Received: 5 June 2019 | Revised: 7 December 2019 | Accepted: 6 February 2020

DOI: 10.4218/etrij.2019-0299

ORIGINAL ARTICLE

ETRI Journal WILEY

Refined identification of hybrid traffic in DNS tunnels based on regression analysis

Huiwen Bai[1] | **Guangjie Liu**[1,2] | **Jiangtao Zhai**[2] | **Weiwei Liu**[1] | **Xiaopeng Ji**[2] | **Luhui Yang**[1] | **Yuewei Dai**[2]

[1]School of Automation, Nanjing University of Science and Technology, Nanjing, China

[2]School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing, China

Correspondence
Guangjie Liu, School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing, China.
Email: gjieliu@gmail.com

Funding information
National Natural Science Foundation of China, Grant/Award Number: U1836104, 61702235 and 61921004; Fundamental Research Funds for the Central Universities, Grant/Award Number: 30918012204

DNS (Domain Name System) tunnels almost obscure the true network activities of users, which makes it challenging for the gateway or censorship equipment to identify malicious or unpermitted network behaviors. An efficient way to address this problem is to conduct a temporal-spatial analysis on the tunnel traffic. Nevertheless, current studies on this topic limit the DNS tunnel to those with a single protocol, whereas more than one protocol may be used simultaneously. In this paper, we concentrate on the refined identification of two protocols mixed in a DNS tunnel. A feature set is first derived from DNS query and response flows, which is incorporated with deep neural networks to construct a regression model. We benchmark the proposed method with captured DNS tunnel traffic, the experimental results show that the proposed scheme can achieve identification accuracy of more than 90%. To the best of our knowledge, the proposed scheme is the first to estimate the ratios of two mixed protocols in DNS tunnels.

KEYWORDS
DNS tunnels, ratios of two mixed protocols, regression model, temporal-spatial analysis

1 | INTRODUCTION

At a time when network security and monitoring are of vital importance due to increasing network traffic and growing cybersecurity concerns, comprehensive network surveillance is a necessity. Information, such as protocol type or application signature, proves invaluable for network administrators and network designers. However, with the need for privacy to be protected and the prevalence of malware, anti-monitoring techniques are developing rapidly. There are diverse techniques that afford network users the capability of circumventing censorship.

With the growing use of cryptographic protocols such as SSL and SSH, increasing amounts of network traffic cannot be monitored using deep packet inspection (DPI) tools. This has enhanced the undetectability of some malicious network activities. One covert communication technique for distributing malware is to set up a tunnel to transmit data over a frequently-used standard protocol [1]. Several techniques have been designed for disguising one protocol as another to bypass censorship at network boundaries. There has recently been an increase in the malicious use of tunnel technology. Currently, the most common tunneling technologies are HTTP tunneling [2], SSH tunneling [3], and DNS tunneling [4]. For these tunnels, the actually prohibited traffic is encapsulated within the payload of covering protocols. As a result, the flow in the tunnel cannot be directly inspected,

which has increased the difficulty of identifying malicious traffic.

To resolve the risks caused by these tunneling technologies it is necessary to analyze the features of the tunnel traffic to reveal the nature of the protocol running inside.

However, it is an industry axiom that neither port numbers nor DPI can be used to reliably identify network applications. Many malicious Internet applications with random port numbers have equipped enabled tunneling techniques, such as DNS, to penetrate firewalls without being blocked [5]. Monitoring a comprehensive network is becoming more difficult due to the existence of these techniques which disrupt the network traffic. DNS, as a distributed infrastructure that can store, update, and disseminate data, matches the requirements for the carrier of network tunnels. DNS packets are allowed by almost all the security devices. As a result, recent years have witnessed the increasing use of stealthy, prevailingly malicious DNS tunnels. Several strains of malware such as the Morto worm and Feederbo [6], and variants of point-of-sale malware such as BernhardPOS and FrameworkPOS [7] demonstrate the increased popularity of DNS tunneling to implement stealthy communications. Application-layer tunnels simply can be simply built on top of DNS by exploiting the way how regular standard DNS requests for a given domain are forwarded to its authoritative servers. Several free DNS tunnel tools, such as Dnscat, Iodine, NSTX, DeNiSe OzymanDNS, and Heyoka, are commonly used. The original traffic cannot be directly inspected as it becomes a part of the domain name of the DNS packets. There have been several schemes for discovering DNS tunnels [8–12]. Nevertheless, more detailed information about DNS tunneling is still unavailable to the network inspectors. The network audit facility can still not determine the protocol or application inside the DNS tunnel. To effectively prevent malicious activities on the network, it is necessary for network analysis to identify the tunnel traffic's carrier protocol especially the protocol within the tunnel. Currently, there is a lack of detailed analysis of the traffic in DNS tunnels. It was first proposed to identify the protocol within DNS tunnels in [13]. This method, however, simply classifies the traffic under the DNS tunnels and has many restrictions. For some scenarios in C&C communication and to circumvent censorship, more than one protocol may be used in a DNS tunnel, which can make render the identification method for single protocol ineffective. Thus, further additional identification methods for multi-protocols within DNS tunnels should be developed to bridge this gap.

The existing work on DNS tunnel identification aims to identify the protocol within the DNS tunnel, these concentrate on the ideal case when only a single protocol is used in the DNS tunnel, the protocol-mixed case is not taken into consideration. In this paper, we concentrate on the identification of two protocols combined in a DNS tunnel. A feature set is first derived from the DNS request and response flows, which are incorporated by deep neural networks to construct a regression model. To the best of our knowledge, there is an absence of efforts to identify hybrid protocols in DNS tunnels. The central proposition is that the size and transferred pattern information carried by queries and responses in DNS packets are sufficient to infer the nature of the application protocols in the DNS tunnels, even when two protocols are combined. By characterizing the relationship between the properties and the component, we demonstrate that it is possible to calculate the packet ratio in the case of two protocols being used in DNS tunnels.

The main research contributions of this paper are:

1. We designed a feature set based on DNS request and response flows, which could discriminate among protocols in DNS tunnels, specifically to be adopted in the case of two protocols being combined within a DNS tunnel.
2. We prove that the two combined protocols in the tunnel can be further analyzed and the packet ratio of the two protocols in flows can be estimated with the proposed scheme.
3. We provide an experimental analysis of the proposed scheme using real DNS tunnel traffic, to prove the effectiveness of the proposed scheme.

The rest of the paper is organized as follows. In the next section, we summarize the related works about the detection of tunnels based on the main application layer protocol, especially the DNS tunnels, and the identification of internal tunnel traffic. Section 3 is dedicated to the description of the features to discriminate the flows with different multiple protocols in DNS tunnels and to contribute to estimating the ratio of the two protocols hiding in one tunnel. In Section 4, we introduced the proposed scheme for managing the hybrid DNS tunnel traffic containing two combined protocols. In Section 5, we introduce the data sources. In Section 6, we establish three regressive models for HTTP & SMTP, HTTP & SSH, and SMTP & SSH. respectively. In Section 7, we used the data set to validate the proposed method. In Section 8, we offer the conclusions of this research and discuss future work.

