Encrypted Live Streaming Channel Identification with Time-sync Comments

MINGKAI WANG¹, XIANGDONG TANG¹, FEI CHEN¹, AND QIAN LU¹
¹College of Computer Science & Technology, Qingdao University, Qingdao, Shandong Province 266071, China.
e-mail: (wmkqua@163.com), (qdutxd@163.com), (feic@qdu.edu.cn), (luqian0203@gmail.com).
Corresponding author: Fei Chen (e-mail: feic@qdu.edu.cn).

ABSTRACT The time-sync comments have been prevalent in modern live streaming systems to provide a real-time interaction experience for viewers. Whereas, the time-sync comments traffic can also act as a delicate fingerprint of encrypted live channels, leading to potential risks of privacy leakage. Most of previous video channel identification strategies with video bitrate-based fingerprint presume strict requirements on the implementation environments, which often assume that there is no interference from irrelevant traffic flows or network conditions. However, the time-sync comments sessions are distinct from other irrelevant traffic flows, and the traffic pattern is resilient to various network conditions, e.g., bandwidth limitation and transmission delay. In this paper, we design a system for encrypted live channel identification with time-sync comments traffic analysis. Specifically, both the inter-application and inner-application traffic filters are proposed to eliminate the irrelevant traffic flows, respectively. Further, a comment rate estimation method is developed through investigation of relationship between comment number, comment length and packet length. Finally, the dynamic time warping algorithm is improved for similarity matching in delay tolerant environment. In order to evaluate the system performance, we setup the prototype system with AWS EC2 server and utilize the real world trace data from Youtube and BiliBili. The experimental results show that the accuracy of the filter can reach 93.2%, and the accuracy of the comment rate estimation method can reach up to 91%. The match accuracy between fingerprint and comment rate can reach 92.1% within 200 seconds eavesdropping, which is 2% higher than using bitrate fingerprint and traffic pattern in the latest research, and can be increased to 98.2% when the eavesdropping time extends to 500 seconds.

INDEX TERMS live streaming, time-sync comments, privacy leakage, traffic analysis

I. INTRODUCTION
Live streaming service with time-sync comment has been an emerging killer application in recent years, which quickly sweeps across the world, in terms of online streaming for various social activities, such as live e-commerce, sports events, or festival ceremony. The major online video service platforms, e.g., YouTube, Twitch.tv, and TikTok, provide the video player with a built-in interface for live comments, which are synchronized with the video playback. Therefore, the viewers watching the same video can share the opinions and feelings at this moment, with improved interactive and immersive experiences. For example, according to the report of SullyGnome [1], which is a website tracker for Twitch, the average number of channels can reach hundreds of thousands with more than 4 million viewers every day. Meanwhile, according to the statistics of the well-known Chinese live video website Douyu, the number of time-sync comment in it has exceeded 10 billions in 2020. In addition, the traffic flows of time-sync comments can be highly dynamic. Especially, during the live streaming for some world-wide social events, such as the league of legends S10 World Finals on October 31, the peak traffic of time-sync comments can surge 300% according to SullyGnome [1].

Conventionally, the streaming channel fingerprint can be revealed by video traffic pattern analysis in despite of transportation layer encryption (e.g. TLS). There have been several attempts to identify encrypted streaming channels through the analysis of correlation between video content and bitrate variations [2] [3] [4]. Most prior works assume that the encrypted video stream can be directly observed by the adversary without interference of irrelevant traffic flows. Actually, nowadays the live video streaming channels are usually...
delivered from CDNs which are shared by other encrypted traffic flows with the unified domain name. Therefore, the encrypted video channel can be hardly filtered from the noisy environment during real world implementation. Furthermore, the effectiveness of bitrate based identification solutions is highly dependent on the network conditions. In the cases of network fluctuations or degradations, the characteristics of bitrate fingerprint diminish apparently.

As a distinctive and long-lasting characteristic, the prevalence of time-sync comments brings a new risk to the streaming channel identification. Instead of the CDN based content delivery, the encrypted time-sync comments traffic flow is usually transmitted from the dedicated server owned by the content providers directly. In addition, the fingerprint of encrypted time-sync comment is resilient to the network condition variations, as the comment traffic is relatively small (e.g., the length of a single time-sync comment in a packet from YouTube is usually less than 1KB), and the transmission is usually delay tolerant. In this paper, we will present a channel identification method for encrypted live streaming with time-sync comments. We will show that the delicate traffic flow of time-sync comments can be a fine-gained fingerprint after appropriate preprocess. Initially, a convolutional neural network is trained for feature extraction and filter the relevant packets of time-sync comments from the dedicated server. After that, the least square is utilized to compute the comment rate online. Further, a dynamic time warping (DTW) based feature extraction method is improved with traffic trend analysis to accommodate the transmission delay. Finally, support vector machine (SVM) is used in similarity measure to identify the target channel.

