proposed from for characterizing SS flows, which are "*Flow Context*" from the perspective of the relationship between flows, "*Flow Burst*" and "*Destination IP address Entropy*" from the perspective of host popularity, and "*Sensitive Domain Name*" and "*Unassociated Domain Name*" from the perspective of host DNS behavior. Several experiments were performed and verified that these features can be used to identify SS traffic.
- A novel SS detection method based on flow context and host behavior was proposed. This method applies big data statistics, association and data processing technology to implement the association of flows and DNS data, feature extraction and calculation and detection model construction. The detector not only can accurately identify SS traffic, but also is suitable for large-scale network environments.
- The effectiveness of our proposed method was evaluated on the experimental dataset from the real network and compared it with the state-of-the-art methods. The SS running environment was built based on our campus network and two virtual servers running on different public cloud platforms. All experimental data is collected from the specific edge router of our campus network and labeled. The experimental results show that the method is effective for identifying SS flow and is superior to the comparison methods.

The main novelties of this paper are the proposed four features about flow burst and the feature of the number of unassociated domain names. The four features about flow burst characterize the concurrency of multiple flows between the source host and the target service. The feature of unassociated domain name characterizes the temporal association between the requested domain names and the initialized flows by the source host. These features are firstly proposed as far as we know and can reflect the difference between SS flows and non-SS flows from the perspective of host behavior.

Experimental results show that these features can be used to identify SS traffic.

The rest of this paper is organized as follows. In Section II, the related work in the fields of traffic identification, encryption and anonymous traffic identification is briefly introduced. Then, in Section III, the operation mechanism of SS is analyzed, and we analyze the characteristics of SS communication from the aspects of SS flow context, host behavior on flow, and host behavior on DNS. After that we elaborate on the SS flow detection method in Section IV. In Section V several experiments were conducted to verify the effectiveness of our proposed method. Finally, the work of this paper is summarized in Section VI.

II. RELATED WORK

A. METHODS OF TRAFFIC IDENTIFICATION

Currently, the main methods of traffic identification are based on port, payload, statistics and behavior. Port-based methods and payload-based methods are quick and simple. However, these two kinds of methods are significantly lower in identification accuracy because of the wide application of dynamic port technology, protocol masquerading and confusion, encryption technology, etc. Payload-based methods can still be used to identify and coarse-grained specific encrypted traffic. For example, SSL can be identified based on the unencrypted header information of the message during the handshake [14]. Statistics-based methods, which are not affected by changes in payload, are based on statistical properties of traffic and leverage machine learning methods to identify traffic. They are suitable for the identification of encryption protocols or private protocol traffic. According to different statistical granularities, statistics-based methods can be divided into methods on packet level and methods on flow level. The former uses the distribution features of the packets in flows in terms of direction, arrival time and byte size, such as unidirectional min/max inter packet time, unidirectional min/max packet size. The latter is based on statistical features such as numbers and bytes of packets about up-flow and down-flow, and the duration of flows. Tan *et al.* [18] extracted 40 features from the flow level and the packet level and built a P2P identification model with a detection rate higher than 95%. Soleimani *et al.* [19] identified Tor plugins traffic using statistical features of flow-level and the statistical features of the first 10 to 50 packets with a high certainty. Silva *et al.* [20] extended the flow feature set of OpenFlow calculators in SDN into three categories: statistical features, scalar features and complex features to realize traffic classification in SDN networks. Behavior-based methods characterize the network traffic behavior of a host to achieve traffic identification by the connection mode that the host with the other hosts (connection degree, number of ports, etc.), the traffic communication of the host, or all traffic of the host communication with a specific IP and a port, etc. Li *et al.* [21] combine traffic statistics features with host behavior to identify VoIP traffic. XIONG *et al.* [22] proposed a method based on communication modes between hosts, and between hosts and servers to identify encrypted P2P traffic. However, these methods are typically based on pattern of communication and the behavior of hosts in specific protocol, and cannot be directly used for SS traffic identification.

In recent years, the correlation between flows is also used to characterize flow behaviors or host behaviors to implement traffic identification and classification. Zhang *et al.* [23] used destination side heuristic to design a traffic classification method based on bag-of-flows (BOF). The method combined flows with the same destination address, destination port, and protocol into one BOF, and then labeled each flow with the classification of most flows in the BOF to improve the accuracy of traffic classification. Furthermore, the authors also analyzed and proved that BOF ideas could be used to improve classifier accuracy from both theoretical and practical perspective [24]. Ding *et al.* [25] proposed a traffic classification method based on the relation of flows. The method introduced seven types of relationships between flows as extended feature vectors of a flow to perform flow classification based on difference shared quaternion attributes. Traffic classification methods based on flow association relationships provide a very good idea for realizing traffic classification and improving traffic classification accuracy at the flow level. Whereas, these methods only focus on the contribution of the number of associated flows and the shared attributes to the classification. They do not analyze more characteristics between flows.

Big data analysis technology has been applied to traffic identification and analysis in network operations, security analysis and intrusion detection [26]–[28], which makes it possible to use more environmental or contextual information to help specific traffic identification. Environmental and contextual information has been used by researchers to solve many practical problems in a variety of application scenarios [29]–[31]. In some network traffic analysis scenarios, such as only flow-level traffic data available, the information provided by a single flow is not enough to identify specific traffic. Some methods based on flow context data were thus proposed. Anderson *et al.* [32] proposed a method for malware traffic identification based on flow context data. The method utilizes background traffic information related to TLS flow, such as DNS response, HTTP header information in a 5 minute time window of the same source IP address, and unencrypted TLS handshake information, to identify the encrypted malicious traffic. In order to improve the adaptability of classification methods in different scenarios in [33], the context features, such as network conditions and user preferences were added to the network traffic identification to meet the needs of intelligent network selection in 5G environment, and different traffic is allocated to the appropriate network interface.

B. ENCRYPTED AND ANONYMOUS TRAFFIC IDENTIFICATION

The detection of encryption and anonymous traffic have been the focus of many researchers for many years. Research subjects include Tor, SSL/TLS, SSH, etc. In terms of Tor traffic identification, Rao *et al.* [34] proposed a Gravitational Clustering Algorithm (GCA) for realizing Tor anonymous traffic identification with a detection rate of over 80%. For the problem of accessing to Tor networks through bridge nodes to avoid detection, Lingyu *et al.* [35] extracted four communication characteristics from the packet length and packet arrival time dimension of the flows between clients and bridge nodes, and then used the decision tree method to identify Tor traffic. Cuzzocrea *et al.* [36] extracted the 23-dimensional features such as the packets sending time interval, the packets arriving time interval, the number of flows per second, and the number of packets per second from the flow level and packet level of the network traffic to determine whether the host is generating TOR traffic. Finally, they validated the validity of the proposed feature using a variety of machine learning methods. Mercaldo and Martinelli [37] used a similar method to identify Tor traffic too. In other aspects of encryption and anonymous traffic identification, Draper-Gi *et al.* [38] proposed a method for identifying VPN traffic based on 23 features such as packet arrival time interval, active time of packets in flows, number of packets per second and so on. After that, their team used the same method to prove that the Tor traffic can be identified and portrayed only using time-based features with accuracy and recall rates exceeding 90% [39]. Ding and Li [40] proposed a hybrid method to identify and classify SSL/TLS encrypted traffic, extracting features from the dimensions of maximum, minimum, average, and standard deviation of packet length and time interval respectively, and applying C4.5 decision tree to construct traffic identification model. In the case of the first 4 packets, the accuracy could reach 99.3%. Lee *et al.* [41] proposed a Skype traffic detection system based on integrated mode.

However, these researches are targeted at specific anonymous protocol or application, lacking of solutions to identifying SS traffic. In the few researches about SS traffic, most of them are the identification of specific websites and web pages on SS traffic based on fingerprint attacks [42], [43]. To the best of our knowledge, only Deng *et al.* proposed a random forest-based SS detection method in 2017 [44]. The method extracts more than 3000 features from bidirectional flows and network packets host profile, and the detection rate reaches 85% or more on the experimental data set. However, this method extracts a lot of features at the packet level and is apt to suffer the problems of performance and availability in large-scale network environments. In this work, we will perform SS traffic identification at the flow level in order to be applicable to large-scale networks.

III. ANALYSIS OF PROBLEM AND SS TRAFFIC CHARACTERISTICS

In this section, we first descript the operating mechanism of SS and analyze the difficulties of SS traffic detection. Then we analyze the differences between SS flows and non-SS flows from traffic characteristics and host behavior. These work will provide the basis for establishing SS traffic identification model.

A. PROBLEM ANALYSIS

SS is a single-hop anonymous proxy system based on the Socks5 protocol and its traffic exposed to the network supervisor is encrypted TCP flows [4]. SS software is composed of two parts: the client *SS_Local* and the server *SS_Server*. *SS_Local* is typically deployed on a local machine, router, or other machine on the local network, and starts a local Socks5 agent server. *SS_Server* is deployed outside the firewall. The communication process of SS is shown in Figure 1. Specific steps are as follows: (1) The user application communicate with the *SS_Local* based on the Socks5 protocol. (2) After the connection is established, *SS_Local* automatically encrypts the application request data and forwards them to the *SS_Server* via TCP flows. (3) *SS_Server* decrypts the encrypted data received from *SS_Local*, parses the user requests and forwards them to the target server. (4) After obtaining the responses of the target server, *SS_Server* sends the response data back to the client through TCP flows in the same encryption mode. *SS_Local* can communicate with *SS_Server* based on multiple options including encryption method, encryption key, and service port. These options are pre-configured, and remain unchanged once transmission begins.