2 | RELATED WORKS

A lot of reference works have been done to detect abnormal DNS traffic. Born and Gustafson [8] empirically showed that normal DNS question names follow Zipf's law, that is, an English-like distribution. Question names of DNS tunnels, however, show a much flatter distribution since tunneled traffic is often compressed or encrypted and then encoded for transmission. They also developed an n-gram visualization called NgViz [9], so that an operator can use spatial reasoning to quickly identify anomalies in DNS traffic. Their analysis

was limited to unigrams and bigrams, that is, 1-grams and 2-grams. Qi and others [10] also used bigrams for DNS tunnel detection. They calculated the bigrams present in each DNS query and used bigram frequencies calculated in an offline training phase to score the DNS query. Ellens and others [11] used a traffic-analysis approach to DNS tunnel detection. They ran the DNS tunnel tool, Iodine on a host in a campus network, and use the tunnel to exfiltrate dummy data, run an interactive session mimicking a command and control channel, and also browse web content bypassing security measures on the network. In contrast to payload techniques, the authors analyzed only flow metadata collected using an IPFIX generator called yaf. Cejka and others [12] focused on the detection of communication tunnels and other anomalies in DNS traffic. The proposed detection module is designed to process huge volumes of data and to detect anomalies in near real-time. It is based on the combinations of statistical analyses of several observed features including application layer information. Kara and others [14] presented a system to analyze the resource record activities of domain names and build DNS zone profiles to detect payload distribution channels. Their work was based on an extensive year-long analysis of malware datasets, and a near real-time feed of passive DNS traffic. Anna and others [15] extracted features from the data set that employed a penetration testing effort within a DNS tunnel and trained random forest classifiers to distinguish normal DNS activity from tunneling activity. Aiello and others [16–19] have done extensive research on DNS tunnels. They first evaluated the performance of the DNS tunnel and some common tools [16]. In [17] they undertook DNS tunneling detection by means of simple supervised learning schemes, applied to the statistical features of DNS queries and answers. In [18], they obtained results from experiments conducted on a live network by replicating individual detections over successive samples over time and making a global decision using a majority voting scheme. The technique overcomes traditional classifier limitations. Then they [19] investigated the application of a second-level classifier, which applies a fusion of the classifications made by the traditional classifiers. Finding the minimum number of samples of the second-level classifier to achieve reliable detection is their ultimate objective. Liu and others [20] proposed an effective and applicable DNS tunnel detection mechanism. This approach analyzes Recursive DNS (RDNS) for detection. The main difference between this method and the previous is that it focuses on the character distribution differences between normal and tunnel traffic and also time frequency. To better apply the machine learning method to tunnel flow analysis, Davis and Foo [21] proposed automated feature engineering to derive a suite of additional features from a given set of basic features with the aim of improving classifier accuracy through discriminative features and to assist data scientists through automation.

Since the discovery introduction of application protocol tunneling, some studies on traffic identification in DNS tunnels have been carried out on traffic identification in DNS tunnels. Homem and Papapetrou [22] presented a machine learning approach, based on feature subsets of network traffic evidence, to aid forensic analysis through automating the inference of protocols carried within DNS tunneling techniques. They explore four network protocols, namely, HTTP, HTTPS, FTP, and POP3. Three features are extracted from the DNS tunneled traffic: IP packet length, DNS Query Name Entropy, and DNS Query Name Length. Almusawi and Amintoosi [23] proposed a multi-label support vector machine in order to detect and classify the DNS tunneling. The proposed method was evaluated using a benchmark dataset that contains contained numerous DNS queries and was compared with a multi-label Bayesian classifier based on the number of corrected classified DNS tunneling instances. In this paper, we concentrate on the refined identification of two protocols mixed combined in a DNS tunnel. A feature set is first derived from DNS request and response flows, which is then incorporated with deep neural networks to construct a regression model.

3 | CHARACTERISTICS ANALYSIS

In this section, the features extracted from DNS flows with two-protocol mixed in DNS tunnels are introduced. We analyze the DNS tunnel flows from two perspectives, namely length and time. DNS tunnels operate by encapsulating original data into query, name or response of a DNS packet, being encoded and encrypted, which render the DNS packet data unreadable. In such cases, we can only use the metadata of the DNS flows, namely length and interarrival time (IAT) of packets. Based on the metadata of the DNS tunnel flows, we further obtain a series of statistical features that characterize behaviors of real protocols or applications. Considering the difference of bandwidth in two directions of DNS flows, we deal with incoming and outgoing DNS traffic and extract features separately. In DNS tunnels, packets can be divided into two types according to their functions, namely data packets and control packets. Data packets are used for transferring user data and control packets are used for heartbeat or padding. For different protocols in DNS tunnels, the amount or frequency of control packets also varies. We parse all DNS tunnel packets including data packets and control packets, and extract the query name or answer, from which we obtain metadata, namely the length of user data and IAT. We provide a summary of features of DNS tunnel flows as shown in Table 1.

The amount of data transmitted per packet is important for the identification of different protocols. It is also the basic factor that influences the ratio between two protocols in DNS

TABLE 1 Features of DNS tunnel flows

No.	Short	Description
F_1	min_IAT	Minimum packet IAT
F_2	q1_IAT	First quartile IAT
F_3	med_IAT	Median IAT
F_4	mean_IAT	Mean IAT
F_5	q3_IAT	Third quartile packet IAT
F_6	max_IAT	Maximum packet IAT
F_7	var_IAT	Variance in packet IAT
F_8	min_data	Minimum of the query name/answer size
F_9	q1_data	First quartile of the query name/answer size
F_{10}	med_data	Median of the query name/answer size
F_{11}	mean_data	Mean of the query name/answer size
F_{12}	q3_data	Third quartile of the query name/answer size
F_{13}	max_data	Maximum of the query name/answer size
F_{14}	Var_data	Variance of the query name/answer size
F_{15}	control	Proportion of heartbeat packets

tunnels. When two protocols are combined, the amount of data transmitted varies while the mixing ratio changes. To describe this feature more accurately, we select mean and variance as measures of the data transferred. Mean and variance can determine how much data has been transferred in the DNS Tunnels and the difference between per packet and the mean. Besides, with the use of different protocols in DNS tunnels, features extracted from DNS flows also behave differently. The size of the user data tunneled in DNS packets is limited to a certain length, which is determined by the DNS tunneling server and client, the used method used and so on. Consequently, the transferred data generally conforms to a certain pattern depending on the inner network protocol. It can be inferred that when two different protocols are used in DNS tunnels, these features will change with the difference of the packet ratio between the two protocols. Each protocol transmits data in a regular pattern which is reflected in its length can be found in the length distribution of a certain rule.

The existence of heartbeat packets is an important feature of DNS tunnels. Since the DNS tunnels use a way of enduring connections, the client must ensure that it remains connected to the server during free time. Namely, when there is no user data being transmitted, the client still has to send packets to the server and gets responses, which is known as the heartbeat. Because the client is only ensuring that the server is connected, heartbeat packets are usually short. Besides, when a specific protocol is used in a DNS tunnel for transmission, the heartbeat packet can play a greater than expected role. Namely, when the request sent by the DNS tunnel internal protocol to the server or the data sent by the server in response is hosted by multiple DNS packets, the corresponding reverse direction packets must be supplemented by the heartbeat packets. For different protocols, the number of packets used to transmit data for a request or response is different, and the number of heartbeat packets also varies.

Besides, IAT also has a great impact on the ratio between two protocols in DNS tunnels. Among the three protocols, HTTP and SSH are highly interactive application protocols, while SMTP, on the contrary, is a low interactive application protocol. The differences between the two kinds of protocols are that the IAT of the protocol with high interactivity is greater than that of the protocol with low interactivity. IAT with high interactivity is not regular in general, but it is relatively fixed with low interactivity.

4 | SCHEME OF REFINED IDENTIFICATION OF HYBRID TRAFFIC IN DNS TUNNELS

In this section, we introduce the framework of the proposed scheme, which depicts the process of identifying two-protocol combined in a DNS tunnel. Figure 1 shows the framework proposed for the scheme. We first describe the factors influencing the ratio between the two protocols in DNS tunnels and the method of extracting their principal components. A regression model based on a supervised learning methodology is proposed to analyze the ratio between them.