The rest of this paper is organized as follows: The literature is explored in Section II. The data analysis is presented in Section III. The system design and traffic filter are illustrated in Section IV and Section V. The comment rate is estimated in Section VI, and delay tolerant similarity matching strategy is proposed in Section VII. The system performance is evaluated in Section VIII. Finally, Section IX concludes this paper.

II. RELATED WORK

A. PRIVACY LEAKAGE AND PROTECTION

Along with the growth of Internet users, new security issues arise while traditional security issues become more severe [5]. On one hand, the new paradigms could bring facilities and impressive experiences, such as data forecasting [6], content sharing [7] [8] and computation offloading [9]. On the other hand, the challenge of defense strategy is elevated, and sophisticated protection mechanism should be involved in every aspect from end devices to infrastructures, e.g., mobile devices [10], edge server [11], cloud data center [12] [13], and blockchain [14]. Specifically, the development of machine learning improves the traditional data analysis methods with smart attack, which greatly increases risk of privacy leakage through feature extraction from various user behaviors [15] [16] [17]. In this paper, we are motivated to explore the privacy detection methods of video viewers from following three aspects.

B. PRIVACY LEAKAGE FROM VIDEO BITRATE STREAM

The privacy leakage of encrypted video caused by side channel attack has attracted wide attention in recent years. Saponas et al. utilized a window based DFT to make fingerprints for video stream identification in VBR encoding which divides the video into 100ms segments [2]. Aceto proposed a novel heuristic to reconstruct application-layer messages from encrypted traffic [18]. Gu et al. further utilized DTW to match video fingerprints and traffic patterns, with a classifier to identify encrypted video streams from multiple websites [19]. As the prevalence of deep learning, neural network has an advantage of feature extraction in sophisticated environment. Schuster et al. proposed a CNN-based model to identify the traffic pattern fingerprints for DASH videos from main video stream website [3]. Some works even proposed an end-to-end learning approach with the integration of both feature extraction and classification in a neural network [20] [21]. There are some works considering the traffic filter issue in practical implementation in Table 1. Nevertheless, all the bitrate-based video identification strategies need the assumption of stable network. Otherwise, once the bandwidth varies or delay occurs, the delicate bitrate fingerprint can be easily erased. In contrast, the traffic fingerprint based on time-sync comments is highly robust to network conditions.

C. TRAFFIC NOISE FILTER

Most of existing works assume that the traffic data is clean or after pre-processed. Actually, the encrypted video streams are usually mixed with irrelevant traffic flows as noises, which seriously erase the traffic pattern characteristics of target bit rate. There are several previous works considering the traffic noise issues. For example, Zhang et al. proposed a flow-level method to identify the zero-day traffic from the tagged traffic using k-means and random forest [22]. Hu et al. proposed an improved SVM method, which can eliminate the influence of weak correlation and outlier samples [23]. Comparatively, the encrypted time-sync comments traffic flow is easier to discriminate from irrelevant traffic of other applications, as the time-sync comments traffic flow is usually transmitted from the dedicated server owned by the content providers directly. Nevertheless, as to inner application noises, encrypted traffic filter still needs to be carefully designed to eliminate irrelevant traffic flows brought by user behavior or advertisement.

D. SIMILARITY MEASURE

Distance or similarity measures are essential in solving many pattern recognition problems such as classification and clustering, which are prevalent in signal analysis, speech recognition and other applications [24]. Traditional methods mainly include point based solutions (e.g., Edit Distance on Real Sequence (EDR) [25], Dynamic Time Warping (DTW))
TABLE 1: Overview of related works