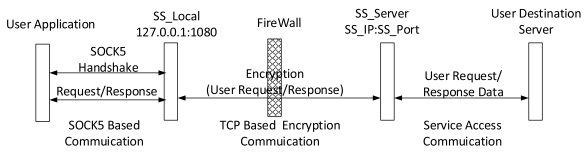


FIGURE 1. The communication process of SS system.

From the operation mechanism of SS, SS traffic detection is mainly faced with the two challenges of *Invisible Negotiation Process* and *Transparent Data Transmission* compared to the other anonymous traffic. They are detailed as follows.

- *Invisible Negotiation Process*. Leveraging the plaintext information in protocol negotiation process or the unencrypted header field message, we can identify the encrypted traffic such as SSH or VPN. However, the SS traffic passing through the firewall is only TCP-based encrypted flows. There is no key exchange and negotiation process between the *SS_Local* and the remote server *SS_Server*. Whenever the middleman between the client and the server appears, the communication content is encrypted. No unencrypted information is available in the SS traffic to distinguish it from a variety of flows.
- *Transparent Data Transmission*. Studies have shown that most encryption and anonymous systems, such as Tor and VPN, can re-encapsulate user data. We can extract traffic characteristics from the dimensions of packet length, packet arrival time, and packet direction to identify their traffic and achieve better results. However, *SS_Local* forwards client requests transparently in

SS. It creates a new connection when application initials a new proxy connection request, and only converts user's original data into encryption packets with encryption and obfuscation before forwarding them to *SS_Server*, and then does not generate cell packets with specific length. Hence, no obvious flow characteristics are available in an SS flow.

Therefore, It is hard to detect SS only rely on the self-characteristics of a flow. Faced with these challenges, we need mine more useful information from context and external of flows and other relevant data to help identify SS traffic.

B. ANALYSIS OF SS TRAFFIC CHARACTERISTICS AND HOST BEHAVIOR

A client host communicates with the target server through SS when SS enabled. The connections that should be established between the client and the target server will all be changed into connections between the client and the SS Server, which will change the correlation between a flow and its adjacent flows on the flow quintuple attributes and on the time characteristics and alter the connection relationship between the client and the other hosts on flow. The client no longer relies on the IP address obtained from the local DNS responses to access target services, and its behavior on DNS will change too. Therefore, in this section, we will analyze the distinctions between SS flow and the other flows from three aspects: flow context, host behaviors on flows and host behaviors on DNS.

1) FLOW CONTEXT

- Related Definition

For the sake of description, we will define the key concepts as follow.

Definition 1: Bidirectional flows (bi-flow): Bi-flow is an assemblage of all packets that have the same 5-tuple (source address, destination address, source port, destination port, protocol) or reversal 5-tuple (source IP, source port, destination IP and destination port are equal to destination IP, destination port, source IP and source port respectively.) The packets in a bi-flow start with the first SYN packet and end with the last FIN or RST packet. A bi-flow can be expressed as $f = (sip, sport, dip, dport, stime, prot)$. The attributes in f indicate the source IP address, source port, destination IP address, destination port and transport layer protocol in turn. In this paper, we only concern about the analysis of TCP flows because the SS communication is based on TCP protocol. Unless otherwise specified, the "*flow*" used in this paper is a bi-flow. A bi-flow can be thought as the communication between two network terminals. We refer to the end that initiates the flow as the "*source-side*" and the other end as the "*destination-side*". The combination of the destination IP address and the destination port is called the "*destination service*".

Definition 2: Flow correlation. Flow correlation refers to the relationship among multiple flows based on flow identification attributes. From the perspective of the source-side, destination-side and destination service of a flow, we define the following five types of flow correlation according to the different flow attributes between two flows.

Definition 2-1: Flow correlation of sharing source and destination service R_{S_DP}. Let f and g be any two flows, if $f.sip = g.sip$, $f.dip = g.dip$ and $f.dport = g.dport$, then the Flow correlation R_{S_DP} exists between f and g. R_{S_DP} can be used to reflect the behavior of a source-side host requests one same destination service.

Definition 2-2: Flow correlation of sharing source and destination R_{S_D}. Let f and g be any two flows, if $f.sip = g.sip$ and $f.dip = g.dip$, then the Flow correlation R_{S_D} exists between f and g. R_{S_D} can be used to reflect the behavior of a source-side host requests one or more services of the destination.

Definition 2-3: Flow correlation of sharing destination service R_{DP}. Let f and g be any two flows, if $f.dip = g.dip$ and $f.dport = g.dport$, then the Flow correlation R_{DP} exists between f and g. R_{DP} can be used to reflect the behavior of one or more source-side hosts request one same destination service.

Definition 2-4: Flow correlation of sharing destination R_D. Let f and g be any two flows, if $f.dip = g.dip$, then the Flow correlation R_D exists between f and g. R_D can be used to reflect the behavior of one or more source-side hosts request one same destination.

Definition 2-5: Flow correlation of sharing source R_S. Let f and g be any two flows, if $f.sip = g.sip$, then the Flow correlation R_S exists between f and g. R_S can be used to reflect the behavior of a source-side host requests one or more destination or destination services.

There are one or more of the above flow relationships between two flows at the same time. Let $isR_x(f, g)$ be a function indicating whether there is a flow correlation R_x between the flow f and the flow g, as shown in equation (1). Among them, $x = \{S\_DP,\ S\_D,\ DP,\ D,\ S\}$ identifiers representing the five different flow associations.

$$isR_x(f, g) = \begin{cases} 1 & \text{if there is } R_x \text{ between } f \text{ and } g. \\ 0 & \text{if there is not } R_x \text{ between } f \text{ and } g. \end{cases} \tag{1}$$

Let $is\ R_x(f, g)$ be the function that indicates whether there is the flow correlation R_x between f and g. Its expression is shown as (1), where $x = \{S\_DP,\ S\_D,\ DP,\ D,\ S\}$, which is the five flow correlations tags set.

Definition 3: Flow Context. Given a flow f, the Flow Context of f is the collection of all flows that have the five flow correlations in $R = \{R_{S\_DP}, R_{S\_D}, R_{DP}, R_D, R_S\}$ with f in a given time window $[f.stime - w_b, f.stime + w_p]$, where w_b and w_p refer to length of time. Let G^f be the flow context of f, then G^f can be expressed as $G^f = \{G^f_{S\_DP}, G^f_{S\_D}, G^f_{DP}, G^f_D, G^f_S\}$, where $G^f_{S\_DP}$, $G^f_{S\_D}$, G^f_{DP}, G^f_D and G^f_S are the subset of G^f and represent a collection of all flows that have five flow correlations R_{S_DP}, R_{S_D}, R_{DP}, R_D and R_S with f respectively. Take $G^f_{S\_DP}$ as an example, its formalization is as shown in (2).

$$G^f_{S\_DP} = \{g_i | isR_{S\_DP}(g_i, f) = 1,\\ g_i.stime \in [f.stime - w_b, f.stime + w_p], i = 1, 2, 3, \ldots\} \tag{2}$$

Furtherly, we quantify the flow context using the number of flows of per flow correlation subset. Let $FV(G^f)$ be the function of quantifying G^f, then $FV(G^f) = (d^f_{S\_DP}, d^f_{S\_D}, d^f_{DP}, d^f_D, d^f_S)$, where d^f_x indicates the number of elements in G^f_x.

- Flow Context Analysis

When users access the target service using SS, all connections initiated by the host application to the target service are proxied, causing the context of the flow to change. For example, when a user do not use SS to access a web page whose content is provided by a web server, an image server, a video server and an advertisement server, the user host establishes connections with the n target hosts which provide the content of the web page at the same time. For any of the flows generated in the above process, its flow context is mainly composed of flows that have the flow correlation R_S with it, and the number of flows correlation R_{S_DP} with it is limited. Let x_i be the number of concurrent connections between the user and the i-th host. When using SS, the host initializes $\sum_{i=1}^{n} x_i$ connections with SS_Server at the same time. For any of the flows generated in the process, the number of flows that have the flow correlations R_{S_D} and R_{S_DP} with it in its flow context will be higher than that of a flow generated in the no SS process significantly. What's more, when multiple users in the network use the same SS service at the same time, the flows between different hosts to multiple different target services are performed by the SS proxy. So, the flow correlation R_{DP} exists among the formed $\sum_{j=1}^{n_{user}} \sum_{i=1}^{n_j} x_{j,i}$ flows, where $x_{j,i}$ is the number of connections between SS users j and host i, n_j is the number of target hosts requested by user j, and n_{user} is the number of users that connect to the SS_Server. While, when SS is not used, the flows between different hosts and different target servers are irrelevant. Obviously, in this scenario, for any one of these flows, the number of flows that have the flow correlations R_D and R_{DP} with it in its flow context will be significantly higher than that of a non-SS flow.

Therefore, there is a difference between SS flows and non-SS flows in flow context characteristics. SS flows' context quantitative characteristics have higher values on multiple dimensions than non-SS flows.

2) HOST FLOW BEHAVIOR

- Distribution of Destination IP Address

When a host using SS, the network connections that the client should have established with the target server changes to the connections with the SS_Server, which makes the destination of more flows to be aggregated on one host. Hence, there is a certain centralized characteristic in the distribution of destination IP address of flows generated by the hosts that use SS, rather than a decentralized distribution on the hosts that do not use SS. we calculate the entropy of the destination IP address set accessed by a host in a given time to measure the degree of dispersion of the destination IP addresses. Let *dipSet* be the set of the destination IP addresses of flows connected by a host in a given unit time, then $dipSet = \{ip_1, ip_2, \cdots ip_i, \cdots, ip_n\}$, where ip_i represents the i-th IP address. Let P be the probability distribution set of $dipSet$, then $P = \{p_1, p_2, \cdots, p_i, \ldots, p_n\}$, where p_i refer to the probability of ip_i. Hence, the entropy E of the destination IP address set accessed by the source-side host of a flow can be expressed as (3).

$$E = -\sum_{i=1}^{n} p_i \ln p_i \tag{3}$$

The larger the entropy value is, the more decentralized the IP address of the destination host accessed by the host is, the lower the probability that the host will use SS is; the smaller the entropy value is, the more concentrated the IP address of the destination host accessed by the host is, the higher the probability that the host will use SS is. It is found that the entropy of the source host of SS flows is generally smaller than that of the non-SS flows in the 5-minute time window from the experiment in Section V.C.1). So, the entropy can be used to represent the flow behavior of the source-side host of SS flows.

