WENC: HTTPS Encrypted Traffic Classification Using Weighted Ensemble Learning and Markov Chain

Wubin Pan*†, Gaung Cheng*†, Yongning Tang‡
*School of Computer Science and Engineering, Southeast University, China
†Key Laboratory of Computer Network and Information Integration, Ministry of Education, China
‡School of Information Technology, Illinois State University, USA
Email:{wbpan, gcheng}@njnet.edu.cn, ytang@ilstu.edu

***Abstract*—SSL/TLS protocol is widely used for secure web applications (i.e., HTTPS). Classifying encrypted SSL/TLS based applications is an important but challenging task for network management. Traditional traffic classification methods are incapable of accomplishing this task. Several recently proposed approaches that focused on discriminating defining fingerprints among various SSL/TLS applications have also shown various limitations. In this paper, we design a Weighted ENsemble Classifier (WENC) to tackle these limitations. WENC studies the characteristics of various sub-flows during the HTTPS handshake process and the following data transmission period. To increase the fingerprint recognizability, we propose to establish a second-order Markov chain model with a fingerprint variable jointly considering the packet length and the message type during the process of HTTPS handshake. Furthermore, the series of the packet lengths of application data is modeled as HMM with optimal emission probability. Finally, a weighted ensemble strategy is devised to accommodate the advantages of several approaches as a unified one. Experimental results show that the classification accuracy of the proposed method reaches 90%, with an 11% improvement on average comparing to the state-of-the-art methods.**

***Keywords*-SSL/TLS; HTTPS services; HMM; Markov chain, weighted ensemble**

I. Introduction

With the increasing awareness of network security, SSL/TLS protocol (i.e., HTTPS) is widely used for secure web applications. HTTPS applications account for at least 29% of Web traffic, and will continue to increase [1].

Classifying encrypted SSL/TLS based applications is an important but challenging task for network management. However, traditional port-based and payload-based approaches [2], [3] cannot achieve fine-grained classification for HTTPS applications. Several recently proposed approaches that focused on discriminating defining fingerprints among various SSL/TLS applications have also shown various limitations.

To identify various HTTPS applications, Message Type has been used as the fingerprint to establish a first-order Markov chain model. Due to the similar handshaking process, this fingerprint itself is often insufficient for application classification. Correspondingly, the packet length of certificate has been introduced to increase the accuracy of application discrimination. However, there are cases that the packet lengths of certificates in different applications are clustered into the same class, especially with the increase of application types.

In this paper, we design a Weighted ENsemble Classifier (WENC) to tackle the limitations above. WENC studies the characteristics of various sub-flows during the HTTPS handshake process and the following data transmission period. To increase the fingerprint recognizability, we propose to establish a second-order Markov chain model with a fingerprint variable jointly considering the packet length and the message type during the process of HTTPS handshake. Furthermore, the series of the packet lengths of application data is modeled as HMM with optimal emission probability. Finally, a weighted ensemble strategy is devised to accommodate the advantages of several approaches as a unified one. The main contributions of this paper is as follows:

(1) To increase the recognizability of HTTPS application model, we propose a second-order Markov chain with a fingerprint variable jointly considering Message Type and Packet Length in HTTPS handshaking process.

(2) The emission probability is adjusted according to the relationship between the current packet size and adjacent packet size to optimize HMM model. A weighted ensemble classifier is proposed to improve the classification performance and its applicability.

The rest of paper is organized as follows. Section 2 briefly reviews the SSL/TLS protocol and related HTTPS service classification. Section 3 describes the design of HTTPS service classification method WENC. Section 4 shows the experiment results, and analyzes the performance of classification methods. Section 5 concludes the paper.

II. Related work

A. SSL/TLS background

SSL/TLS protocol, from the beginning to the end, experiences different states and takes different actions accordingly. The temporal sequence of states is the accurate reflection of temporal feature of the protocol. Fig.1 presents an example of message exchange between a client and a server during an SSL/TLS session.

The initial Hello message exchange between a client and a server establishes several protocol attributes in the following data transfer: Protocol Version, Session ID, Cipher Suite, and Compression Method. The key exchange uses up to four messages: server certificate, server key exchange, client certificate, and client key exchange. Then, the client sends Change Cipher

2324-9013/17 $31.00 © 2017 IEEE
DOI 10.1109/Trustcom/BigDataSE/ICESS.2017.219

IEEE computer society

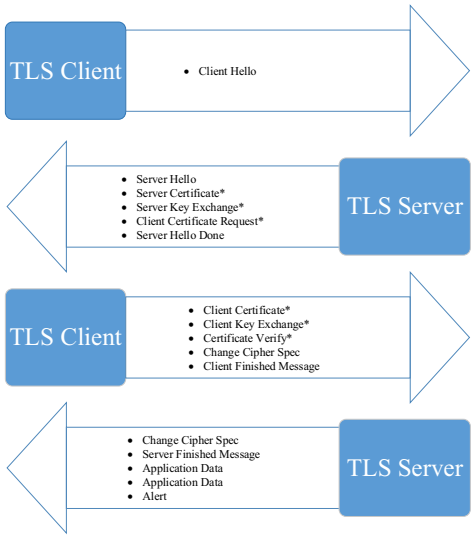

Fig. 1. Message exchange of SSL/TLS protocol

Spec, and the next Finished message is encrypted with the new algorithms and keys. In response, the server sends its own Change Cipher Spec message and the Finished message under the new cipher specification, which completes the SSL/TLS handshake and the two parties can exchange application data. The server terminates the session with an Alert message. Note that during the SSL/TLS handshake, most of information is sent as plaintext. However, after the Server Hello done or Change Cipher Spec protocol message, only the protocol type, the length of a record, and the SSL/TLS version are not encrypted.