There are multiple correlations between the factors influencing the ratio of the two protocols. The multiple

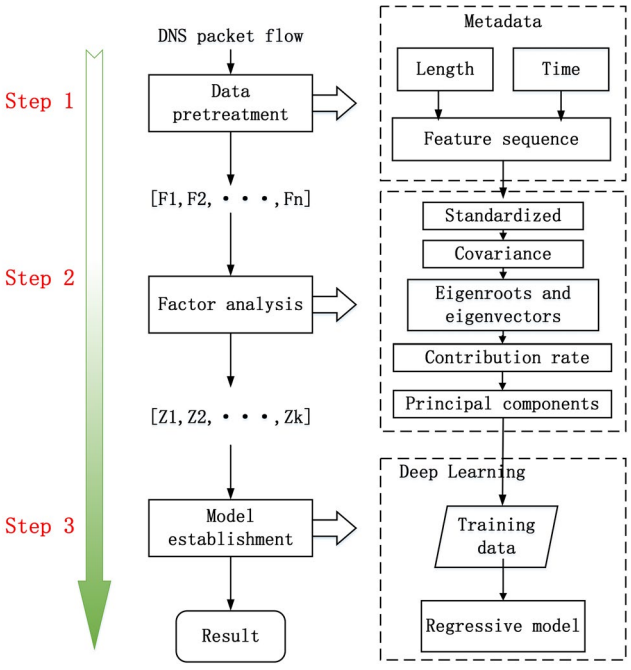


FIGURE 1 Framework of identifying hybrid DNS tunnel traffic containing two combined protocols

correlations, also known as multicollinearity, refer to a linear correlation between variables. In regression analysis, when there is a multiple correlation between independent variables, it will have a great impact on the estimation of the regression coefficient. Therefore, if multiple correlations among various influencing factors are observed before regression analysis, principal component analysis method should be adopted to eliminate the multiple correlation. The principal component analysis is a method to convert multiple indexes into a few comprehensive indexes. The solution steps of principal component extraction are as follows:

Step 1: The normalized data of each indicator were processed to obtain the normalized matrix before the principal component analysis, in order to eliminate the dimensional differences between statistical indicators. Suppose there are p factor independent variables, and a total of n samples. Then the factor independent variable matrix is $X = [x_1, x_2, \ldots, x_p]$. The normalization formula is as follows:

$$\hat{X} = \left(\frac{x_{ij} - \bar{x}_j}{a_j} \right)_{n \times p}, \tag{1}$$

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij}, \tag{2}$$

$$a_j^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left(x_{ij} - \bar{x}_j \right)^2. \tag{3}$$

In the formula, $i = 1, 2, \ldots, n; j = 1, 2, \ldots, p$.

Step 2: Calculate the codifference matrix of the sample data matrix of independent variables of the influencing factors after standardized transformation, and the calculation formula is

$$S^2 = \left(s_{ij}^2 \right)_{p \times p} = \frac{1}{n-1} \hat{X}^T \hat{X}. \tag{4}$$

Step 3: Replace the total codifference matrix with the sample codifference matrix. Calculate all characteristic root λ_j of S with the corresponding characteristic vector, namely principal component and arrange in order of size of λ_j.

Step 4: Calculate the contribution rate and cumulative contribution rate of the principal components.

$$K_j = \frac{\lambda_j}{\sum_{j=1}^{p} \lambda_j}, \tag{5}$$

$$K_r = \frac{\sum_{j=1}^{r} \lambda_j}{\sum_{j=1}^{p} \lambda_j}. \tag{6}$$

In (5), K_j is the contribution rate of the main component. In (6), K_r is the cumulative contribution rate of the first r^{th} principal components.

Step 5: Determine the number of principal components according to the cumulative contribution rate. Generally, the cumulative contribution rate is not less than 85%. When the number of principal components whose cumulative contribution rate reaches 85%. Then the principal components $Z = [z_1, z_2, \ldots, z_k]$ are extracted.

Based on the extracted principal components $Z = [z_1, z_2, \ldots, z_k]$, regression analysis was carried out on the flows with two protocols in DNS tunnels. Assume that the multiple regression model is denoted as

$$y = f(A, Z) + \varepsilon, \tag{7}$$

where ε is a random error term, which indicates the influence of other random factors on the ratio of two application protocols y. It is generally assumed that it obeys the normal distribution $N(0, \delta)$. A is the model regression coefficient. Because there are many factors that influence the ratio of two protocols in DNS tunnels, and the relationship between the factors and the ratio is complicated, we use the deep neural network to establish the relationship model between factors and the ratio.

The deep learning framework in this paper consists of an input layer, three hidden layers and an output layer. The input layer accepts the k factors of traffic features, namely $Z = [z_1, z_2, \ldots, z_k]$. Then these k factors are connected to the hidden layers. There are three hidden layers in total. The hidden layers can be represented as $H_p = [h_{p1}, h_{p2}, \ldots, h_{pm}]$, where p represents the hidden layers and m represents the number of hidden layers. The first hidden layer contains 16 nodes, the second contains 8 nodes, and the third layer contains 4 nodes. The three hidden layers are all fully connected layers, namely each node in the previous layer is connected to each node in the subsequent layer. The connection method is denoted as

$$H_p = WH_{p-1} + B, \tag{8}$$

where W is the matrix of the weight and B is the bias vector. In the simplest case, the data is linear, namely, only a straight line is needed to accurately classify or fit the samples. However, in complex cases, the data are linearly indivisible, so nonlinear factors are introduced to process the samples. Because the linear model cannot meet the specifications, non-linear factors must be introduced. Sigmoid and tanh are the two frequently used nonlinear activation functions in the fully connected layer. Their form is denoted as:

$$f(x) = \frac{1}{1 + e^{-x}}, \tag{9}$$

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (10)$$

The activation function used in the first two hidden layers is the tanh function and the activation function used in the output layer is the sigmoid function.

The learning process is divided into forwarding propagation and back-propagation. Forward propagation allows input information to be transmitted to the output layer under the corresponding weight, threshold and activation functions. When the error between the output result and the expected value is greater than the specified precision, the sensitivity is back-propagated, and the weight and threshold of each layer are corrected incrementally. With repeated iteration, the error finally reaches the specified precision. In this research, we first calculated the error between the output and the actual value. Then the gradient descent method was used for iterative calculation, and the error reached the minimum value.

5 | DATASET SETUP

To analyze the performance of the scheme for estimating the ratio of components in the DNS Tunnels, we used network traffic generated by three protocols, namely HTTP, SSH, and SMTP running in DNS Tunnels. As shown in Figure 2, two DNS tunnels were built for our experiments. The first tunnel was built between an Amazon EC2 cloud server located in Dublin, Ohio, USA, and clients which have access to the Internet via the campus network in Nanjing University of Science and Technology (NJUST), and the second tunnel was built between the Amazon EC2 cloud server and another Amazon EC2 server. With the two DNS tunnels, we generated hybrid DNS tunnel traffic containing two combined protocols. DNS tunnel server access networks to meet the needs of clients to access the network. These protocols were selected as they represent three important types of applications on the Internet, namely Websites, remote control, and E-mail. To capture the traffic of DNS Tunnels, the destination was implemented in Amazon Elastic Compute Cloud (Amazon EC2), which is a web service that provides secure, resizable compute capacity in the cloud. The source was a laptop which was connected to the Internet via public WiFi on the NJUST campus. There is another DNS Tunnel client in the public network. And then the DNS Tunnel was built using a common tool named Iodine.

In practice, without loss of generality, we assume that the three network protocols were used to get Web pages, control a remote host, and send E-mail. For the HTTP protocol, the Web pages we got from the Internet are all from the Alexa Top 1 000 000 Sites. For the SSH protocol, we used the most frequently-used command to control the destination. For the SMTP protocol, we sent E-mail with different content to the addressee.

For simplicity, we divided the two-protocol mixed traffic in the DNS tunnel into 9 cases, in which the percentages of all protocols are the integer multiples of 10%. To achieve a more accurate estimation, the ratio step should be smaller. The model complexity and the training dataset would also expand rapidly with smaller ratio steps.

To generate DNS tunnel traffic that combines the two protocols at a certain ratio, we constructed a mass of network activities using these three types of protocols, and recorded the corresponding number of packets when it was independently implemented within the DNS tunnel. For two-protocol combined cases, we chose two activities from the activity set and placed them in the DNS tunnel. The ratio of their corresponding amount of the packets should be consistent with the tested ratio. The record of each traffic type is shown in Table 2. For the three types of mixed protocols in Table 2, the number of flows with each ratio is given. There are 12 586 DNS flows comprised of 13 GB of traffic data in total in the dataset. We use a C++ program, which was designed to process DNS flows, to extract the metadata, and calculate the statistical characteristics related to length and time. We saved all the features extracted from the DNS flows in a file, in which each row represents an instance of a DNS flow with a specific ratio between two protocols in the DNS tunnels. Also, each column represents an attribute of a DNS flow while the last column is marked as the actual ratio of the two protocols.