- Time Characteristic Analysis of Multiple Flows

Applications usually limit the number of concurrent flows per user to prevent service resources from being maliciously occupied by a few users. This mechanism limits the number of concurrent flows between the source-side host and the target server. When the SS service enabled, the multiple concurrent flows initialized by the user to connect different services will be turned into one set of concurrent flows which belong to only SS service. In order to analyze the temporal concurrency of multiple flows between the source-side host and the target service, based on the idea of *Bursty Traffic* metrics [45], we propose the concept of *Flow Burst* and use its features to characterize the source-side host behavior of a flow.

Definition 4: Flow Burst. Suppose that a source-side host initialize multiple flows with target service in a given time window. After sorting these flows by time, if there are at least consecutive N flows whose start time interval is less than the given threshold T, a *Flow Burst* is formed. *Burst Length* is the number of flows contained in a *Flow Burst* and then the minimum value of *Burst Length* equal to N.

In a given time window, multiple Flow Bursts may be formed between the source-side host and the target service. The number of Flow Bursts is called Burst Number (*bnum*), the max of Flow Burst length is called Max Burst Length (*maxBLen*), the sum of all Burst Length is called Total Bust Length (*totalBLen*), and the ratio of Total Bust Length to Burst Number is called Average Burst Length (*avgBLen*). Therefore, given a flow f, its source-side host behavior on flow can be represented as $B(f) = (bnum, maxBLen, totalBLen, avgBLen)$.

We will see that in subsection V.(C).1, when N = 3 and T = 100ms, the values of these features of SS flows are generally higher than that of non-SS flows and these behavioral features of the source-side host of a flow can be used to distinguished between SS flows and non-SS flows.

3) HOST DNS BEHAVIOR

- Sensitive Domain Name Request Behavior Analysis

SS provides PAC (Proxy Auto-Config) and Global modes. In Global mode, all sites requested by users are accessed through proxy. In PAC mode, only the sites whose domain are appeared in the given PAC configuration list are proxied by SS. Usually, the configuration list consists of domain names that are restricted by local security policies. We name these domain names as *Sensitive Domain Names*. Since SS servers are usually deployed outside firewalls and network bandwidth is limited, if the global mode is used, all user requests are proxied by SS, including local services, which will lead to slow service access and a poor user experience. Therefore, PAC mode is widely used to get access to restricted network resources.

By analyzing SS traffic, we find that there is a DNS leakage problem [46] when SS is working in PAC mode or some versions' Global mode. When application uses SS to connect the target server, the local DNS server can see what domain name the user is asking for. Obviously, the more sensitive domain name requests are made in a user's DNS requests, the greater the probability that the user will use SS.

- Unassociated Domain Request Behavior Analysis

Generally, application sends DNS request to obtain the IP address of the target host before accessing the target service. Under most circumstances, After an application requests a domain name (for example, *www.example.com*) and get all IP addresses mapped by this domain, it will establish a connection with one of the IP addresses (mostly the first IP address [47]) in a very short time. Figure 2 shows the process and the relationship between a DNS response and its corresponding flows.

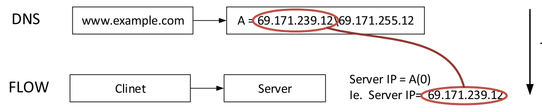


FIGURE 2. The relationship between domain name requests and flows.

SS is a transparent proxy. Since the existence of DNS leakage problem, all domain names requested by user in the process of accessing target services can be seen in local DNS server, but user do not initialize any connection to the host of target services. Therefore, we call these domain names as *Unassociated Domain Names*. Apparently, the more unassociated domain names the host requests, the higher the frequency of domain name leaks and the greater the probability of using SS is.

IV. DETECTION METHOD

In this section, we introduce the framework of our proposed SS traffic identification method firstly, and then detail the implementation of each part of the framework.

A. DETECTION METHOD FRAMEWORK OVERVIEW

Figure 3 shows the overview of our proposed SS traffic detection method, including three stages: data preprocessing, feature extraction and SS traffic detection. In the stage of data preprocessing, data of flows and DNS in a given time window is read from network or offline data. Then a set of flow association data is formed based on flow correlations under different time parameters in the Flow Correlator, a set of DNS behavior data of source host is formed in the Flow & DNS Correlator based on the time sequence relation between the host requesting domain names and host initializing flows. In the feature extraction stage, flow feature vectors are formed by extracting and computing flow features. In this process, the quantitative features of flow context and the flow behaviors features of the source-side host of a flow extracted from the flow context data, and DNS behavior features of the source-side host abstracted from DNS behavior data of the source-side host. In the SS traffic detection stage, based on the flow feature vectors, the Random Forest algorithm is used to construct classifier to detect SS flow. The whole framework is designed and implemented on the big data platform based on Hadoop and Spark. In the next subsection, we will describe the three processes of data preprocessing, feature extraction, and model construction in detail.

B. DATA PREPROCESSING

1) FLOW CORRELATION

The function of the Flow Correlator is to form a subset of neighbor flows for each flow based on different flow correlations between flows with different time parameters, which provides a basis for the features of flow context and host behaviors extraction. In order to fully reflect the neighborhood relationship of a flow, we proposed a data extraction method of looking forward and backward in time axis. The detail of this method is as follows. The neighbor flow subset of f is constructed on the set of adjacent flows of f, which consists of all flows in the w_b time window before the start time of f and in the w_p time window after the start time of f. That is, the neighbor flows of f consist of all flows that are in the time-zone $(f.stime - w_b, f.stime + w_p)$ (usually $w_b = w_p$). We use the data processing mechanism based on sliding window to realize flow correlation. Suppose that the time window size of data to be processed each time is w_{f_deal}, the sliding window is formed by forwarding w_p and backing w_b on the basis of the time window of data in order that each flow in the window can obtain all adjacent flows in the front and rear time windows. That is to say, the final sliding window size is $w_{flow} = w_{f\_deal} + w_b + w_p$. Then set the size of sliding step of the sliding window as $b = w_{f\_deal}$ to achieve non-overlapping processing of all flows. The process

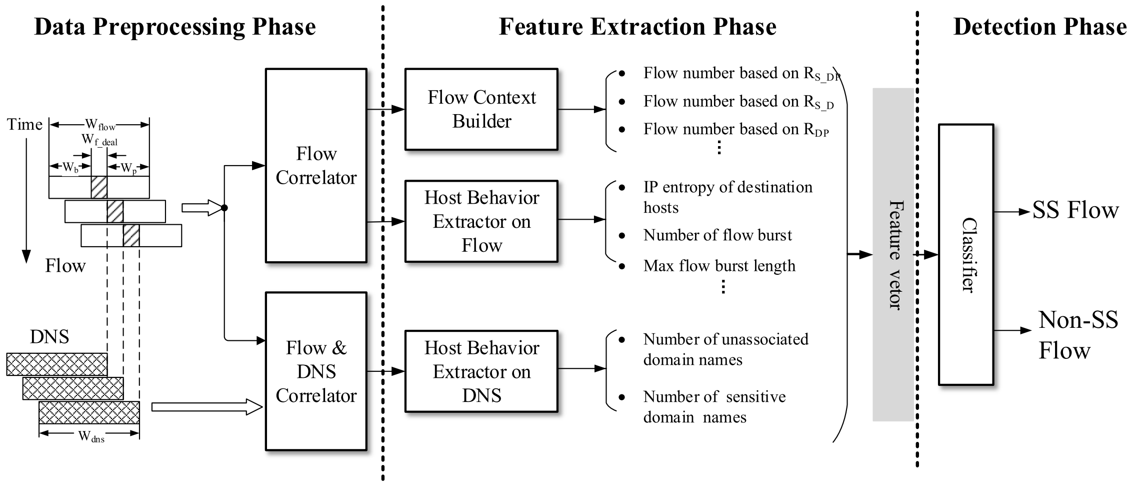


FIGURE 3. SS traffic identification framework overview.

is shown in the Data Preprocessing Phase of Figure 3, where the shaded part of the slashes tagged by w_{f_deal} is the data block of each flow to be processed. Algorithm 1 describes the process of calculating the flow context characteristics while flow association. Given the Flow data $FlowData\_W$ in any one sliding window, the flow data to be processed is get according to the time filter parameters (line 2), and then each flow is processed (lines 3-7). Next, the set of adjacent flows (line 4) is obtained through the designed filter process, and then the number of flows (lines 5-7) under different correlations is counted according to the flow Association rules, forming the flow context feature set (line 8).