SSL/TLS protocol can be divided into two sub-layers: the upper layer includes SSL handshake protocol, SSL change cipher spec protocol, and SSL alert protocol; the lower layer is SSL record protocol. The header of SSL record structure consists of three parts: content type, version, and length. The corresponding compact notations of commonly used message types are shown in Table I.

In the following discussion, we use compact notations instead of messages types. For instance, we represent the Application Data as 23: and the Server Hello message as 22:02. In our study, we consider only server-side message types of an SSL/TLS session. Modeling HTTPS application with the separation of client and server side observsable evidence helps tackling the problem of asymmetric routing when we can only observe traffic in one direction. Depending on different configurations, we expect slightly different characteristics on the model based on client side activities, whereas the service-side model should be representative in all networks.

B. Summary of SSL/TLS encrypted traffic classification

Classifying encrypted HTTPS traffic brings new challenges to network management. It is highly desirable to investigate effective classification methods for SSL/TLS based applications. In recent years, researchers have been focusing on

TABLE I
ENCODING SCHEME

Notation	SSL/TLS Message Type
20	Change Cipher Spec
21	Alert
22:02	Server Hello
22:11	Certificate
22:12	Server Key Exchange
22:13	Certificate Request
22:14	Server Hello Done
22:17	Encrypted Handshake Message
22:18	New Session Ticket
22:20	Finished
23	Application Data

encrypted traffic classification [4]. Kim [5] proposed a method to generate service signatures automatically from SSL/TLS payload data. They utilized the certificate publication information field in the certificate exchanging record of SSL/TLS traffic for the service signatures, which occurs when SSL/TLS performs Handshaking before encrypted transition. Bernaille [6] proposed a method based on the size of the first few packets of an SSL connection, which enables an early application identification with the accuracy of more than 85%. Sun[7] proposed a hybrid method to identify SSL/TLS encrypted applications with a combination of a signature-based and a flow-based statistical analysis scheme.

Shbair [8] proposed a robust technique to precisely identify the services run within HTTPS connections, i.e. to name the services, without relying on specific header fields that can be easily altered. They have defined dedicated features for HTTPS traffic that are used as input for a multi-level identification framework based on machine learning algorithms. Korczynski [9] proposed stochastic fingerprints for application traffic flows conveyed in SSL/TLS sessions. The fingerprints are based on first-order homogeneous Markov chains for which we identify the parameters from observed training application traces. As the fingerprint parameters of chosen applications considerably differ, the method results in a very good accuracy of application discrimination. Their analysis of the results reveals that obtaining application discrimination mainly comes from incorrect implementation practice, the misuse of the SSL/TLS protocol, various server configurations, and the application nature. After that, author in [10] proposed a certificate-aware encrypted traffic classification method based on second-order Markov chain. To increase the diversity of application fingerprints, they developed a new model by incorporating the certificate packet length clustering into the second-order homogeneous Markov chains. Results show that the proposed method lead to a 30% improvement on average. But the method fails in some case while the packet lengths of application Certificate are subject to the same cluster.

To address the limitations above, this paper proposes an ensemble classification method based on sub-flows during the HTTPS handshake process and the following data transmission period. The joint feature of message type and packet length in HTTPS handshake process is used to establish a Markov chain model. The packet size sequence of data transition

(Application Data) is used to establish an HMM model. The weighted ensemble classifier is built to improve its overall applicability and performance.

III. WENC TRAFFIC CLASSIFICATION METHOD

A. Motivation

We adopt first-order and second-order Markov chains to construct fingerprint models for classifying HTTPS applications as in [9], [10] . Although these fingerprint models contribute to classifying encrypted traffic, they may fail in some cases due to several reasons. Firstly, these models only consider two discrete transition states, which cannot fully describe distinguishable features of different HTTPS applications. State transitions due to a specific sequence of SSL/TLS message types of an HTTP application provides an observable evidence or fingerprint of the application. Since the overlap of the fingerprints from different HTTPS applications could be high, high fingerprint matching probability may still lead to a wrong classification result. Secondly, model state based on a single variable (i.e., message types of SSL/TLS session) provides limited expressiveness, and is prone to lead to low discrimination of fingerprints.

Although the method [10] improves classification accuracy in most instances, it still could result in misclassification with small probability. Since the packet lengths of certificates from different applications could be clustered together, misclassification is unavoidable under the maximum likelihood criterion.

Solution Overview. We intend to address the above problems in three aspects as follows.

1. We use second-order Markov chain model instead of first-order one, in which the current state is not only related to the previous state, but also the state before previous one. In this way, we intend to address the limited expressiveness of first-order Markov chain on application fingerprints.