| Related works     | Summary                                                                                                                                                                                                 | Neural network | Similarity matching | Traffic filter | Stable network |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|---------------------|---------------|---------------|
| Saponas et al. [2]| a video stream identifier through bitrate variations caused by VBR encoding.                                                                                                                             | ✓               | ✓                   | ✓             | ✓             |
| Schuster et al. [3]| a CNN-based model to identify the traffic pattern fingerprints                                                                                                                                          | ✓               | ✓                   | ✓             | ✓             |
| Gu et al. [19]    | a method using DTW to match video fingerprints and traffic patterns                                                                                                                                      | ✓               | ✓                   | ✓             | ✓             |
| Zhang et al. [22] | a method to identify the zero-day traffic from the tagged traffic using k-means and random forest                                                                                                      | ✓               | ✓                   |               |               |
| Hu et al. [23]    | an improved SVM method, which can eliminate the influence of weak correlation and outlier samples                                                                                                        | ✓               | ✓                   |               |               |
| Wang et al. [31]  | to reduce the negative effect of mislabeled samples                                                                                                                                                       | ✓               | ✓                   |               |               |
| Aceto et al. [18] | a novel heuristic to reconstruct application-layer messages in the common case of encrypted traffic                                                                                                       | ✓               | ✓                   | ✓             | ✓             |
| Lotfollahi et al. [20]| a deep learning based approach which integrates both feature extraction and classification phases into one system                                                                                       | ✓               | ✓                   |               |               |
| Aceto et al. [21] | a mobile traffic classifier based on automatically-extracted features using deep learning                                                                                                              | ✓               | ✓                   |               |               |

FIGURE 1: Rate variations of time-sync comments in live channels

FIGURE 2: Bitrate with different bandwidth limitations

III. DATA ANALYSIS AND MOTIVATION

In this section, we will present the real world data analysis to illustrate the traffic flow features of video bitrate and time-sync comments, as well as the influence of traffic noises.

A. TRAFFIC FLOWS OF TIME-SYNC COMMENTS AS FINGERPRINTS

Video traffic analysis is usually performed with variable bitrate encoding (VBR), in which the bitrate can be adaptive to the video content and network condition fluctuations. Generally, the transportation layer encryption (e.g. TLS) can be utilized to hide the content, but not the traffic features. Therefore, the privacy of viewers could be disclosed through traffic pattern analysis. Similarly, the traffic features of time-sync comments could also act as fingerprints of a live...
channel. Figure 1 shows the rates of time-sync comments from two live channels of YouTube for 100 seconds, as a video game streaming channel captured in 19:43:02 on Sep 1st 2020, and an animation streaming channel captured in 10:22:05 on Sep 2nd, 2020. It can be seen that the number of time-sync comment ranges from 1 to 40 in each time period. The time varying comment rates and the distinct length of each comment together leads to the comment traffic variation, which can act as a unique fingerprint even after data encryption.

B. TRAFFIC FEATURES WITH BANDWIDTH LIMITATIONS

Here we will present the traffic features under bandwidth limitations, for video bitrate and comment rate, respectively. The video segments and the related time-sync comments are all collected from YouTube between 10 a.m. to 5 p.m. on Sep 1st 2020. The resolution is 240p with VBR encoder, and Network Emulator for Windows Toolkit is utilized to emulate the bandwidth limitation of the victim. Figure 2 presents the traffic pattern with 300kb bandwidth limitation. As the maximum bitrate of the video (about 4500kbps) is much higher than the bandwidth limitation, and some features will disappear, leading to seriously deteriorated matching accuracy. In a word, the bitrate-based video fingerprints raise stringent requirements on network conditions. In contrast, the requirement for comment-based fingerprints are much more lenient, as the traffic flow of time sync comment is relatively tiny. Figure 3 shows the impact of bandwidth limitations on time-sync comment features. Specifically, the five regions divided by dashed lines from left to right represent the limited bandwidth from 0 to 200kbps, the distance (red line) shows the difference of comment number between bandwidth with limitation or not. It can be seen that the traffic features of time sync comments will not be influenced until the bandwidth is reduced to 150kbps.

C. TRAFFIC FEATURES UNDER INTERFERENCE

In this part, we will illustrate the influence of interference to the traffic features. Specifically, here we consider four types of interferences, which are widespread in modern video streaming platforms.

1) Inter-application traffic interferences
The eavesdropping traffic usually contains packets from multiple applications, which makes it difficult to model the traffic pattern. Figure 4 shows a traffic segment from YouTube game channel, with traffic pattern as the filtered time sync comment traffic, and unfiltered traffic as the original traffic (traffic pattern mixed with video stream). It can be seen that the traffic pattern of comments is covered with mixed traffic flows, and can be hardly distinguished without traffic filtering.

2) Intra-application traffic interferences
Even in the same application, the data traffic can be significantly affected by user operations, which are usually unpredictable. In Figure 5, we show the traffic burst caused by users browsing video list. When a user browses a video recommendation list, the traffic volume caused by the video information will surge, as appeared in 120s to 200s of Figure 5.