2) FLOW AND DNS CORRELATION

Flow & DNS Correlator provides a basis for the feature extraction of source-side host DNS behavior of flows. According to the mapping relationship between DNS behavior and flow behavior of a host, it forms a list of pairs between domain name and server IP address requested by source hosts in a given time window through association. Because of the caching mechanism of DNS system, for a given flow, we use DNS requests from the host to correlate with the flow for time w_{flow} before the flow is initialized. The process diagram is shown in the Data Preprocessing Phase in Figure 3. We use the correlation method of flows and DNS based on different sliding window sizes proposed in our previous work to realize the correlation between them. First, we read the data of flows in the time window w_{flow} and get the data of DNS in the time window w_{dns}. Then we enter the next data reading window with the same sliding step size. In order to unify the processing mechanism of flow data in flow association, the end time of each read DNS data block is same as that of w_{f_deal}. The association method between Flow and DNS data is shown in line 1-5 of algorithm 3. More detailed information will be introduced in Section IV.C.3).

C. FEATURE EXTRACTION

Based on the analysis of Section III.(B), 12 features were extracted from three aspects: flow context, behavior of source-side hosts on flow and DNS behavior of source-side host on DNS, and then they are be used to construct the detection model. They are all shown in Table 1. The extraction and calculation methods of these features are described in the next subsection.

Algorithm 1 The Computation of Flow Correlation and Flow Context Features

```
Input: $FlowData\_W$, $w_b$, $w_p$, $w_{f\_deal}$
Output: Flow Context Features
1: Let $t\_start$ be the starttime of $FlowData\_W$
2: flowContext = $FlowData\_W$.filter($f$ => $t\_start + w_b < f.stime < t\_start + w_b + w_{f\_deal}$)
3:   .map($f$ => {
4:     neighborFlows = $FlowData\_W$.filter($g$ => $isR_r(f, g) == 1$)
5:     For r in ($S\_DP, S\_D, DP, D, S$)
6:       $d_r$ = neighborFlows.count($g$ => $isR_r(f, g) == 1$)
7:     EndFor
8:     ($d_{S\_DP}, d_{S\_D}, d_{DP}, d_D, d_S$) })
```

1) FEATURES OF FLOW CONTEXT

Flow context features are formed by calculating the quantitative value of the flow context. That is, based on a set of adjacent flows of a flow obtained in the previous steps, the flow context is formed according to definition 3, and then the flow context feature set is quantized. The detailed calculation process is shown in algorithm 1 (line 5-8).

2) SOURCE-SIDE HOST BEHAVIOR FEATURES ON FLOW

The source-side host behavior features on flow include the entropy of the destination IP accessed by the source-side host of the flow and four features of the Flow Burst of the host based on R_{S_DP}, and their calculation process is shown in algorithm 2. The algorithm takes all flows in one sliding window as input, which is denoted by the symbol $FlowData\_W$. After obtaining the flow data to be processed

TABLE 1. The list of features proposed in our method.

Feature Category	No	Features	Description
Flow Context	1	d_{S_DP}	The number of flows having flow correlation R_{S_DP} with the flow.
	2	d_{S_D}	The number of flows having flow correlation R_{S_D} with the flow.
	3	d_{DP}	The number of flows having flow correlation R_{DP} with the flow.
	4	d_D	The number of flows having flow correlation R_D with the flow.
	5	d_S	The number of flows having flow correlation R_S with the flow.
Host Behavior on Flow	6	E	The entropy of destination host IP address accessed by the source-side host of the flow.
	7	$bnum$	The number of Flow Burst.
	8	$maxBLen$	The max of Flow Burst length.
	9	$totalBLen$	The sum of all Flow Burst length.
	10	$avgBLen$	The ratio of $totalBLen$ to $bnum$.
Host Behavior on DNS	11	$sensDNum$	The number of sensitive domain names requested by the source-side host of the flow.
	12	$unAssDNum$	The number of unassociated domain names requested by the source-side host of the flow.

Algorithm 2 The Computation of Source-side Host Behavior Features on Flow

```
Input: $FlowData\_W$, $w_b$, $w_p$, $w_{f\_deal}$, $minFlowTime$, $minBurstLen$
Output: Source-side host behavior features on flow
1:  Let $t\_start$ be the starttime of $FlowData\_W$
2:  flowContext = $FlowData\_W$.filter($f$ => $t\_start + w_b$
        $< f.stime < t\_start + w_b + w_{f\_deal}$)
3:  .map($f$ => {
4:   bFlows = $FlowData\_W$.filter($g$ =>$isR_{S\_DP}(f, g) == 1$)
        .sortby(stime)
5:  flowsInterval = flowsTimeInterval(bFlows)
6:  loc=0
7:  For $inv\_i \in$ flowsInterval Do
8:      If $inv\_i > minFlowTime$
9:          bLen = i - loc; loc = i;
10:         If bLen >= $minBurstLen$ Then
11:             $bBuf$.append(bLen)
12:         EndIf
13:     EndIf
14: EndFor
15:  $IPEntropy$ = Entropy($FlowData\_W$.filter($g$ =>
        $isR_S(g, f) == 1$).map((dip)))
16: ($IPEntropy$,$bBuf$.size,$bBuf$.max,$bBuf$.sum,$bBuf$.avg)
    })
```

(line 2), the Flow Burst features of the source-side host of each flow are calculated. The following is the detailed processes. Firstly, all flows satisfying flow correlation R_{S_DP} with the current flow are filtered out from the sliding window and sorted according to the start time of the flow (line 4). Then we compute all Flow Burst features according to Definition 4. and store them in a list *bBuf* (lines 6-14). Finally, the four features of Flow Burst are calculated. Line 15 is the calculation process of IP entropy.

3) SOURCE-SIDE HOST BEHAVIOR FEATURES ON DNS

The source-side host behavior features on DNS include two features: the number of sensitive domain names and the number of unassociated domain names requested by the source-side host of a flow. The calculation process is shown in algorithm 3. The algorithm takes all flows denoted as $FlowData_w$ in one sliding window w_{flow} and all dns data denoted as *DNSData* in one sliding window w_{dns} as inputs. Firstly, the algorithm handles $FlowData_w$ to obtain the flow data to be processed according time filter parameters (line 1-2), and then transformes them into a (*Key*, *Value*) format with the source IP address (*clientIP*) as the keyword (line 3). Next, in line 4, *DNSData* are also transformed into the (*Key*, *Value*) format, the *Key* of which is the IP address of the client host in dns response data and the *Value* of which consists of the requested domain name and the first server IP address obtained, namely (*clientIP*, (*domain*, *serverIP*)). Then correlation calculation on the two processed data are performed to form a (*Key*, *Value*) pair with *clientIP* as the keyword and in form of (*clientIP*,(*flowDataList*,*dnsDataList*) (line 5), where *flowDataList* refers to the *Value* list of *Flows_Processed* and *dnsDataList* refers to the *Value* list of *DNS_Processed*. Then the number of sensitive domain names (line 7) and the number of unassociated domain names (lines 8-13) requested by the source-side host of the flow are calculated based on the associated results. Sensitive domain names are determined if the domain names in the *dnsDataList* corresponding to each *clientIP* exist in the *pacList* (generated according the default PAC configure file in SS software) and counting the number of existing domains. The calculation of unassociated domain names first requires the destination host IP address set *dstIPCon* requested by the client (line 8-11), and the first server IP address set *dstIPDns* obtained by the client requesting the domain name (line 12). Then we calculate the number *unAssDNum* of elements in *dstIPDns* that do not appear in *dstIPCon*.

D. DETECTION MODEL

The Random Forest Algorithm is a classification algorithm which contains multiple decision trees in a random way. It has strong generalization ability and is easy to be implemented in parallel. Compared with other algorithms, it has a good performance on many data sets. Therefore, we select the random forest algorithm to construct SS traffic classifier on big data platform based on the 12-dimensional feature vectors proposed in Section IV.(C).

Algorithm 3 The computation of Host DNS Behavior Features

```
Input: $FlowData\_W$, $DnsData$, $pacList$, $w_{f\_deal}$, $w_{dns}$
Output: Source-side host behavior features on DNS
1:  Let $t\_start$ be the starttime of $FlowData\_W$
2:  Flows_Processed = $FlowData\_W$.filter($f$ =>
        $t\_start + w_b < f.stime < t\_start + w_b + w_{f\_deal}$)
3:      .map(data =>($clientIP$,(data)))
4:  DNS_Processed = $DnsData$.map(data =>
        ($clientIP$, ($domain$, $serverIP$))
5:  DNSBehavior = Flows_Processed
        .cogroup(Dns_Processed)
6:  .flatMap($t$ =>{
7:  sensDNum =t.dnsDataList.count( isSensitive(domain,
        $pacList$)==True)
8:  dstIPCon = $FlowData\_W$
9:      .filter($f$ =>$t\_start + w_b + w_{deal} - w_{dns} <$
          $f.stime < t\_start + w_b + w_{f\_deal}$)
10:     .filter($f$ => ($f.clientIP == t.clientIP$))
11:     .map($f$ => ($f.dip$))
12: dstIPDns = t.dnsDataList.filter(dns =>
        t.stime$-w_{f\_deal}$ < dns.time ).map((serverIP))
13: $unAssDNum$ = dstIPDns.diff(dstIPCon).length
14: ($sensDNum$, $unAssDNum$) })
```

V. EXPERIMENT AND RESULT ANALYSIS

A. DATA SET GENERATION AND EXPERIMENTAL ENVIRONMENT.

To create a representative SS traffic data set, we captured the real traffic generated by our lab members. We created a virtual machine respectively on each of two different large public cloud platform outside the firewall, and deployed SS service on it. We installed and configured the SS client (version. 4.1.0 or 4.1.2) on hosts of 10 members. These hosts are not only used to daily study and access Internet, but also intermittently access a variety of applications and services outside the firewall through SS server. The logical topology structure of the experimental environment and the location of experimental data acquisition points are shown in Figure 4. The flow data was collected from the boundary network devices of our laboratory. The bidirectional flows of TCP were reconstructed while traffic data were collected. Each flow includes attributes such as source IP, source port, destination IP, destination port, start time and protocol. the DNS data collected from the server area of the campus network. Three DNS servers are deployed in the area, providing domain name resolution service for campus users including our laboratory. Each DNS record includes source IP, source port, destination IP, destination port and all header fields defined in the DNS protocol.