2. Observing the correlation between the packet length in a handshake process and its corresponding application, we combine both Message Type (MT) and Packet Length (PT) as a joint traffic feature $\langle MT, PT \rangle$ to enrich the state expressiveness on application fingerprints.

3. To further enhance the robustness of the application classifier, we adopt Hidden Markov Model (HMM) to incorporate the observable but indirect evidence Packet Length of Application Data (ADPT) into the reasoning process. Apparently, ADPT is always related to the behavior of the corresponding application. However, ADPT by itself is ambiguous to determine the type of applications. Properly adopting it into the reasoning process is a challenge. In our system, we combine HMM with the Markov chain model to construct a robust application classifier via weighted ensemble learning.

B. Classification System Based on Weighted Ensemble Learning

The architecture of the weighted ensemble learning-based HTTPS application classification system is shown in Fig.2, which consists of the following functional modules: traffic preprocessing, learning process, and classification process.

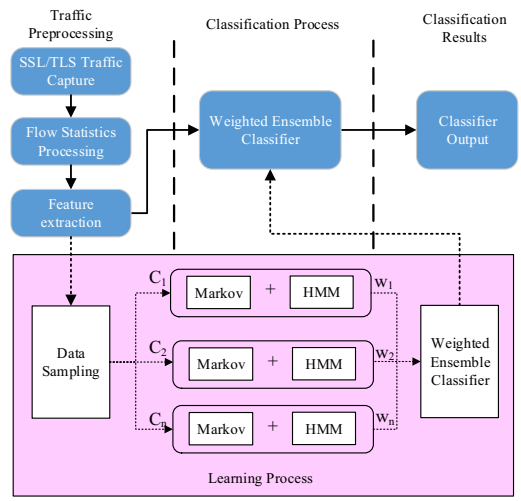

Fig. 2. System architecture

The traffic preprocessing module captures SSL packets and pairs the packets of the same session into the same flow, and then obtain the flow features, which is ready for the learning model construction. The main function of the learning process module is to establish an ensemble classification model based on the selected flow feature sets of the SSL flows, the Markov and HMM models are built on the sub-flows of handshake process and data transition process respectively, and then combine the models by weighted ensemble learning to construct the weighted ensemble classifier for classification process. Classification process module is to label the SSL/TLS flows based on the constructed weighted ensemble model.

C. Model Improvement

In this section, we mathematically formulate the process of modeling based on the second-order Markov chain model and a weighted ensemble with the reasoning result of HMM based on the evidence of ADPT.

1) Markov Chain Model built on Handshake Process: In this paper, we propose a technique for traffic classification based on a statistical approach that takes into account some new properties of network traffic, trying to offer a contribution in terms of techniques to increase the discrimination by different network applications. Indeed, as explained in the following sections, the use of the second-order Markov Chain model allows us to take into account the joint feature $\langle MT, PT \rangle$, as well as their temporal correlation. We use studies from the work based on Markov Chain[9]: the traffic generated by a specific application is modeled as a flow of packets, seen as a sequence of $\langle MT, PT \rangle$ pairs generated according to different distributions depending on the hidden state of the source.

An approach in [11] presented two separate classifiers working separately on Inter Packet Time (IPT) or on Packet Size (PS), and a left-to-right structure for the state topology of HMM is used. However, a proposal for extending their approach was later presented in a technical report [12], where

they try to account for joint IPT and PS modeling via vector quantization. The joint feature is designed for one-dimensional observable variables. The authors in [12] present a very complex state structure depending on the length of the training sequence, with a pair of different states for each packet. Compared to [12], our model works directly on a two-dimensional observable variable $\langle MT, PT \rangle$, without requiring any pre-processing like vector quantization.

2) Second-order Homogeneous Markov Model: This paper proposes a second-order homogeneous Markov chain to describe a SSL/TLS session based on the joint feature of message type and packet length. Assuming discrete-time random variable X_t for any $t = t_0, t_1, ..., t_n \in T$, the corresponding values $i_t \in \{1, 2, ..., s\}$, where i_t is a joint feature (Message Type(MT) and packet length (PT) $\langle MT, PT \rangle$) or a sequence of integration feature, the vector of integration feature are selected as the state for Markov model. Assuming X_t is a second-order Markov chain, considering using the first two states to estimate the current state.

$$\begin{aligned} &P(X_t = i_t | X_{t-1} = i_{t-1}, X_{t-2} = i_{t-2}, ..., X_1 = i_1) \\ &= P(X_t = i_t | X_{t-1} = i_{t-1}, X_{t-2} = i_{t-2}) \end{aligned} \tag{1}$$

Assuming that the second-order Markov chain is homogeneous, a state from time $t-2$ and $t-1$ transmitted to time t is time-invariant:

$$\begin{aligned} &P(X_t = i_t | X_{t-1} = i_{t-1}, X_{t-2} = i_{t-2}) \\ &= P(X_t = k | X_{t-1} = j, X_{t-2} = i) = p_{i-j-k} \end{aligned} \tag{2}$$

Therefore, the transition matrix is expressed as follows

$$P = \begin{bmatrix} p_{1-1-1} & p_{1-1-2} & \cdots & p_{1-1-s} \\ & & \cdots & \\ p_{1-s-1} & p_{1-s-2} & \cdots & p_{1-s-s} \\ \vdots & \vdots & \ddots & \vdots \\ p_{s-s-1} & p_{s-s-2} & \cdots & p_{s-s-s} \end{bmatrix} \tag{3}$$