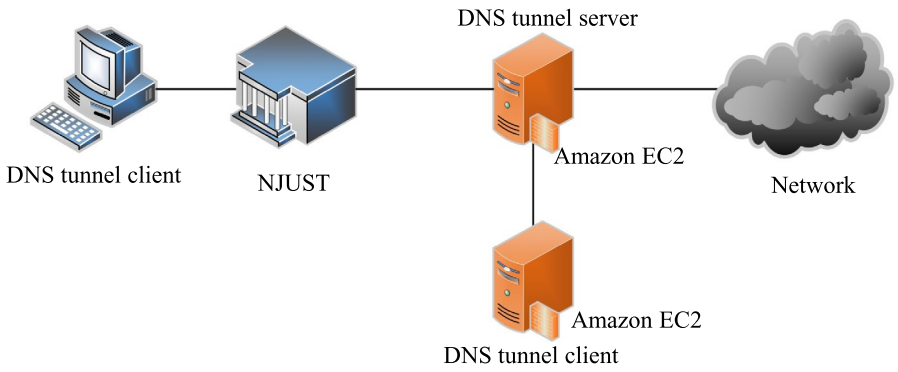


FIGURE 2 Network topology of experimental data source

TABLE 2 Different ratios of the two mixed protocols and the corresponding number of flow samples

Ratio of the two mixed protocols	Numbers of mixed protocol flows		
	HTTP & SMTP	HTTP & SSH	SMTP & SSH
0:10	468	496	496
1:9	383	425	441
2:8	506	480	444
3:7	420	418	610
4:6	363	492	489
5:5	468	470	578
6:4	450	480	376
7:3	384	444	521
8:2	403	529	498
9:1	451	510	553
10:0	448	448	468
Total	4744	5192	5474

6 | REGRESSIVE MODEL OF REFINED IDENTIFICATION OF HYBRID TRAFFIC IN DNS TUNNELS

In this section, we used three kinds of traffic, namely HTTP & SMTP, HTTP & SSH, and SMTP & SSH respectively, to establish a regressive model. Because the experimental data are collected from the actual network environment, and both sides of the communication follow multiple routes, the original time features failed to distinguish different protocols. More explicitly the time features in Table 1 had an irregular influence on the mixture ratio. Hereinafter, due to the diversity and instability of the time-related features when the network connections are established between hosts in different network domains, we only used the length-related features, namely, F_8–F_{15}. As both outgoing and incoming flows are used to extract bidirectional features, the dimension of the length-related features is 16. There were three kinds of traffic in the DNS tunnels, namely HTTP & SMTP, HTTP & SSH, and SMTP & SSH. Each traffic type was divided into nine samples according to their protocol ratios. Although the instance data used for training and testing was labeled 1–9, this tag is not very precise. Across a narrow range, the value of the mark contains an error. In the instance of HTTP & SMTP at mark 3, the actual ratio of HTTP to SMTP might be 2.8 or 3.3.

We first analyzed the correlation coefficient between various features that affect the ratio estimation of the two protocols, three types of samples including HTTP & SMTP, HTTP & SSH, and SMTP & SSH were analyzed. We analyzed the features F_8–F_{15} for both outgoing and incoming flows. There is a degree of correlation among these features that can be measured with (11).

$$\rho_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}. \tag{11}$$

In (11), cov represents covariance and D represents variance. With the calculated correlation coefficient, we found a strong correlation between particular features. Figure 3 shows the correlation coefficients among the features F_8–F_{15} in Table 1 for the three hybrid DNS tunnel flows. In order to eliminate the correlation between the features and select the features with the highest degree of correlation to the real ratio, we used the principal component analysis method. The eigenvalue, variance contribution rate, and cumulative contribution rate are shown in Table 3. We selected the eight features with the most impact on the estimation accuracy.

After the principal component was obtained, the data set was used as the input of the deep learning network, and the regression model was obtained after the training. In order to make full use of the sample data and increase the credibility of the model, we used the 10-fold cross-validation method. The experimental data was divided into two portions, nine-tenths for training data, and the remainder as test data. The method was iterated 10 times. To facilitate the evaluation of the results of the regression model calculation, we rounded the test results calculated by the model to obtain the final predicted value, and compared this value with the premarked value to obtain the final calculation accuracy. Compared with the actual ratio, the average accuracy was obtained. We trained regressive models for each of the three types of hybrid DNS tunnel traffic. After every 200 training iterations, we recorded the accuracy of the model. The process of the 10 000 training iterations of the three types of hybrid traffic is shown in Figure 4A and the average accuracy of the 10 sample sets in the testing process is shown in Figure 4B.

The model generated by the training will be retained for further testing. Prior to having it, we made a simple evaluation of the regression model. Figure 5 shows the residuals of the three final models compared with the marked values after prediction. It can be seen from the figure that the residual distribution is above and below the 0 value and the value is stable within a certain range, thus conforming to the law of random error. Therefore, the model is effective for estimating the ratio of the two protocols in the DNS tunnel.

7 | EVALUATION OF REGRESSIVE MODEL

In this section, we evaluate the performance of the regression model which was trained by using a supervised deep neural network using HTTP & SMTP, HTTP & SSH, and SMTP & SSH traffic, respectively, which was collected from the networks

	Y	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16
Y	1.00000																
F1	-0.99130	1.00000															
F2	0.91172	-0.92361	1.00000														
F3	-0.67218	0.64654	-0.57003	1.00000													
F4	-0.47916	0.43337	-0.14563	0.67817	1.00000												
F5	-0.39266	0.40283	-0.37310	0.41210	0.25745	1.00000											
F6	0.90867	-0.92858	0.86501	-0.57062	-0.34887	-0.38551	1.00000										
F7	0.38725	-0.38508	0.37657	-0.67013	-0.44250	-0.25979	0.46881	1.00000									
F8	-0.31710	0.32100	-0.36952	-0.13706	-0.10680	0.07700	-0.16165	0.18821	1.00000								
F9	-0.98571	0.99293	-0.91154	0.62819	0.41792	0.38194	-0.90272	-0.33170	0.35470	1.00000							
F10	-0.18914	0.21204	-0.42550	0.26413	-0.23973	0.14921	-0.24855	-0.27207	0.04906	0.19238	1.00000						
F11	0.55271	-0.53291	0.49549	-0.79412	-0.61754	-0.27812	0.44358	0.62806	-0.03632	-0.50133	-0.24353	1.00000					
F12	0.51135	-0.50918	0.60145	-0.66447	-0.29744	-0.27797	0.45819	0.69138	-0.09384	-0.48376	-0.43991	0.69322	1.00000				
F13	0.78203	-0.78945	0.88608	-0.57545	-0.16932	-0.34865	0.73897	0.43573	-0.29730	-0.77424	-0.49846	0.52187	0.63880	1.00000			
F14	0.86800	-0.88237	0.99346	-0.56943	-0.08532	-0.35983	0.83016	0.40226	-0.34773	-0.86832	-0.48736	0.50602	0.64502	0.89648	1.00000		
F15	0.22935	-0.20353	0.14731	-0.14645	-0.18229	-0.03749	0.27156	0.13270	0.31006	-0.19552	0.05084	0.07986	0.02601	0.09346	0.12616	1.00000	
F16	-0.95125	0.95901	-0.96785	0.64695	0.29383	0.37432	-0.88084	-0.40573	0.34112	0.95782	0.34529	-0.56101	-0.61891	-0.86417	-0.95206	-0.15207	1.00000

(A)