We use the flow data and the DNS data collected between Oct. 20, 2018 and Nov. 30, 2018 for experiments. The Raw samples were formed by the data preprocessing and features extraction methods described in Section IV. Then, these

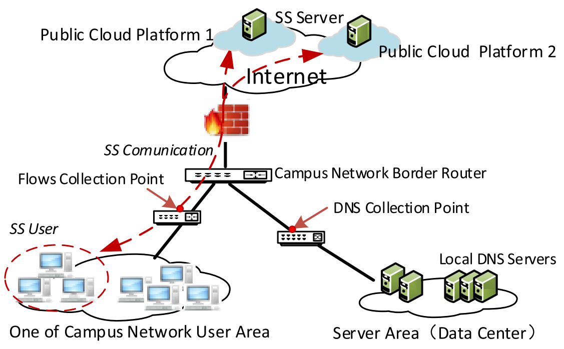


FIGURE 4. The logistic topology diagram of experimental network.

samples were marked according to the SS server IP and port number. The flows, which match the destination IP address and port number with the IP address and port number of the SS server deployed in the experiment, were labeled as SS flows, and the rest were labeled as Non-SS flows. The details of the data set are shown in Table 2. The DNS part in the table lists the total number of DNS records extracted and the number involved in the sample data calculation process. The flow samples are composed of the samples labeled as SS flows and the background traffic samples are randomly extracted from the host traffic that never used SS. Considering the problem of sample balance, the number of the adopted background flows is roughly 2-folds of the number of SS flows samples. Since massive traffic data in a long period of time needs to be processed and correlated in our experiments, all researches and experiments were conducted on NTCI-BDP, a big data analysis platform established by authors' team. Based on Hadoop and Spark, the platform is configured with 1 master node and 14 data nodes, and the communication bandwidth between nodes is 10 Gbps. All experimental data is formatted according to the protocol and stored on HDFS in Parquet format and all algorithms are implemented with Spark and Spark MLib.

B. EVALUATION INDICES

In order to evaluate the performance of SS classifier, we use Accuracy (Ac), Precision (Pr), Recall (Rc), and F1 score ($F1$) as evaluation indices. The formula of the above indicators is shown as (4)-(7), Where TP, FP, FN and TN represent true positive, false positive, false negative and true negative respectively.

$$Ac = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$Pr = \frac{TP}{TP + FP} \tag{5}$$

$$Rc = \frac{TP}{TP + FN} \tag{6}$$

$$F1 = \frac{2 \cdot Pr \cdot Rc}{Pr + Rc} \tag{7}$$

TABLE 2. The details of the experimental data set.

Data type	# of all flows	# of SS flows	# of adopted background flows	# of all DNS records	# of all DNS records in the area
Amount	4,509,512	81,474	136,623	8,300,303,405	7,626,742

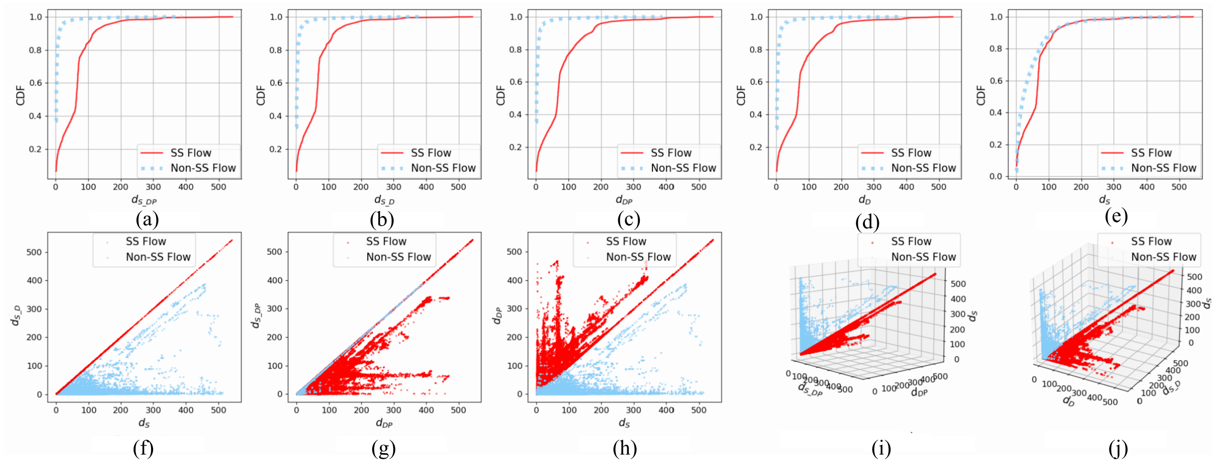


FIGURE 5. The distribution diagrams of flow context features.

C. EXPERIMENT AND ANALYSIS

1) ANALYSIS OF FLOW CONTEXT AND HOSTS' BEHAVIOR

In order to verify the results of the analysis of SS traffic context characteristics and host behavior characteristics in section III.(B), in this section, we verify the discrimination of each one-dimensional feature proposed in this paper to SS flows on the real-world network data set described in section V.(A).

- Flow Context

We analyze the flow context characteristics of SS and non-SS flows on the experimental data set. For any flow f, Set the time window for forming the flow context as $[f.stime - 15s, f.stime + 15s]$, which means $w_b = w_p = 15s$. In Figure 5, (a) - (e) is a comparison of Cumulative Distribution Function (CDF) of quantization vectors $FV(G^f)$ of SS and non-SS flows in each dimension of flow context. As we can see, SS flows and non-SS flows have different distribution characteristics in the four dimensions of d_{S_DP}, d_{S_D}, d_{DP} and d_D. About 85% of non-SS flow values in these four dimensions are less than 15, about 65% of SS flow values in the [15, 200] range, and SS flow values are generally higher than non-SS flow values. The results are consistent with our analysis in section III.(B).1). It is worth nothing that Figure 5 (e) shows that the range and distribution of values of SS and non-SS flows on d_S are consistent, which means that the use of SS does not change the characteristics of the number of connections initiated by hosts.

We also conducted several experiments and analyzed the relationship between SS flow and non-SS flow in multiple dimensions of the flow context, to further verify the difference between them in the flow context features. Figure 5 (f) - (h) shows the sample scatter distribution of SS flow and non-SS flow in two-dimensional feature space: (d_S,d_{S_D}), (d_{DP},d_{S_D}) and (d_S,d_{DP}). The mapping relationship between d_S and d_{S_D} reflects the relationship between the number of flows which are initialized by source hosts and destination host, and the number of all flows which are initialized by source hosts. Figure 5 (f) shows that the sample points of SS flows are mainly distributed on d_{S_D}=d_S and a small number of samples are distributed below d_{S_D}=d_S, which indicates that the source hosts only establish connection with SS server within 30 seconds of requesting SS server. Most of the samples of non-SS flows are distributed below d_{S_D}=d_S. With the increase of d_S, the value of d_{S_D} is more distributed in [0,30], which indicates that connections are also established between the source host and other hosts within 30 seconds. The larger the number of connections between a source host and destination hosts are, the fewer connections between a source host and other hosts are established. The mapping relationship between d_{DP} and d_{S_DP} reflects the relationship between the total number of connections that the destination service is accessed and the number of connections accessed by the source host of the current flow. Figure 5 (g) shows that both SS flow and non-SS flow samples are distributed in the region above and below d_{DP}=d_{S_DP}, but the non-SS flows are more concentrated near the origin of the coordinate axis, and SS flows are scattered in the larger space of this region. This result shows that SS servers produce more flows than non-SS servers in most cases. The mapping relationship between d_S and d_{DP} reflects the relationship between the number of connections initiated by the source-side host and the number of connections connected by the destination of a flow. When $d_S < d_{DP}$, it indicates that there are connections between the destination service and other hosts besides the current source, which means that the destination service is accessed by multiple different hosts at the same time. When d_S=d_{DP}, it means that the source host only establishes connections with the destination service, and the destination service is only accessed by the current source. When $d_S > d_{DP}$, it means that the source host accesses other hosts or services in addition to the destination service. As we can see from Figure 5 (h), SS flows are all concentrated in the area of $d_S \leq d_{DP}$, while non-SS flows are mainly concentrated in the area of $d_S > d_{DP}$ and a few in the area of $d_S \leq d_{DP}$. The result of Figure 5 (h) indicates that SS services are accessed by one host or more different hosts at the same time, while the source hosts of non-SS services establish connections with multiple hosts or services at the same time. In Figure 5 (j)and (k) are sample scatter distributions in three-dimensional space: (d_{S_DP}, d_{DP}, d_S) and (d_D, d_{S_D}, d_S). As can be seen from the figures, there is a clear spatial interface between SS flow and non-SS flow. In a word, we can conclude that SS flow and non-SS flow are distinguished in the quantization vectors of flow context.

- Analysis of Host Flow Behavior

Destination IP Addresses Entropy. According to Eq. (3), we analyze the distribution of destination IP address entropy accessed by source host of flows on the experimental data set. Figure 6 shows that there are differences between SS flows and non-SS flows in destination IP address entropy distribution of flows that initiated by the source hosts. 80% of the IP entropy of the destination host requesting from the source side of SS flow is in [0, 2], while only 15% of the destination IP entropy of non-SS flow is in (0,2) and 85% is in [2, 5]. From the above, the IP addresses of destination hosts during the use of SS have obvious centralized characteristics.