Where: $\sum_{k=1}^{s} p_{i-j-k} = 1$. We define the ENter Probability Distribution (ENPD) to enter the second-order Markov chain of the first two states:

$$Q = [q_{1-1}, ..., q_{1-s}, q_{2-1}, ..., q_{2-s}, ..., q_{s-s}] \tag{4}$$

Where $q_{i-j} = P\left(X_{t+1\,=\,j}, X_{t\,=\,i}\right)$ at time t_0. We define EXit Probability Distribution (EXPD) to exit the second-order Markov chain of the last two states:

$$W = [w_{1-1}, ..., w_{1-s}, w_{2-1}, ..., w_{2-s}, ..., w_{s-s}] \tag{5}$$

Where w_{i-j}=$P\left(X_{t+1\,=\,j}, X_{t\,=\,i}\right)$ represents the probability that the session finishes when it is in state i at time t_n. Based on these definitions, the probability of representing the SSL/TLS session is as follows:

$$P(\{X_1, ...X_T\}) = q_{i1-i2} * \prod_{t=3}^{T} p_{i_{t-2}-i_{t-1}-i_t} * w_{i_{T-1}-i_T} \tag{6}$$

The resulting probability indicates how the joint feature during a SSL/TLS session is close to a model of an application flow: a larger value means that the SSL/TLS session is closer to the model.

3) HMM built on ADPT : **Definition 1** $V = \{v_1, v_2, ..., v_M\}$ represents the set of ADPT feature values in the case of discretization, where M is the total number of ADPT values.

The set of network operating states U is regarded as the set of hidden states of the HMM model, $O = (O_1 = o_1, ..., O_M = o_M)$ is the observable state of the output and M is the number of observations in the sequence. $A_{ij} = \{a_{ij}\}$ represents the probability vector of network operating state transition, $1 \leq i, j \leq N$ (N is the number of states), $a_{ij} = p(u_j/u_i)$ represents the probability transmitted from the state q_i to the state u_j, $\sum_{j=1}^{N} a_{ij} = 1$, $1 \leq i \leq N$. $B = \{b_{im}\}$ represents the probability of the corresponding ADPT output value obtained from the network operation at a certain time, $b_{im} = P(v_m/u_i)$ represents the probability of outputting the characteristic value v_m at the given state u_i. Randomly generated $\pi = \{\pi_i, i = 1, ..., N\}$, $\sum_{i=1}^{N} \pi_i = 1$ is the initial probability distribution for the network operating state, then the model built by the network operation state and ADPT characteristics state constructs the HMM model.

The corresponding HMM of a given SSL/TLS session is expressed as H_{app}. The set $F = \{F_1, F_2, ..., F_l\}$ represents the lengths of consecutive l flows generated by this application, which is used as the training sample set of H_{app}. We choose ADPT as the feature to build H_{app}. For a unknown flow $F_i = \{g_1, g_2, ...g_r, ...\}$, $\lambda_{app} = P(F_i|H_{app})$ represents the probability that the application app identified as F_i. As it can be clearly observed that the current ADPT is intrinsically linked with the adjacent ADPT. Due to the standard mechanisms for SSL/TLS data transmission, such as congestion control, reordering, and retransmission, the ADPT possesses the characteristic of periodicity. We could adjust the emission probability based on the adjacent ADPT to optimize the HMM.

4) HMM emission probability optimization: Since the HMM parameters include transition probability and emission probability, location information of packet size in the state can only be reflected by the parameter of HMM, since related to the ADPT, therefore, it can be reflected by the emission probability.

A modified emission probability function is proposed to describe the probability of adjacent packet size to adjust the emission probability. Modify the confusion matrix $B = \{b_{im}(o_{t-1})\}$, $1 \leq i \leq N$, $1 \leq m \leq M$, where o_{t-1} represents the output of previous observation state, describes the probability of the observable characteristics v_m at time t, given the observable characteristics of a state o_{t-1} and hidden state u_i. Based on the dependencies between the before and after states, the emission probability $b_{im}(o_{t-1})$ of current state is determined by the emission probability function, then use the transition probability matrix to calculate the system implicit state transition probability of the next time. Compared to other models, our model is beneficial to simplify the higher-order dependencies to improve the accuracy.

In this paper, we use different emission probability calculation methods depending on the position of the ADPT [13]. Specifically, the ADPT is different depending on the state

of the position and the previous ADPT, choose a different emission probability, the formula is

$$b_{im}(o_{t-1}) = \begin{cases} \frac{c(P(u \nwarrow \sigma))}{\sum_{\rho \in \Sigma} c(P(u \nwarrow \rho))}, & \sigma \neq \sigma_{-1} \\ \frac{c(P(u \uparrow \sigma_{-1} \sigma))}{c(P(u \uparrow \sigma_{-1}))}, & \sigma = \sigma_{-1} \end{cases} \tag{7}$$

Where σ_{-1} represents the previous ADPT of σ. $u \uparrow \sigma$ represents state u emit ADPT σ, $u \nwarrow \sigma$ represent σ is the first ADPT of state u, $u \nearrow \sigma$ represents state u emit ADPT σ and σ is not the first ADPT of state u, P is the emission probability, so that $P(u \uparrow \sigma)$=P($u \nwarrow \sigma$)+P($u \nearrow \sigma$). $C(x)$ represents the number of frequency of events x in training set, in order to calculate $c(x)$, the model needs to be saved some state information in the process of consolidation, including the frequency and sequence relationship of each feature values appeared in status.