	Y	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16
Y	1.00000																
F1	-0.98020	1.00000															
F2	0.75807	-0.73979	1.00000														
F3	-0.97715	0.97126	-0.77951	1.00000													
F4	-0.80182	0.78343	-0.99478	0.82387	1.00000												
F5	-0.04274	0.04064	-0.05026	0.03741	0.04767	1.00000											
F6	0.77212	-0.82773	0.58959	-0.69429	-0.61650	-0.03985	1.00000										
F7	0.75707	-0.81098	0.53249	-0.67746	-0.56254	-0.03429	0.93910	1.00000									
F8	0.65620	-0.69931	0.58257	-0.60348	-0.60024	-0.05071	0.84730	0.74325	1.00000								
F9	-0.97916	0.98703	-0.70804	0.98653	0.75600	0.03743	-0.73802	-0.73167	-0.61559	1.00000							
F10	-0.13795	0.15776	-0.33452	0.15732	0.31909	0.01401	-0.17515	-0.14150	-0.20233	0.12899	1.00000						
F11	0.60720	-0.59326	0.86428	-0.62878	-0.84619	-0.04327	0.47605	0.42370	0.48184	-0.56355	-0.34389	1.00000					
F12	0.72811	-0.70745	0.99471	-0.74850	-0.98329	-0.04919	0.56815	0.50921	0.56634	-0.67314	-0.33351	0.87547	1.00000				
F13	0.72811	-0.70745	0.99471	-0.74850	-0.98329	-0.04919	0.56815	0.50921	0.56634	-0.67314	-0.33351	0.87547	1.00000	1.00000			
F14	0.72811	-0.70745	0.99471	-0.74850	-0.98329	-0.04919	0.56815	0.50921	0.56634	-0.67314	-0.33351	0.87547	1.00000	1.00000	1.00000		
F15	-0.02150	0.04539	-0.06104	0.14936	0.06319	0.00296	0.22565	0.18953	0.23421	0.09964	-0.01715	-0.04611	-0.05973	-0.05973	-0.05973	1.00000	
F16	-0.87882	0.87234	-0.85364	0.89210	0.88420	0.04166	-0.67954	-0.64928	-0.60866	0.85828	0.23016	-0.69846	-0.82729	-0.82729	-0.82729	0.21311	1.00000

(B)

	Y	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16
Y	1.00000																
F1	0.96123	1.00000															
F2	-0.98725	-0.92935	1.00000														
F3	-0.20144	-0.05664	0.29527	1.00000													
F4	-0.23732	-0.10847	0.32889	0.77884	1.00000												
F5	0.06558	0.00458	-0.17253	-0.31750	-0.10766	1.00000											
F6	-0.89592	-0.89975	0.88665	0.32496	0.28820	-0.04038	1.00000										
F7	0.53211	0.41038	-0.56457	-0.55595	-0.55835	0.07729	-0.48208	1.00000									
F8	0.89001	0.94521	-0.86600	0.04974	-0.05778	0.08852	-0.82403	0.35672	1.00000								
F9	0.95209	0.98787	-0.92672	-0.04571	-0.11853	0.08242	-0.88008	0.43865	0.95870	1.00000							
F10	0.98449	0.93732	-0.98692	-0.25732	-0.31086	0.09155	-0.88462	0.54917	0.86413	0.92651	1.00000						
F11	0.21982	0.17685	-0.25537	-0.59246	-0.69013	-0.00132	-0.33696	0.48172	0.09398	0.17960	0.27857	1.00000					
F12	-0.23542	-0.30399	0.17102	-0.43067	-0.62171	-0.05190	0.11859	0.20111	-0.30643	-0.29494	-0.15819	0.32567	1.00000				
F13	-0.63886	-0.63719	0.61588	-0.03003	-0.16089	-0.12563	0.62006	-0.32622	-0.61100	-0.64313	-0.57707	0.00578	0.37191	1.00000			
F14	-0.99064	-0.93471	0.99939	0.27892	0.31581	-0.14325	0.88891	-0.56149	-0.86873	-0.92946	-0.98838	-0.24645	0.17926	0.62219	1.00000		
F15	-0.58021	-0.47056	0.59768	0.48892	0.51383	-0.00617	0.61524	-0.59434	-0.38548	-0.46888	-0.59978	-0.52474	-0.09802	0.35759	0.59648	1.00000	
F16	0.95682	0.94844	-0.94588	-0.20055	-0.25977	0.09019	-0.90798	0.54187	0.90265	0.96108	0.93796	0.27977	-0.19213	-0.63935	-0.94713	-0.60574	1.00000

(C)

FIGURE 3 Correlation coefficient matrix of features F_8–F_{15} in Table 1 of HTTP & SMTP, HTTP & SSH, and SMTP & SSH: (A) HTTP & SMTP; (B) HTTP & SSH; (C) SMTP & SSH

environment described in Section 5. The label of each sample instance is marked according to the actual ratio of the two network application protocols in the DNS tunnel. After the samples were input into the algorithm, the calculated results were output in the ratio of having one protocol in the DNS tunnel. The results obtained from the regression model represent four situations. Ratio type A serves as an example to produce the following conclusions:

a. True positive (TP): It belongs to the ratio A and is identified as A.
b. False positive (FP): It does not belong to the ratio A and is identified as A.
c. False negative (FN): It belongs to the ratio A but is not identified as A.
d. True negative (TN): It does not belong to the ratio A and is not identified as A.

The core evaluation indexes in machine learning are precision rate (PR), recall rate (RR) and accuracy rate (AR). AR represents the ratio of correctly identified samples in the total number of samples, as shown in (12). PR represents the ratio

TABLE 3 Principal components that affect the ratio of the mixing protocols within DNS tunnels

	HTTP & SMTP			HTTP & SSH			SMTP & SSH		
Principal	**Eigenvalue**	**Variance**	**Cumulative**	**Eigenvalue**	**Variance**	**Cumulative**	**Eigenvalue**	**Variance**	**Cumulative**
Z_1	8.55	50.44%	50.44%	10.11	63.19%	63.19%	8.87	55.44%	55.44%
Z_2	2.17	13.56%	64%	1.83	11.44%	74.63%	3.27	20.44%	75.88%
Z_3	1.46	9.13%	73.13%	1.25	7.81%	82.44%	1.12	7.0%	82.88%
Z_4	1.17	7.31%	80.44%	1.0	6.25%	88.69%	0.72	4.5%	87.38%
Z_5	0.82	5.13%	85.57%	0.79	4.94%	93.63%	0.54	3.38%	90.76%
Z_6	0.49	3.06%	88.63%	0.42	2.63%	96.26%	0.40	2.5%	93.26%
Z_7	0.44	2.75%	91.38%	0.24	1.5%	97.76%	0.38	2.38%	95.64%
Z_8	0.3	1.89%	93.27%	0.20	1.25%	99.01%	0.30	1.88%	97.52%

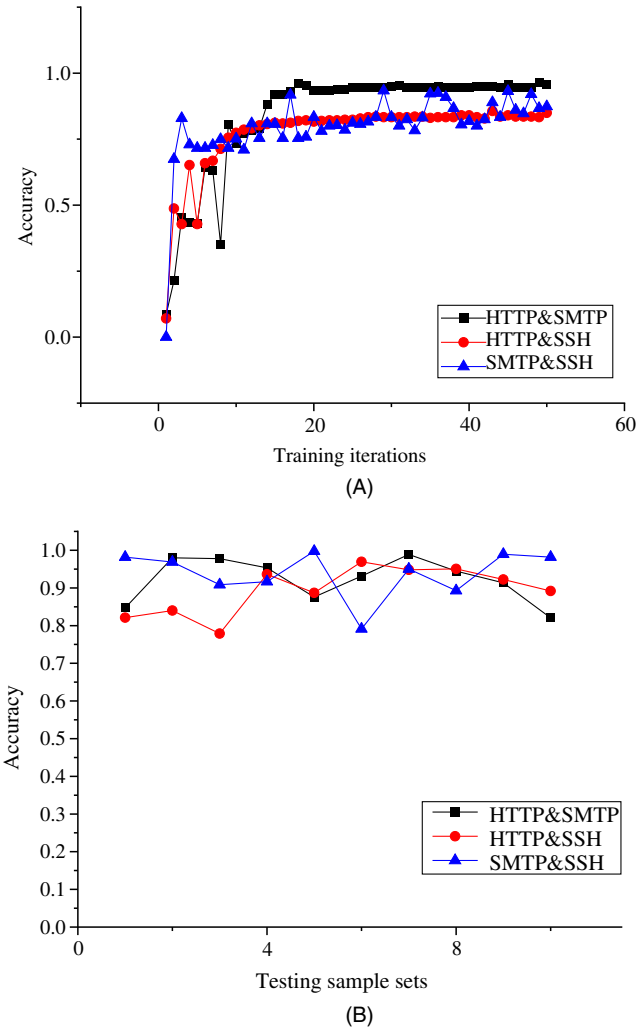


FIGURE 4 The accuracy in the training and testing process for HTTP & SMTP, HTTP & SSH, and SMTP & SSH: (A) Accuracy in the iterated training process; (B) Averaged accuracy of 10 sample sets in the testing process

of correctly identified samples in the number of all assigned samples, as shown in (13). RR represents the percentage of correctly classified samples that should have been correctly classified, as shown in (14).