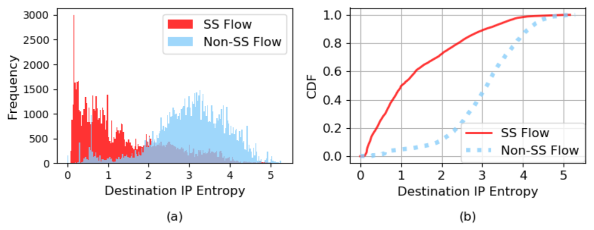


FIGURE 6. The distribution diagrams of the entropy of destination IP addresses.

Flow Burst Features. We analyzed the flow behavior of source-side hosts for SS and non-SS flows on the experimental data set. Here we define the minimum Burst length N = 3 and the time interval threshold T = 100ms. As the results are shown in Figure 7, About 80% of the source-side hosts of non-SS flows do not form flow Burst, while 90% of the source hosts of SS flows form flow Burst with $bnum \in [0, 100)$. Different dimensional distributions of source host behaviors of non-SS and SS flows are different. In the three-dimensional feature space, we further observed the distribution of the source-side host flow behavior of SS flows and non-SS flows. As we can see from (e) and (f) in Figure 7, non-ss flows are relatively concentrated near the origin in two different three-dimensional features spaces, while SS flows are dispersed in a larger space. Obviously, there is a certain distinction between SS flows and non-SS flows in multi-flow time characteristics based on the source-side hosts of flows and target services.

- Analysis of Host DNS behavior

On the experimental data set, we analyzed the difference between the source-side host of an SS flow and that of a non-SS flow in the number of request sensitive domain names and the number of request unrelated domain names. Figure 8 (a)-(c) are the distribution diagrams of the number of user requests for sensitive domain names in the past 5 minutes before the initialization of the flow for the source-side hosts of the SS flow and that the non-SS flow (the sample order has been randomized). The three figures report that the two distributions are different, and the value range of SS flows is in [0, 500], while 90% of the value of non-SS flow is distributed in (0,50). The results show that the source-side host of a non-SS flow will hardly request sensitive domain names. Even if there are a few sensitive domain names accessed, the request amount is much lower than the source-side hosts of SS flows. From the above, we can conclude that the source hosts of SS flows and non-SS flows have distinction in sensitive domain name request behaviors.

In Figure 8 (d)-(f) are the distribution of the number of unassociated domain names of the source-side host of an SS flow and that of a non-SS flow. We can learn that the distribution of the two is different. SS flows have a wider range of values, ranging in [0,700], and there are more samples distributed in [100,400]. More than 80% of non-SS flows are 0, and the rest are mainly distributed in (0,100). The results indicate that there is a mapping relationship between domain name requests and popular behavior in source hosts of non-SS flows in most cases while there are a lot of requests for unassociated domain names in the source host of SS flows. In a word, the source-side hosts of SS flows and non-SS flows differ in behaviors of requesting unassociated domain names.

The above experimental results show that SS flows are different from non-SS flows in terms of flow context characteristics, host behavior and host DNS behavior, and these kinds of features can be used to identify SS traffic.

2) COMPARISON AND ANALYSIS ON DIFFERENT FEATURE SETS

In order to analyze the effect of the features extracted from different dimensions on the detection results, we carries out experiments based on the three dimensions of flow context, host flow behavior and host DNS behavior, and compares the three features and their results on evaluation indicators Ac,Pr,Rc and $F1$. According to the analysis of SS traffic characteristics and host behavior in Section III.(B), we set the flow context characteristic time window parameter $w_b = w_p = 15s$ and host flow characteristic time window parameter $w_b = w_p = 2.5min$. As for host DNS behavior, the extraction time window size of DNS data is 15 minutes.

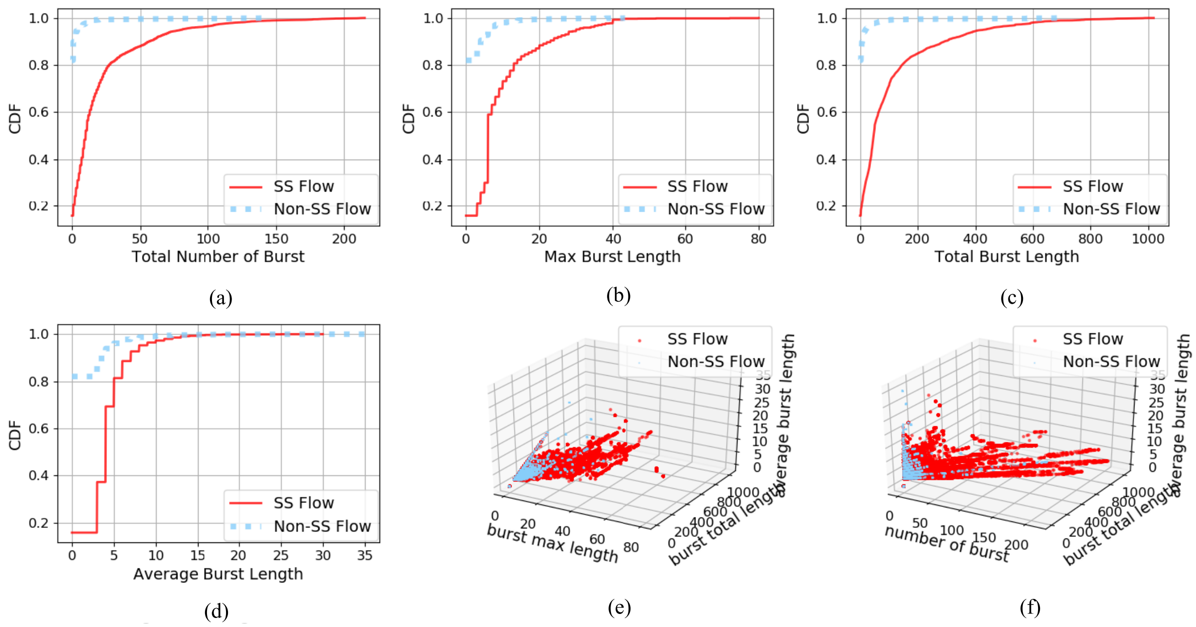


FIGURE 7. **The distribution diagrams of flow context features.**

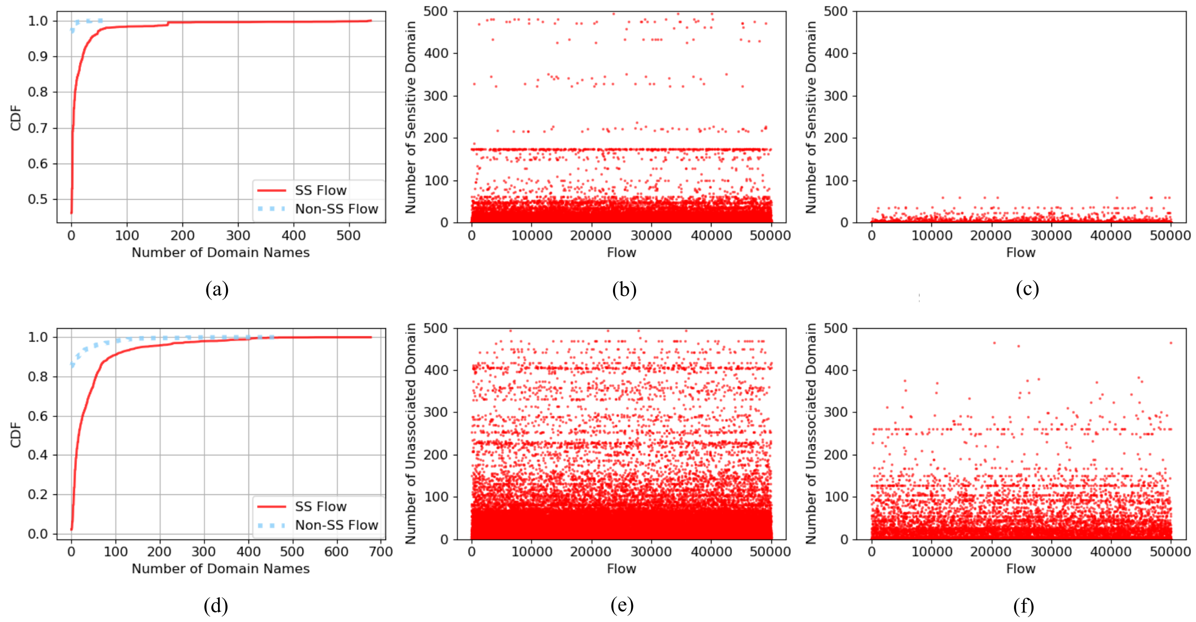


FIGURE 8. **The distribution diagrams of host behavior on dns.**

We use the Random Forest method to implement the classifier. The sample data set is divided into 7:3 scales. Seven parts are used as training sample set and the remaining three are used as test sample set. Repeat the process ten times, and finally measure the classification performance based on the average value of each evaluation index.