D. Weighted ensemble classifier

In the following, we propose a novel weighted ensemble classifier combined with Markov chain and HMM. Next, we show how to weigh the classifiers in ensemble [14].

The incoming flows are divided into a set of chunks S_i ($i \in 1, 2, \ldots, n$) in a time series, and the size of each chunk is identical. Each chunk S_i is used to build a classifier C_i. We weigh each classifier C_i reversely proportional to the error of C_i. The weight of classifier C_i is obtained by computing its error on test samples. The classifier built on the latest training samples is close to the distribution of the current test samples. Thus, we can get weights of classifiers approximately by measuring the error on S_n.

Specifically, the error of test samples (x, c) on C_i is $1 - f_c^i(x)$, where $f_c^i(x)$ is the probability given by C_i and x is an instance of class c. Thus, the mean square error of classifier C_i is:

$$MSE_i = \frac{1}{|S_n|} \sum_{(x,c) \in S_n} (1 - f_c^i(x))^2 \tag{8}$$

Suppose that we have a random guess – the chance of X being classified as Y equals to $Y's$ probability distribution $P(Y)$ over the entire class space C:={all classes}. A random classifier classifying instances in such way has the mean square error as follows:

$$MSE_r = \sum_c p(c)(1 - p(c))^2 \tag{9}$$

For example, a binary class with uniform class distribution, the random guess on this classification task will have 0.25 mean square error. Note that for a determined dataset, the MSE_r is a fixed value. WENC method guarantees to select base classifiers perform at least better than a random guess, the weight $w(i)$ of a base classifier i is calculated as follows:

$$w_i = MSE_r - MSE_i \tag{10}$$

If the classifier t performs not better than a random guess, its weight will be set to 0, making it not to be used in ensemble. Also this setup ensures that if error is bigger, the weight would be smaller. The final ensemble result could be described as:

$$H(x_t) = \arg\max_{y \in Y} \sum_{i=1}^{n} w_i H_i^y(x_t) \tag{11}$$

E. WENC Algorithm

Algorithm 1 summarizes the steps of WENC algorithm. The algorithm is divided into two parts. The first part is to build the weighted ensemble classifier (lines 1-6). Line 2 builds classifiers h_i based on different feature attributes; line 3 calculates the error rate of h_i; line 4 calculates the error rate of random classifier; line 5 computes the weight of each classifier, and determine the weighted classifier. The second part describes the classification process (lines 7-10), which outputs the final classification results based on maximum confidence with weight.

Algorithm 1 WENC

1: **Input** : $labeled\ samples\ L; unlabeled\ sample\ U;$
2: **Output** : $classification\ result;$
3: **for** **do**$Attr_1\ and\ Attr_2 \in L$
4: $h_i \leftarrow combine(Markov(Attr_1) + HMM(Attr_2))$
5: $MSE_i = error\ rate\ of\ h_i\ via\ the\ cross\ validation$
$MSE_i = \frac{1}{|S_n|} \sum_{(x,c) \in S_n} (1 - f_c^i(x))^2$
6: $MSE_r = error\ rate\ of\ the\ random\ classifiers$
$MSE_r = \sum_c p(c)(1 - p(c))^2$
7: $w_i = MSE_r - MSE_i$
8: **end for**
9: **while** $flow\ U(x_t, y_t)\ is\ available$ **do**
10: $get\ classifiers\ output\ \forall_y \forall_i \{H_i^y(x_t)\} \in [0, 1]$
11: $Output\ H(x_t) = \arg\max_{y \in Y} \sum_{i=1}^{n} w_i H_i^y(x_t)$
12: **end while**

IV. EVALUATION

In this section, we discuss the evaluation methodology and the corresponding performance results.

A. Preliminary

(1) **Methods in Comparison**. In order to present a comprehensive evaluation on WENC, we compare it to the following three state-of-the-art work First-Order Markov chain fingerprint (FOM), which is analogous to the method for encrypted traffic classification in [9].

(2) Second-Order Markov chain fingerprint (SOM), which replaces the First-Order Markov chain in FOM with the Second-Order Markov chain.

(3) Second-Order Markov chain on two-dimensional feature (TSOM), which replaces the Message Type in SOM with the feature of $\langle MT, PT \rangle$. Similarly, TFOM is the First-Order Markov chain of TSOM.

(4) Second-Order Markov chain fingerprint considering Certificate packet length (SOCRT), which replaces the Second-Order Markov chain fingerprint with Certificate packet length [10].