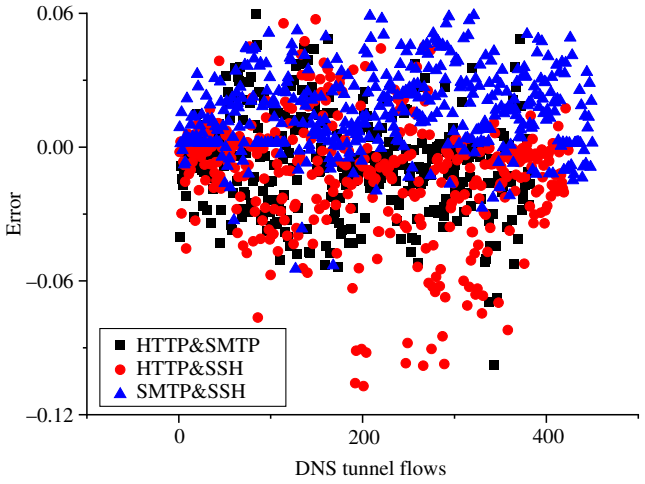


FIGURE 5 Error of regressive model trained of HTTP & SMTP, HTTP & SSH and SMTP & SSH

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{TN}+\text{FN}}, \tag{12}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \tag{13}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}. \tag{14}$$

Precision rate and recall rate reflect two aspects of the performance of the model. Therefore, the comprehensive index *F*-score of the precision rate and the recall rate is also needed to evaluate the performance. The calculation of an *F*-score is shown in (15).

$$F_\beta = (1+\beta^2)\times\frac{\text{Precision}\times\text{Recall}}{\beta^2\times\text{Precision}+\text{Recall}}. \tag{15}$$

It can be seen that *F*-score is a weighted harmonic average of precision rate and recall rate, and a high *F*-score indicates a good classification performance. We take the parameter $\beta = 1$,

namely, the accuracy and recall are equally important. We tested the performance of three regression models for HTTP & SMTP, HTTP & SSH, and SMTP & SSH, respectively, and it can be seen that our proposed method for estimating the packet ratio of two protocols in hybrid DNS tunnel traffic is effective.

Ratio of HTTP and SMTP		Actual ratios of HTTP and SMTP										
		0:10	1:9	2:8	3:7	4:6	5:5	6:4	7:3	8:2	9:1	10:0
Identified ratios of HTTP and SMTP	0:10	440	0	0	0	0	0	0	0	0	0	0
	1:9	0	362	4	0	1	1	3	0	0	0	0
	2:8	0	0	501	7	2	1	1	0	0	0	0
	3:7	0	0	0	413	8	0	0	0	0	0	0
	4:6	0	0	0	0	335	29	0	0	0	0	0
	5:5	0	0	0	0	17	436	0	0	0	0	0
	6:4	0	0	0	0	0	1	446	6	0	0	0
	7:3	0	0	1	0	0	0	0	378	0	0	0
	8:2	0	0	0	0	0	0	0	0	403	2	0
	9:1	8	0	0	0	0	0	0	0	0	449	0
	10:0	0	21	0	0	0	0	0	0	0	0	468

(A)

Ratio of HTTP and SSH		Actual ratios of HTTP and SSH										
		0:10	1:9	2:8	3:7	4:6	5:5	6:4	7:3	8:2	9:1	10:0
Identified ratios of HTTP and SSH	0:10	448	0	0	0	0	0	0	0	0	1	0
	1:9	0	421	2	0	0	0	0	0	0	0	0
	2:8	0	4	477	22	5	0	0	0	0	0	0
	3:7	0	0	1	392	12	1	1	0	0	0	0
	4:6	0	0	0	0	474	18	0	0	1	0	0
	5:5	0	0	0	4	1	451	29	0	0	0	0
	6:4	0	0	0	0	0	0	450	12	4	0	0
	7:3	0	0	0	0	0	0	0	416	111	0	0
	8:2	0	0	0	0	0	0	0	16	412	7	0
	9:1	0	0	0	0	0	0	0	0	1	502	0
	10:0	0	0	0	0	0	0	0	0	0	1	496

(B)

Ratio of SMTP and SSH		Actual ratios of HTTP and SSH										
		0:10	1:9	2:8	3:7	4:6	5:5	6:4	7:3	8:2	9:1	10:0
Identified ratios of SMTP and SSH	0:10	468	0	0	0	0	0	0	0	0	0	0
	1:9	0	440	0	0	0	0	0	0	0	0	0
	2:8	0	1	443	0	0	0	0	0	0	0	0
	3:7	0	0	1	601	3	0	0	0	0	0	0
	4:6	0	0	0	9	478	0	0	0	0	0	0
	5:5	0	0	0	0	8	569	1	0	0	0	0
	6:4	0	0	0	0	0	9	358	6	0	0	0
	7:3	0	0	0	0	0	0	17	515	0	0	0
	8:2	0	0	0	0	0	0	0	0	498	0	0
	9:1	0	0	0	0	0	0	0	0	0	553	0
	10:0	0	0	0	0	0	0	0	0	0	0	496

(C)

FIGURE 6 Confusion matrix of two-protocol combined at different ratios: (A) HTTP & SMTP; (B) HTTP & SSH; (C) SMTP & SSH

7.1 | HTTP & SMTP

There are 4744 flows in total. The confusion matrix is shown in the Figure 6A. With the regression model, the number of the correctly classified flows is 4631.

The model's precision and recall identification are shown in Table 4. The *F*-score value is shown in Figure 7A.

It can be seen that, for the HTTP & SMTP protocol, the model's resolution is better in the case of large ratio values for which the *F*-score is above 95%, especially reaching 100% at 10:0. Of 451 flows with the ratio of 9:1, 449 were correctly identified while only 2 flows were wrongly identified as 8:2. The reason is that the features of the HTTP and SMTP protocols differ, so the larger the difference in the number of packets between the two protocols is, the easier it is to get the correct ratio value. For this reason, when the ratio value is small, the *F*-score value of the model will be relatively low. Of 468 flows with a ratio of 5:5, 436 were correctly identified while 29 flows were wrongly identified as 4:6 and the other 3 were wrongly identified as 1:9, 2:8, and 6:4, respectively. of 363 flows with a ratio of 4:6, 335 were correctly identified while 17 flows were wrongly identified as 5:5; 8 were wrongly identified as 3:7; 2 were wrongly identified as 2:8, and 1 was wrongly identified as 1:9.

7.2 | HTTP & SSH

There were 5192 flows in total. The confusion matrix is shown in Figure 6B. With the regression model, 4939 flows were correctly classified.

The model's precision and recall rate of identification are shown in Table 4. The *F*-score is shown in Figure 7B.