The experimental results are shown in Table 3 and the comparison is shown in Figure 9 (a). From the results, we can see that SS traffic can be identified to a certain degree based on the characteristics of flow context and host flow behavior, and it can reach more than 80% in several evaluation indicators. The detection performance of flow context features is slightly better than that of host flow features because the host flow features are 1.69% higher in accuracy than the flow context features, but 4.01% lower in recall rate, which ultimately leads to better performance of up-flow context feature than host flow feature in $F1$. The combination of flow context features and host flow features can significantly improve the

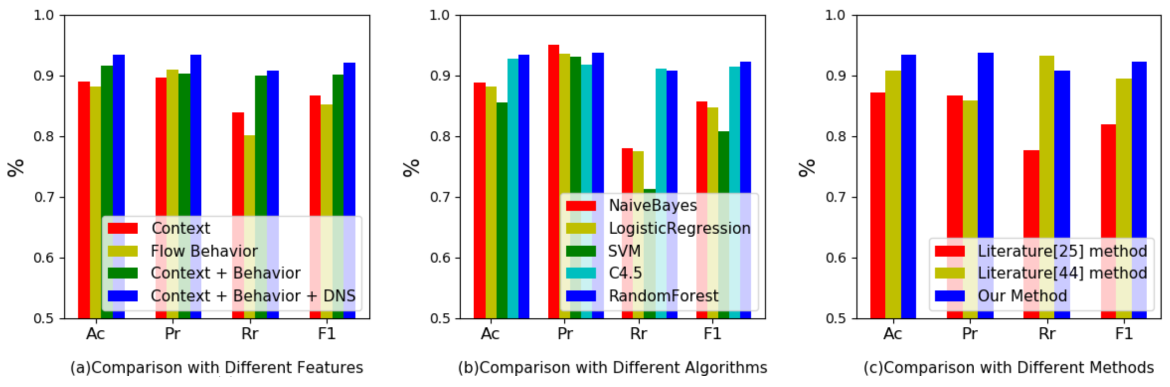


FIGURE 9. **The distribution diagrams of flow context features.**

TABLE 3. **The comparison results on different feature sets.**

Features	Ac	Pr	Rc	$F1$
Flow Context	88.95%	89.54%	83.89%	86.62%
Host Flow Behavior	88.18%	91.23%	79.88%	85.18%
Flow Context + Host Flow Behavior	91.61%	90.33%	89.92%	90.12%
Flow Context + Host Flow Behavior + Host DNS Behavior	**93.43%**	**93.74%**	**90.66%**	**92.17%**

TABLE 4. **The comparison results on different classification algorithm.**

Features	Ac	Pr	Rc	$F1$
Naive Bayesian	88.83%	95.02%	77.91%	85.62%
Logistic Regression	88.09%	93.47%	77.52%	84.75%
SVM	85.53%	93.09%	71.21%	80.70%
C4.5	92.68%	91.72%	**91.13%**	91.34%
Random Forest	**93.43%**	**93.74%**	90.66%	**92.17%**

ability to recognize SS traffic, especially in the recall rate, which is 6.01% higher than that of using flow context features alone and 10.16% higher than that of using host flow behavior alone. Besides, the combined features increased by nearly 5% in $F1$. After further adding the DNS behavior characteristics of hosts, the evaluation indexes were significantly improved, which were 1.82%, 3.41%, 0.54% and 2.05% respectively. Therefore, it is reasonable to combine three kinds of features to construct SS traffic classifier.

3) COMPARISON AND ANALYSIS ON DIFFERENT CLASSIFICATION ALGORITHMS

In order to verify the validity of the proposed feature vectors, we construct classifiers using Naive Bayesian, Logical Regression, SVM, C4.5 and Random Forest respectively. We compare the results on evaluation indicators: Ac, Pr, Rc and $F1$. The experimental method is the same as the previous experiments. The sample data set is divided into 7:3 scales, and the process is repeated ten times. Finally, the classification performance is measured based on the average value of each evaluation index.

The experimental results are shown in Table 4 and the results are compared as shown in Figure 9 (b). The experimental results show that the proposed feature performs well in the classifier constructed by several algorithms. Most evaluatiion indices' values are higher than 80%. The performance of tree structure-based classifier is better than that of other methods, and each index reaches more than 90%. Furthermore, the performance of random forest algorithm is the best, with the accuracy of 93.43% and the $F1$ of 92.17%.

4) COMPARISON WITH OTHER METHODS

In order to verify the validity of the method, we compare it with the classification methods proposed in literature [44] and literature [25]. There are two main reasons we choose these two methods for comparative experiments. First, Literature [44] is the only method for identifying SS traffic as far as we know. Second, in literature [25], a traffic classification method based on flow extension vector was proposed based on the relationship between flows, which was applied to traffic classification including HTTP and HTTP proxy. There is no SS traffic public data set at present and the authors of literature [44] have not published the SS data set used in their literature. Therefore, we carried out a comparative experiment on the data set. The experimental method is the same as the previous experiments. The experimental results are shown in table 5, and the comparison of experimental results are shown in Figure 9 (c).

TABLE 5. **The comparison results on different methods.**

Features	Ac	Pr	Rc	$F1$
Literature [25]	87.16%	86.58%	77.68%	81.83%
Literature [44]	90.66%	85.87%	**93.28%**	89.42%
our Method	**93.43%**	**93.74%**	90.66%	**92.17%**

It can be seen from Table 5 and Figure 9 (c) that the traffic classification method based on flow extension vector proposed in literature [25] has only 87.16% accuracy and 77.68% recall rate in distinguishing SS traffic. Therefore, the methods only relying on the correlation between flows cannot effectively identify SS flows. The identification accuracy of the method proposed in the literature [44] for SS traffic is basically consistent with the experimental results in the literature (the accuracy greater than 85%). The detection method proposed in this paper has the best performance in three methods. The accuracy, precision and accuracy are all good, which are 2.77%-5.27%, 7.16%-7.87% and 2.75%-10.34% higher respectively. Although the recall rate of the proposed method is slightly lower than that of reference [44], the accuracy of the method is 7.87% and 2.75% higher, and the accuracy of SS traffic identification is better. In addition, because our method only used flow-level data, it did not need the distribution characteristics such as the arrival time of data packets in the flow, and not need to retain the information of data packets in flows in the process of data acquisition. So our method is more suitable for the application in large-scale network environment. From above, the method we proposed is effective in identifying SS traffic and has more advantages in accuracy and application range.

VI. CONCLUSION

In this paper, we proposed a method to detect SS traffic based on flow context and host behavior. This method can alleviate the abuse of cloud VPS services by SS users effectively and improve the security of cloud computing platforms. It can also help network administrators to identify malicious network activities using SS and improve network security management capabilities. It is worth mentioning that our proposed method making full use of the information contained in flow contexts and host behaviors and performed the detection of SS traffic on flow-level. Moreover, as far as we know, the features extracted from them, such as the four features about flow burst based on inter-flow relations and the feature of unassociated domain names based on DNS and flow association, are innovatively proposed and applied to flow classification methods by us. Besides, since the method does not rely on the detail distribution characteristics of packets in flows, it is more suitable for large-scale network environment. The experimental results on dataset collected from the real network environment yielded significant improvements to the state-of-the-art methods, validating the effectiveness our proposed method based on flow context and host behavior.

ACKNOWLEDGMENT

The authors would like to thank the editors and anonymous reviewers for their valuable suggestions that will considerably help us to enhance this paper.

REFERENCES

[1] J. Jayson and K. Curran, "Detecting anonymous proxy usage," *Recent Adv. Commun. Netw. Technol.*, vol. 6, no. 1, pp. 55–70, 2017.

[2] G. He, M. Yang, J. Luo, and X. Gu, "A novel application classification attack against Tor," *Concurrency Comput., Pract., Exper.*, vol. 27, no. 18, pp. 5640–5661, Jul. 2015.

[3] H. Zhu, R. Lu, X. Shen, and X. Lin, "Security in service-oriented vehicular networks," *IEEE Wireless Commun.*, vol. 16, no. 4, pp. 16–22, Aug. 2009.

[4] M. Clowwindy and L. Max, *Shadowsocks*. Accessed: Aug. 19, 2018. [Online]. Available: http://www.shadowsocks.org/

[5] X. Li, Q. Wang, X. Lan, X. Chen, N. Zhang, and D. Chen, "Enhancing cloud-based IoT security through trustworthy cloud service: An integration of security and reputation approach," *IEEE Access*, vol. 7, pp. 9368–9383, 2019.

[6] P. Velan, M. Cermák, P. Čeleda, and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," *Int. J. Netw. Manage.*, vol. 25, no. 5, pp. 355–374, Sep./Oct. 2015.

[7] D. Chen, N. Zhang, R. Lu, N. Cheng, K. Zhang, and Z. Qin, "Channel precoding based message authentication in wireless networks: Challenges and solutions," *IEEE Netw.*, vol. 33, no. 1, pp. 99–105, Jan./Feb. 2019.

[8] D. Chen *et al.*, "An LDPC code based physical layer message authentication scheme with prefect security," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 4, pp. 748–761, Apr. 2018.

[9] D. Chen, N. Zhang, N. Cheng, K. Zhang, Z. Qin, and X. S. Shen, "Physical layer based message authentication with secure channel codes," *IEEE Trans. Depend. Sec. Comput.*, to be published.

[10] D. Chen, Z. Qin, X. Mao, P. Yang, Z. Qin, and R. Wang, "SmokeGrenade: An efficient key generation protocol with artificial interference," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1731–1745, Nov. 2013.

[11] Z. Cao, G. Xiong, Y. Zhao, Z. Li, and L. Guo, "A survey on encrypted traffic classification," in *Proc. Int. Conf. Appl. Techn. Inf. Secur.* Berlin, Germany: Springer, 2014, pp. 73–81.

[12] K. Shahbar and A. N. Zincir-Heywood, "How far can we push flow analysis to identify encrypted anonymity network traffic?" in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, Taipei, Taiwan, Apr. 2018, pp. 1–6.

[13] D. Adami, C. Callegari, S. Giordano, M. Pagano, and T. Pepe, "Skype-Hunter: A real-time system for the detection and classification of Skype traffic," *Int. J. Commun. Syst.*, vol. 25, no. 3, pp. 386–403, Mar. 2012.

[14] G. Maiolini, A. Baiocchi, A. Iacovazzi, and A. Rizzi, "Real time identification of SSH encrypted application flows by using cluster analysis techniques," in *Proc. Int. Conf. Res. Netw.* Berlin, Germany: Springer, 2009, pp. 182–194.