TABLE II
PARTICULAR STRINGS IN DOMAIN NAMES OF SPECIFIC APPLICATIONS

Application	Strings
Alipay	*.alipay.com ,*.alipayobjects.com
Amazon	*.amazon.cn, *.amazon.com
Baidu	*.baidu.com,*.bdstatic.com
Dropbox	*.dropbox.com ,*.dropboxstatic.com
Facebook	*.facebook.com
Google	*.google.com, *.google.cn
pinterest	*. pinterest.com, *.pinimg.com
qqmail	*.qqmail.com, *.mail.qq.com
tumblr	*. tumblr.com
twitch	*. twitch.tv, ttvnw.net
twitter	*.twitter.com, *.twimg.com
YouTube	*.youtube.com, *.googlevideo.com, .ytimg.com

TABLE III
HTTPS SERVICE STATISTICS

Application	Campus1		Campus2	
	Flows	Packets	Flows	Packets
Alipay	11362	154120	13297	170132
Amazon	15907	201731	13208	182872
Baidu	9362	182582	8219	145281
Dropbox	9485	116436	10781	191452
Facebook	10275	187704	8219	21864
Google	10579	171784	9346	20975
Pinterest	11380	231758	10142	251628
Qqmail	12138	142776	9623	136225
Tumblr	9287	138568	7821	129726
Twitch	16024	196395	12713	268207
Twitter	7946	167451	8727	173741
YouTube	15290	253326	12032	312893

Datasets. We capture traffic flows from several selected border routers between a campus network and the Internet, and execute additional data pre-processing (e.g. wipe out non-SSL/TLS encrypted traffic) to generate the datasets. The Campus1 and Campus2 datasets come from two specific routers located in different sites in the same campus network. Both datasets contain one-week long traffic traces starting on June 13-19, 2016, where Campus1 dataset includes 139035 flows (2144631 packets) and Campus2 dataset includes 124128 flows (2004996 packets).

Ground Truth. Ground truth is a standard to verify methods for traffic classification. In this paper, we establish the ground truth by two steps. The first step is to find domain name by Fiddler. In this process, we utilize an open web plug-in Fiddler, which is a tool for searching the corresponding domain name of a given application. The second step is to extract and analyze some particular strings so that we can confirm which application specific flows belong to. The approach to establish ground truth cannot apply to those applications that we are unable to collect the signatures of their domain names. Table II shows several particular strings in domain names identified specific applications.

WENC is universal and valid under any circumstance, since it only requires the information from indispensable SSL/TLS handshaking sessions and ADPT. To verify effectiveness for our method, we need to know ground truth in advance. So we select some limitations of application in our experiment evaluation process, we choose the common SSL/TLS applications that easy to establish ground truth by Fiddler and domain name signature resolution. Table III shows the number of flows and packets of each application in the training datasets. For each training dataset, we capture at least thousands of flows and about ten thousands of packets to ensure the necessary fingerprints in our modeling can cover all features of corresponding applications, including video, webcast, mail, search, online payment, social networking, storage, and other commonly used Web applications.

Cross-validation. For the purpose of validating application fingerprints that created by WENC, we use two heterogeneous datasets (Campus1 and Campus2) to establish a 2-fold cross-validation. More specifically, when we use Campus1 as training dataset, Campus2 dataset is used to validate. Similarly, when Campus2 is used as training dataset, Campus1 is used for validation.

B. Evaluation metric

We consider four performance metrics to evaluate the effect of different methods: accuracy, precision, recall and F-Measure, as defined below:

$$accuracy = \sum_{i=1}^{m}(TP_i)/\sum_{i=1}^{m}(TP_i + FN_i) \tag{12}$$

$$precision = TP_i/(TP_i + FP_i) \tag{13}$$

$$recall = TP_i/(TP_i + FN_i) \tag{14}$$

$$F\text{-}Measure = \frac{2 \times presion \times recall}{presion + recall} \tag{15}$$

Here, n is the number of flow samples, and m is the number of application types. n_{ij} represents the number of samples which the actual application type i is labeled as type j. TP represents the number of truly labeled samples which the actual type is i, $TP_i = n_{ii}$. FN represents the number of samples which the actual type i is mistakenly identified as other types, $FN_i = \sum_{j \neq i} n_{ij}$. FP represents the number of samples which actual type non-i was mistakenly identified as type i, $FP_i = \sum_{j \neq i} n_{ji}$.

C. Evaluation

1) Accuracy: To validate the accuracy of WENC, we compared it with other three algorithms, including SOCRT, SOM and FOM. We conducted the validation on Campus 1 and Campus 2 datasets served for training and validation alternatively. Campus1 bar represents the accuracy that Campus 1 as training dataset and Campus 2 as testing dataset, while the Campus 2 bar is on the contrary.

Since WENC combines the advantages of the weighted classifiers with better applicability and distinguishability, WENC outperforms the other three approaches in terms of classification accuracy as shown in Fig.3. In addition, the performance of fingerprint methods (SOM and FOM) perform poorly, because the distinguishability of selected fingerprint

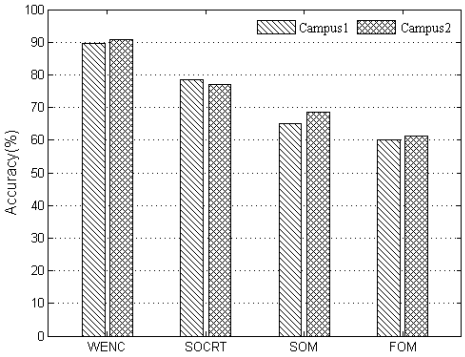

Fig. 3. Classification accuracy

is not sufficient. Although SOCRT considers the length of certificates, which is better than SOM, but the improved distinguish ability is still not satisfactory.