TABLE 4 Prevision rate and recall rate of the trained regression model

Ratio of the two mixed protocols	HTTP & SMTP		HTTP & SSH		SMTP & SSH	
	PR	RR	PR	RR	PR	RR
0:10	1.0000	0.9821	0.9978	1.0000	1.0000	1.0000
1:9	0.9757	0.9452	0.9953	0.9906	1.0000	0.9977
2:8	0.9785	0.9901	0.9390	0.9938	0.9977	0.9977
3:7	0.9810	0.9833	0.9631	0.9378	0.9933	0.9852
4:6	0.9203	0.9229	0.9615	0.9634	0.9815	0.9775
5:5	0.9625	0.9316	0.9376	0.9596	0.9844	0.9844
6:4	0.9845	0.9911	0.9657	0.9375	0.9598	0.9521
7:3	0.9974	0.9843	0.7893	0.9369	0.9680	0.9885
8:2	0.9951	1.0000	0.9471	0.7803	1.0000	1.0000
9:1	0.9825	0.9956	0.9980	0.9843	1.0000	1.0000
10:0	0.9571	1.0000	1.0000	1.0000	1.0000	1.0000

The model's performance for the HTTP & SSH protocol is poor compared to its performance for the HTTP & SMTP protocol. This is caused by the high similarity in the statistical features of the HTTP and SSH protocol flows. In particular, in the statistical features of the flow SSH's file download mode is approximately consistent with HTTP's behavior when obtaining web pages. Other modes, such as command control, differ from the features of HTTP. For this reason, the model performed very poorly at a scale of 7:3 and 8:2. Of 444 flows with the 7:3 ratio, 416 were correctly identified while 16 were wrongly identified as the 8:2 ratio and 12 were wrongly identified as the 6:4 ratio. Of 529 flows with the 8:2 ratio, 412 were correctly identified while 111 were wrongly identified as ratio 7:3. And 4 were wrongly identified as ratio 6:4 and the other two were erroneously identified as ratios 4:6 and 9:1. Of 480 flows with the 2:8 ratio, 477 were correctly identified while 2 were wrongly identified as ratio 1:9 and 1 was wrongly identified as ratio 3:7. For the 1:9 and 9:1 ratios, the model showed high performance, with an *F*-score of greater than 98%. Flows with a mixture ratio of 0:10 and 10:0 were all correctly identified and were not mistakenly allocated to other data flow categories.

7.3 | SMTP & SSH

There were 5474 flows in total. The confusion matrix of the classification results is shown in Figure 6C. With the regression model, 5419 flows were correctly classified.

The model's precision and recall of identification are shown in the Table 4. The *F*-score value is shown in Figure 7C.

For the SMTP & SSH protocol, the model demonstrated peak performance in the last three scale values. Most of the *F*-scores were above 99%. At ratios 0:10, 8:2, 9:1, and 10:0, none were wrongly identified. However, the model's identification performance of the first few scale values was relatively low. The poorest performance was approximately 0.95 when the ratio was 6:4. The cause was the relatively close SMTP and SSH protocol mixing ratio.

7.4 | The case of single protocol

DNS tunnels containing a single-protocol can be viewed as the particular case of the bi-protocol DNS tunnels namely, that the ratio of the two protocols is either 0:10 or 10:0. As proven in Figures 6 and 7, the proposed scheme can also be effective in the case of a single protocol being used.

When HTTP and SMTP protocols are combined in the tunnel with a mixing ratio of 0:10 (This is alternatively expressed as only the SMTP protocol being present), the precision and recall

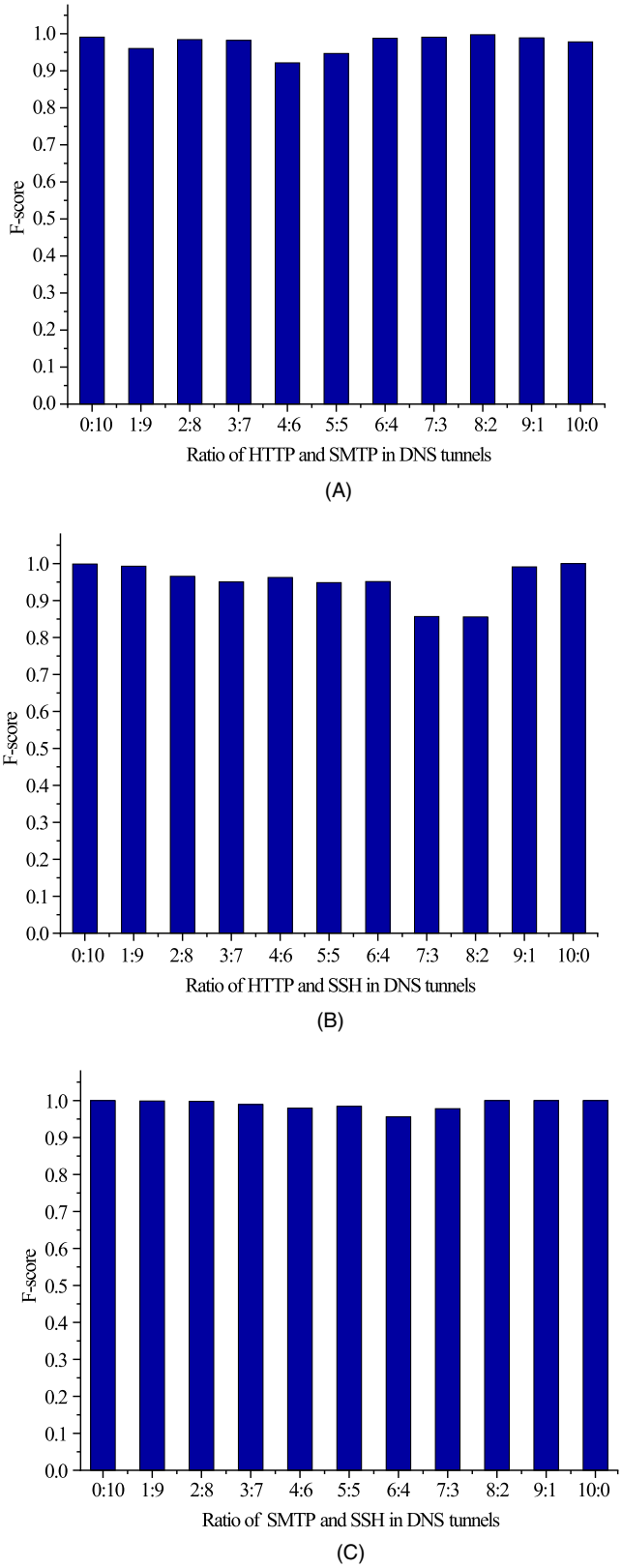


FIGURE 7 *F*-score of different ratios of hybrid HTTP & SMTP, HTTP & SSH and SMTP & SSH protocols in DNS tunnels: (A) HTTP & SMTP; (B) HTTP & SSH; (C) SMTP & SSH

rates are 100% and 98.2%, respectively. When HTTP and SMTP protocols are mixed in the tunnel with a mixing ratio 10:0 (This is alternatively expressed as only the HTTP protocol being present), the model's precision and recall rates are 95.7% and 100%, respectively. When HTTP and SSH protocols are mixed in the tunnel with a mixing ratio 0:10 (This is alternatively expressed as only the SSH protocol being present), precision rate and recall rate are 99.8% and 100%, respectively. When the HTTP and SSH protocols are mixed in the tunnel with mixing ratio 10:0 (This is alternatively expressed as only the HTTP protocol being present), the precision and recall rates are both 100%. When SMTP and SSH protocols are mixed in the tunnel with mixing ratio 0:10 (This is alternatively expressed as only the SSH protocol being present) the precision and recall rates are both 100%. When the SMTP and SSH protocols are mixed in the tunnel with mixing ratio 10:0 (this is alternatively expressed as only the SMTP protocol being present), the precision and recall rates are both 100%.

8 | CONCLUSIONS

In this study, to identify hybrid DNS tunnel traffic, we investigated a scheme that proposes to identify the type, and estimate the ratio, of two combined protocols. We started by extracting features from the request and response flows in DNS tunnels, and undertook the processes of feature screening, correlation analysis, and model training. In a real-world network environment, we use the three commonly employed protocols to generate hybrid DNS tunnel traffic containing two of the protocols, either HTTP, SMTP, or SSH in all possible combinations. According to the general criteria applied, the performance of the proposed method was evaluated using the hybrid DNS tunnel traffic. The experimental results show that the proposed scheme is effective for estimating the ratio of hybrid DNS tunnel traffic containing two mixed protocols.