[15] L. Bernaille and R. Teixeira, "Early recognition of encrypted applications," in *Proc. Int. Conf. Passive Act. Netw. Meas.* Berlin, Germany: Springer, 2007, pp. 165–175.

[16] T. Yildirim and P. J. Radcliffe, "VoIP traffic classification in IPSec tunnels," in *Proc. Int. Conf. Electron. Inf. Eng.*, Kyoto, Japan, Aug. 2010, pp. V1-151–V1-157.

[17] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing Skype traffic: When randomness plays with you," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, Aug. 2007, pp. 37–48.

[18] J. Tan, X. S. Chen, M. Du, and K. Zhu, "Internet traffic identification algorithm based on adaptive BP neural network," *J. Univ. Electron. Sci. Technol. China*, vol. 41, no. 4, pp. 580–585, Jul. 2012.

[19] M. H. M. Soleimani, M. Mansoorizadeh, and M. Nassiri, "Real-time identification of three Tor pluggable transports using machine learning techniques," *J. Supercomput.*, vol. 74, no. 10, pp. 4910–4927, Oct. 2018.

[20] A. S. D. Silva, C. C. Machado, R. V. Bisol, L. Z. Granville, and A. Schaeffer-Filho, "Identification and selection of flow features for accurate traffic classification in SDN," in *Proc. IEEE 14th Int. Symp. Netw. Comput. Appl.*, Sep. 2015, pp. 134–141.

[21] B. Li, M. Ma, and Z. Jin, "A VoIP traffic identification scheme based on host and flow behavior analysis," *J. Netw. Syst. Manage.*, vol. 19, no. 1, pp. 111–129, Mar. 2011.

[22] G. Xiong, W. Huang, Y. Zhao, M. Song, Z. Li, and L. Guo, "Real-time detection of encrypted thunder traffic based on trustworthy behavior association," in *Proc. Int. Conf. Trustworthy Comput. Services*. Berlin, Germany: Springer, May 2012, pp. 132–139.

[23] J. Zhang, Y. Xiang, W. Zhou, and Y. Wang, "Unsupervised traffic classification using flow statistical properties and IP packet payload," *J. Comput. Syst. Sci.*, vol. 79, no. 5, pp. 573–585, Aug. 2013.

[24] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, and Y. Guan, "Network traffic classification using correlation information," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 104–117, Jan. 2013.

[25] L. Ding, J. Liu, T. Qin, and H. Li, "Internet traffic classification based on expanding vector of flow," *Comput. Netw.*, vol. 129, pp. 178–192, Dec. 2017.

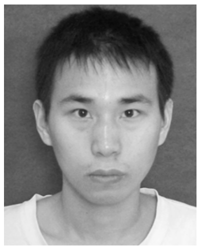
[26] N. Zhang, P. Yang, J. Ren, D. Chen, Y. Li, and X. Shen, "Synergy of big data and 5G wireless networks: Opportunities, approaches, and challenges," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 12–18, Feb. 2018.

[27] S. Suthaharan, "Big data classification: Problems and challenges in network intrusion prediction with machine learning," *SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 4, pp. 70–73, Apr. 2014. [Online]. Available: http://doi.acm.org/10.1145/2627534.2627557

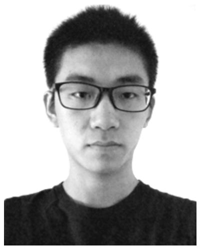
[28] X. Chen, X. Zeng, and W. Wang, "Big data analytics for network security and intelligence," *Adv. Eng. Sci.*, vol. 49, no. 3, pp. 1–12, 2017.
[29] Q. Wang, D. Chen, N. Zhang, Z. Qin, and Z. Qin, "LACS: A lightweight label-based access control scheme in IoT-based 5G caching context," *IEEE Access*, vol. 5, pp. 4018–4027, 2017.
[30] D. Chen *et al.*, "S2M: A lightweight acoustic fingerprints-based wireless device authentication protocol," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 88–100, Feb. 2017.
[31] A. Aleroud and G. Karabatis, "Context infusion in semantic link networks to detect cyber-attacks: A flow-based detection approach," in *Proc. IEEE Int. Conf. Semantic Comput.*, Newport Beach, CA, USA, Jun. 2014, pp. 175–182.
[32] B. Anderson and D. McGrew, "Identifying encrypted malware traffic with contextual flow data," in *Proc. ACM Workshop Artif. Intell. Secur.*, New York, NY, USA, Oct. 2016, pp. 35–46.
[33] L. Xu, "Context aware traffic identification kit (TriCK) for network selection in future HetNets/5G networks," in *Proc. Int. Symp. Netw., Comput. Commun.*, Marrakech, Morocco, May 2017, pp. 1–5.
[34] Z. Rao, W. Niu, X. S. Zhang, and H. Li, "Tor anonymous traffic identification based on gravitational clustering," *Peer-Peer Netw. Appl.*, vol. 11, no. 3, pp. 592–601, May 2017.
[35] J. Lingyu, L. Yang, W. Bailing, L. Hongri, and X. Guodong, "A hierarchical classification approach for Tor anonymous traffic," in *Proc. IEEE 9th Int. Conf. Commun. Softw. Netw.*, Guangzhou, China, May 2017, pp. 239–243.
[36] A. Cuzzocrea, F. Martinelli, F. Mercaldo, and G. Vercelli, "Tor traffic analysis and detection via machine learning techniques," in *Proc. IEEE Int. Conf. Big Data*, Boston, MA, USA, Dec. 2017, pp. 4474–4480.
[37] F. Mercaldo and F. Martinelli, "Tor traffic analysis and identification," in *Proc. AEIT Int. Annu. Conf.*, Cagliari, Italy, Sep. 2017, pp. 1–6.
[38] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related," in *Proc. 2nd Int. Conf. Inf. Syst. Secur. Privacy*, 2016, pp. 407–414.
[39] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Tor traffic using time based features," in *Proc. ICISSP*, Jan. 2017, pp. 253–262.
[40] R. Ding and W. Li, "A hybrid method for service identification of SSL/TLS encrypted traffic," in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Oct. 2016, pp. 250–253.
[41] S.-H. Lee, Y.-H. Goo, J.-T. Park, S.-H. Ji, and M.-S. Kim, "Sky-Scope : Skype application traffic identification system," in *Proc. 19th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Seoul, South Korea, Sep. 2017, pp. 259–262.
[42] W. Chen, C. Li, J. Shen, W. Zhang, and G. Yang. (2016). *Webpage Fingerprint Identification Method Aiming at Specific Website Category*. [Online]. Available: https://www.google.com/patents/CN105281973A?cl=en
[43] Z. Zhuo, Y. Zhang, Z.-L. Zhang, X. Zhang, and J. Zhang, "Website fingerprinting attack on anonymity networks based on profile hidden Markov model," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1081–1095, May 2018.
[44] Z. Deng, Z. Liu, Z. Chen, and Y. Guo, "The random forest based detection of Shadowsock's traffic," in *Proc. 9th Int. Conf. Intell. Human-Mach. Syst. Cybern. (IHMSC)*, Hangzhou, China, Aug. 2017, pp. 75–78.
[45] F. Toomey, "Bursty traffic and finite capacity queues," *Ann. Oper. Res.*, vol. 79, pp. 45–62, Jan. 1998.
[46] Tor (Aug. 2018). *I Keep Seeing These Warnings About SOCKS and DNS Information Leaks. Should I Worry*. [Online]. Available: https://www.torproject.org/docs/faq.html.en#WarningsAboutSOCKSand-DNSInformationLeaks
[47] T. Callahan, M. Allman, and M. Rabinovich, "On modern DNS behavior and properties," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 3, pp. 7–15, Jul. 2013.

XUEMEI ZENG received the M.S. degree from the Computer Science College, Sichuan University, China, in 2004, where she is currently pursuing the Ph.D. degree in information security. She is also an Engineer with the CyberSecurity Research Institute, Sichuan University. Her current research interests include network behavior analysis and big data analysis for security.

XINGSHU CHEN received the master's and Ph.D. degrees from Sichuan University, Chengdu, China, in 1999 and 2004, respectively, where she is currently a Full Professor, and also a Doctoral Supervisor with the College of Cybersecurity. Her main research focus is related to cloud computing, big data analysis, and network security.



GUOLIN SHAO is currently pursuing the Ph.D. degree in computer science with Sichuan University, Chengdu, China. Until now, he has published more than 16 academic papers. His research interests include deep learning for cyber security and network behavior analysis.



TAO HE received the B.S. degree from the Computer Science College, Sichuan University, Chengdu, China, in 2017, where he is currently pursuing the M.S. degree (second year) in cyber security. His research interests include network behavior analysis and big data analysis for security.

ZHENHUI HAN received the B.S. degree from the Software College, Sichuan University, Chengdu, China, in 2017, where she is currently pursuing the M.S. degree (second year) in cyber security. Her research interests include network behavior analysis for security, and user behavior and entity analysis.

YI WEN received the B.S. degree from the College of Electronics and Information Engineering, Sichuan University, Chengdu, China, in 2018, where he is currently pursuing the M.S. degree (first year) in cyber security. His research interests include network behavior analysis and big data analysis for cyber security.

QIXU WANG received the B.Sc. degree from the Southwest University of Science and Technology, in 2009, and the Ph.D. degree in computer science and technology from the University of Electronic Science and Technology of China, in 2017. He is currently an Assistant Researcher with the College of Cybersecurity, Sichuan University. His current research interests include cloud computing security, wireless network security, trusted computing, and data privacy protection.