2) F-Measure: A good classification algorithm not only requires high accuracy, but also should have high precision and recall on each identified application. With the uneven distribution of samples of each application, it is particularly important to validate the Precision and Recall of WENC on each application. F-Measure of each algorithm are shown in Fig.4.

Clearly, WENC has better F-Measure on each application due to its robust applicability from the weighted classifier.

Although WENC presents good accuracy on traffic classification in most instances, it still results in misclassification with small probability. The classification methods can cause misclassification as shown in the following situations: (1) Non-standard implementation of SSL/TLS protocol: many protocol implementations do not follow the RFC specifications and behave slightly different from common SSL/TLS stacks. (2) Server configuration: some SSL/TLS protocol messages are defined as optional or context-dependent. And the SSL/TLS server communication parameters can be configured and possibly changed over time (3) Misuse of the SSL/TLS protocol: SSL/TLS tunneling is increasingly used as a tool for bypassing restrictions set by network configuration and security checks instead of using SSL/TSL for enforcing security.

However, the overall misclassification probability similar to the cases above is low. In general, WENC is applicable to most applications.

3) Effect of the feature $\langle MT, PT \rangle$: In order to validate the effectiveness of second-order Markov method on two-dimensional feature $\langle MT, PT \rangle$ against the one-dimensional feature method, we compare TSOM with SOM, and also compare the first-order Markov with second-order Markov by TSOM and TFOM, the results are performed when Campus 1 as training dataset and Campus 2 as testing dataset, shown in Table IV.

Due to its strong expressiveness, it is clear that the accuracy resulting from the methods TSOM and TFOM based two-dimensional feature is better than SOM and FOM. Comparing with SOM, TSOM is better on each application, where accuracy rises about 20%, because the two-dimensional feature increase the discrimination of applications. Comparing with TFOM, TSOM is better almost on each application, where accuracy rises about 4%, because the second-order Markov chain can better depict the state transition of SSL/TLS sessions.

TABLE IV
EFFECT OF THE TWO-DIMENSIONAL FEATURE

Application	TSOM	TFOM	SOM	FOM
Alipay	95.92	92.17	83.93	72.01
Amazon	83.49	76.48	58.73	47.78
Baidu	90.47	82.96	60.05	43.42
Dropbox	87.51	81.24	56.73	52.45
Facebook	85.32	75.92	42.22	45.98
Google	84.26	80.19	69.44	77.23
Pinterest	87.33	83.25	68.32	48.54
Qqmail	92.17	91.21	81.95	80.92
Tumblr	81.95	83.65	59.02	49.18
Twitch	75.73	72.37	67.12	66.61
Twitter	90.12	92.51	64.94	66.06
YouTube	85.96	83.97	63.59	66.41

TABLE V
TABLE V. EFFECT OF WEIGHTED ENSEMBLE LEARNING

Application	WENC	VENC	TSOM	HMM	SOCRT
Alipay	95.86	94,23	95.92	97.35	86.63
Amazon	88.81	85.13	83.49	86.41	76.33
Baidu	97.54	92.8	90.47	81.98	80.93
Dropbox	90.41	89.62	87.51	92.61	80.13
Facebook	91.03	87.18	85.32	94.32	74.86
Google	86.57	85.92	84.26	92.76	70.03
Pinterest	85.35	82.11	87.33	80.24	79.20
Qqmail	96.62	89.69	92.17	99.35	82.05
Tumblr	85.57	78.13	81.95	79.89	80.88
Twitch	81.10	80.21	75.73	75.94	66.02
Twitter	90.90	86.58	90.12	86.14	86.99
YouTube	89.05	82.63	85.96	82.97	83.83

4) Effect of Weighted Ensemble Learning: We compared our weighted classifier WENC with VENC and SOCRT, VENC presents the voted ensemble method on TSOM and HMM, TSOM represents Markov chain on second-order and two-dimensional feature, HMM represents the accuracy of HMM based on ADPT, the results are shown in Table IV.

It is clear that weighted classifier can obtain a better accuracy, reaches 90%. The accuracy of WENC method is higher than the voted ensemble classifier VENC, because the accuracy-based weight can improve the role of high-accuracy classifier. Also, WENC integrates the discrimination of TSOM and HMM methods. Moreover, WENC outperforms SOCRT because the certificates packet size from different applications may be easily clustered into the same category.

D. Efficiency

A timely classification system can better predict abnormal network behavior and take promptly and accurate response. To test the efficiency of the classification method in WENC, the following experiment was repeated 30 times and the corresponding results were averaged. Table VI describes the classification time of WENC algorithm when the samples range between 2000-64000, and compared with other methods.