However, the DNS protocol has a unique packet structure and information interaction mode. The extracted features can only be applied to DNS tunnels, and cannot be directly applied to other types of tunneling. For other l approaches, such as for HTTP and SSH tunnels, we will continue to carry out corresponding research work in the future.

In addition, the proposed method to identify the hybrid DNS tunnel traffic can only be applied to the two-protocol mixed case. When three or more protocols are mixed in a DNS tunnel, the estimated accuracy of the proposed method rapidly drops to approximately 0.6. The cases in which more protocols are combined in DNS tunnels will be studied in our future work.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant no. U1836104, 61702235, and 61921004, and partly supported by Fundamental Research Funds for the Central Universities under Grant no. 30918012204.

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

ORCID

Huiwen Bai https://orcid.org/0000-0002-8985-6977
Guangjie Liu https://orcid.org/0000-0003-4729-7406
Jiangtao Zhai https://orcid.org/0000-0001-8557-9899
Weiwei Liu https://orcid.org/0000-0001-7353-9136
Xiaopeng Ji http://orcid.org/0000-0001-6094-4626
Luhui Yang https://orcid.org/0000-0002-7065-6801

REFERENCES

1. M. Dusi et al., *Tunnel hunter: Detecting application-layer tunnels with statistical fingerprinting*, Comput. Netw. **53** (2009), 81–97.
2. Y. He, Y. Zhu, and W. Lin, *HTTP tunnel Trojan detection model based on deep learning*, J. Phys.: Conf. Series **1187** (2019), 1–11.
3. M. Dusi, F. Gringoli, and L. Salgarelli, *A preliminary look at the privacy of SSH tunnels*, in Proc. Int. Conf. Comput. Commun. Netw. (Thomas, VI, USA), 2008, pp. 1–7.
4. D. Raman et al., *DNS tunneling for network penetration*, in Proc. Int. Conf. Inf. Security Cryptology (Seoul, Rep. of Korea), 2012, pp. 65–77.
5. M. Zhang et al., *State of the art in traffic classification: A research review*, in Proc. PAM Student Workshop (Seoul, Rep. of Korea), 2009, 3–4.
6. J. Dietrich et al., *On botnets that use DNS for command and control*, in Proc. Eur. Conf. Comput. Netw. Defense (Gothenburg, Sweden), 2012, pp. 9–16.
7. I. Valenzuela, *Game changer: Identifying and defending against data exfiltration attempts*, in Proc. SANS Cyber Defense Summit (Nashville, TN, USA), 2015.
8. K. Born and D. Gustafson, *Detecting DNS tunnels using character frequency analysis*, 2010, arXiv preprint arXiv: 1004.4358.
9. K. Born and D. Gustafson, *Detecting DNS tunnels through n-gram visualization and quantitative analysis*, 2010, Ngviz: arXiv preprint arXiv: 1004.4359.
10. C. Qi et al., *A bigram based real time DNS tunnel detection approach*, Procedia Comput. Sci. **17** (2013), 852–860.
11. W. Ellens et al., *Flow-based detection of DNS tunnels*, in Proc. Int. Conf. Autonomous Infrastructure (Barcelona, Spain), 2013, pp. 124–135.
12. T. Cejka, Z. Rosa, and H. Kubatova, *Stream-wise detection of surreptitious traffic over DNS*, in Proc. IEEE Int. Workshop Comput. Aided Modeling Design Commun. Links Netw. (Athens, Greece), 2014, pp. 300–304.
13. I. Homem, P. Papapetrou, and S. Dosis, *Entropy-based prediction of network protocols in the forensic analysis of DNS tunnels*, 2017, arXiv preprintarXiv: 1709.06363.
14. M. Kara et al., *Detection of malicious payload distribution channels in DNS*, in Proc. IEEE Int. Conf. Commun. (Sydney, Australia), 2014, pp. 853–858.
15. L. Buczak et al., *Detection of tunnels in PCAP data by random forests*, in Proc. Annu. Cyber Inf. Security Res. Conf. (Oak Ridge, TN, USA), 2016, p. 16:1–4.
16. M. Aiello, A. Merlo, and G. Papaleo, *Performance assessment and analysis of DNS tunneling tools*, Logic J. IGPL, **21** (2013), 592–602.
17. M. Aiello, M. Mongelli, and G. Papaleo, *Basic classifiers for DNS tunneling detection*, in Proc. IEEE Symp. Comput. Commun. (Split, Croatia), 2013, pp. 880–885.

18. M. Aiello, M. Mongelli, and G. Papaleo, *Supervised learning approaches with majority voting for DNS tunneling detection*, in Proc. Int. Joint Conf. SOCO '14-CISIS'14-ICEUTE'14 (Bilbao, Spain), 2014, pp. 463–472.
19. M. Aiello, M. Mongelli, and G. Papaleo, *DNS tunneling detection through statistical fingerprints of protocol messages and machine learning*, Int. J. Commun. Syst. **28** (2015), 1987–2002.
20. J. Liu et al., *Detecting DNS tunnel through binary-classification based on behavior features*, in Proc. IEEE Trustcom/BigDataSE/ICESS (Sydney, Australia), 2017, pp. 339–346.
21. J. J. Davis and E. Foo, *Automated feature engineering for HTTP tunnel detection*, Comput. Security, **59** (2016), 166–185.
22. I. Homem and P. Papapetrou, *Harnessing predictive models for assisting network forensic investigations of DNS tunnels*, in Proc. Annu. ADFSL Conf. Digital Forensics, Security Law (Daytona Beach, FL, USA), 2017, pp. 79–93.
23. A. Almusawi and H. Amintoosi, *DNS tunneling detection method based on multilabel support vector machine*, Security Commun. Netw. **2018** (2018), pp. 1–9. 6137098

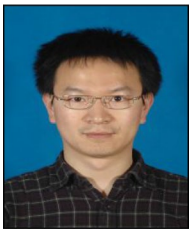
AUTHOR BIOGRAPHIES

Huiwen Bai received the BS degree in automation from the Nanjing University of Science and Technology, Nanjing, in 2015. He is currently pursuing the PhD degree in Nanjing University of Science and Technology. His research interests include deep learning and network security.

Guangjie Liu received the BS degree in information engineering and the PhD degree in control science and engineering from the Nanjing University of Science and Technology, Nanjing, in 2002 and 2007, respectively. From 2016 to 2017, he was a Visiting Scholar with the Department of Computer Science, University of California at Davis. He is currently a Professor with the Nanjing University of Information Science and Technology. His research interests are network and communication security.

Jiangtao Zhai received the BS degree in electrical and computer engineering and the MS and PhD degrees in control science and engineering from the Nanjing University of Science and Technology, Nanjing, in 2007, 2009, and 2013, respectively. He is currently an Associate Professor with the Nanjing University of Information Science and Technology. His research interests include multimedia communication and wireless sensor networks.

Weiwei Liu received the BS degree in automation and the PhD degree in control science and engineering from the Nanjing University of Science and Technology, Nanjing, in 2010 and 2015, respectively. From 2014 to 2015, he was a Visiting Scholar with the Department of Computer Science, University of California at Davis, Davis, CA, USA. He is currently an Assistant Professor with the School of Automation, Nanjing University of Science and Technology. His research interests include multimedia signal processing and network traffic analysis.



Xiaopeng Ji received the BS degree in electrical and computer engineering and the MS and PhD degrees in control science and engineering from the Nanjing University of Science and Technology, Nanjing, in 2007, 2009, and 2013, respectively. He is currently an Associate Professor with the Nanjing University of Information Science and Technology. His research interests include multimedia communication and wireless sensor networks.

Luhui Yang received the BS degree in automation from the Nanjing University of Science and Technology, Nanjing, in 2013. He is currently pursuing the PhD degree in Nanjing University of Science and Technology. His research interests include deep learning and network security.

Yuewei Dai received the BS and MS degrees in systems engineering from the East China Institute of Technology in 1984 and 1987, respectively, and the PhD degree in control science and engineering from the Nanjing University of Science and Technology in 2002. He is currently a Professor with the Nanjing University of Information Science and Technology. His research interests are in multimedia security, system engineering theory, and network security.