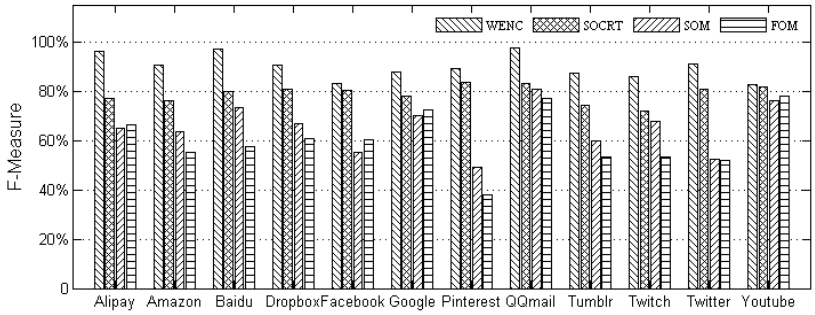

Fig. 4. F-Measure

TABLE VI
EFFICIENCY(MS)

method	Classification time					
	2000	4000	8000	16000	32000	64000
WENC	92	151	240	490	950	1806
SOCRT	46	70	97	155	273	521
SOM	22	16	31	43	80	152
FOM	14	12	18	22	36	62

As shown in Table VI, the consumed time for classification increases with the number of samples increases. WENC algorithm takes roughly 0.092-1.8 seconds, significantly higher than other methods. This is because WENC algorithm ensembles the results of two classifiers. Furthermore, parallel computing can be adopted to improve the efficiency of WENC.

V. Conclusion

Currently, SSL/TLS protocol is widely used for various web applications. In this paper, we propose an HTTPS application classification method using weighted ensemble learning. Second-order Markov model is built based on the two-dimensional joint feature $\langle MT, PT \rangle$ of SSL/TLS traffic. HMM model is constructed based on ADPT, packet length of transition (Application Data), in which HMM emission probabilities are further optimized based on the relationship of the length between packets. Finally, a weighted ensemble classifier is constructed to improve its applicability. The evaluation shows that WENC achieves better classification performance, superior to several state-of-the-art approaches. Our future work will mainly focus on: (1) further improvement on its efficiency to accommodate the need of traffic classification on high-speed network; (2) incorporating the function of abnormal traffic detection in WENC.

Acknowledgment

This work was supported by the National High Technology Research and Development Program (863 Program) (2015AA015603), the Prospective Research Programs Future Internet of Jiangsu Province (BY2013095-5-03), the Six talent peaks of high level Talents Project of Jiangsu Province (2011-DZ024); the Fundamental Research Funds for the Central Universities and the Research and Innovation Project for College Graduates of Jiangsu Province (KYLX15_0118).

References

[1] Sandvine, "Global Internet Phenomena Spotlight: Encrypted Internet Traffic," http://www.internetphenomena.com/2015/04/global-internet-phenomena-spotlight-encrypted-internet-traffic/, 2017, [Online; accessed 19-January-2017].

[2] M. Finsterbusch, C. Richter, E. Rocha, J.-A. Muller, and K. Hanssgen, "A survey of payload-based traffic classification approaches," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1135–1156, 2014.

[3] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[4] W. Pan, G. Cheng, X. Guo, and S. Huang, "Review and perspective on encrypted traffic identification research," *Journal on Communications*, vol. 37, no. 9, pp. 154–167, 2016.

[5] S.-M. Kim, Y.-H. Goo, M.-S. Kim, S.-G. Choi, and M.-J. Choi, "A method for service identification of ssl/tls encrypted traffic with the relation of session id and server ip," in *Network Operations and Management Symposium (APNOMS), 2015 17th Asia-Pacific*. IEEE, 2015, pp. 487–490.

[6] L. Bernaille and R. Teixeira, "Early recognition of encrypted applications," in *International Conference on Passive and Active Network Measurement*. Springer, 2007, pp. 165–175.

[7] G.-L. Sun, Y. Xue, Y. Dong, D. Wang, and C. Li, "An novel hybrid method for effectively classifying encrypted traffic," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. IEEE, 2010, pp. 1–5.

[8] W. M. Shbair, T. Cholez, J. Francois, and I. Chrisment, "A multi-level framework to identify https services," in *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*. IEEE, 2016, pp. 240–248.

[9] M. Korczyński and A. Duda, "Markov chain fingerprinting to classify encrypted traffic," in *Infocom, 2014 Proceedings IEEE*. IEEE, 2014, pp. 781–789.

[10] M. Shen, M. Wei, L. Zhu, M. Wang, and F. Li, "Certificate-aware encrypted traffic classification using second-order markov chain," in *Quality of Service (IWQoS), 2016 IEEE/ACM 24th International Symposium on*. IEEE, 2016, pp. 1–10.

[11] C. Wright, F. Monrose, and G. M. Masson, "Hmm profiles for network traffic classification," in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*. ACM, 2004, pp. 9–15.

[12] C. V. Wright, F. Monrose, and G. M. Masson, "Towards better protocol identification using profile hmms," *JHU Tech. Rep. JHU-SPAR051201*, 2005.

[13] M. Zhang, P. Yin, Z.-H. Deng, and D.-Q. Yang, "Svm+ bihmm: a hybrid statistic model for metadata extraction," *Journal of software*, vol. 2, p. 018, 2008.

[14] R. Wang, L. Shi, and B. Jennings, "Ensemble classifier for traffic in presence of changing distributions," in *Computers and Communications (ISCC), 2013 IEEE Symposium on*. IEEE, 2013, pp. 000 629–000 635.