3) Heterogeneous packets in TLS sessions
The time-sync comment packets usually come along with various noise packets, (such as the information of the comment author, video playback feedback data and so on) in the same TLS session. Figure 6 shows the distribution of comment packet size and noise packet size in a 1200s-TLS session from YouTube. As shown in the figure, the noise packets account for more than 50% of all packets in the TLS sessions, which improve the difficulty to predict the comment rate.

1 Some video sites choose to push the comment content one by one, while others (like YouTube) choose to push multiple comment content at regular intervals.
4) Heterogeneous arrival times

Some platforms choose to send the live comments one by one, and others choose to send several comments in groups periodically. The intermittent traffic pattern is different from video streaming, which is usually transmitted through buffered chunks in a continuous manner. In addition, the comments sent to viewers are not synchronized and the RTT is distinct for each viewer. Thus, received comment rates may be not exactly the same, even for the viewers of the same channel. As traffic pattern and fingerprint are monitored by different hosts respectively, there may be feature dislocation between monitored traffic pattern and observed fingerprint (as shown in Fig. 7, there is a 5s delay between the fingerprint and traffic pattern).

In summary, the bitrate-based traffic features are generally extracted with assumptions of stable network connections without any other irrelevant traffic flows. In contrast, the comment-based traffic features are resilient to network variation and interference in nature. Yet further preprocess still needs to be performed to extract the comment traffic as a fine-gained fingerprint.

IV. SYSTEM DESIGN

In this section, we will present the system design with time-sync comments traffic as a fingerprint for live channel identification. The system structure is presented in Figure 8. The proposed system can be divided into three parts:

- Traffic filter: In order to filter out the irrelevant traffic flows, TLS sessions is identified with the target domain for inter-application traffic filter, and a CNN-based model is proposed for intra-application traffic.
- Comment rate estimation: The time-sync comment rate is predicted with filtered traffic through the least square method.
- Similarity measure: A DTW based solution is proposed for feature extraction between estimated comment rate and local monitored traffic pattern for all channels, and SVM is utilized for channel identification finally.
Since time-sync comments are sent in groups, each group of comments will be encapsulated in a separate session for transmission. Thus, session is denoted as a unit for captured packet sets. ContentType is used to check whether the current packet is a TLS handshake packet, or whether the viewer has created a new session. As the source domain name of the handshake packet is disclosed by SNI, all irrelevant TLS sessions can be filtered out. The filtering process is shown in Algorithm 1.

**Algorithm 1 Packet Filter.**

**Input:**
- Packet sequence \( P \);

**Output:**
- Session sequence \( S \):

1. **while** packet\([i+1]\) != NULL do
2. if ContentType==HandShake and SNI == targetdomain then
3. Create a new sequence \( s \)
4. Add packet\([i].length and packet\([i].time to sequence\)
5. Old_IP = packet\([i].IP \)
6. else if ContentType == ApplicationData and packet\([i].IP == Old_IP then
7. Add packet\([i].length to sequence\)
8. **end if**
9. **end while**

After filtering, we get a set \( S = \{ s_0, s_1, ..., s_j \} \) containing j session parameters, where \( s_j \) is a two-tuple \( <P_{session}^{j}, t_j> \) for the \( j \)th session with \( t_j \) as the arrival time of the ClientHello packet and \( P_{session}^{j} \) as a sequence including the length of each packet in this session.

**B. INTRA-APPLICATION TRAFFIC FILTER**

In this part, we will describe the elimination of irrelevant traffic flows from the same application, which is mainly caused by video feedback data or user operations, such as loading playback list, or viewing home pages.

Initially a set of packets containing time-sync comments are divided into three parts as shown in Figure 9: irrelevant part, head part, mid part and trail part. Packets related to comments are always continuous in a session, and each part has a constant size range and distribution. The irrelevant

The notations used in the paper are shown in Table 2.

**Table 2: Parameter notations**

| Parameter     | Description                          |
|---------------|--------------------------------------|
| \( F^t \)     | time-sync comments timestamp sequence|
| \( C^t \)     | the comment timestamp                |
| \( P^t \)     | the captured packet sequence          |
| \( p_i \)     | \( i \)th captured packet in \( P^t \)|
| \( S_j \)     | the session sequence output from Algorithm I |
| \( P_{session} \) | packet length sequence of a session  |
| \( p^s \)     | the packet length in \( P_{session} \)|
| \( t_j \)     | arrival time of the ClientHello packet|
| \( J_j \)     | the relevant packet sequence contains in \( P_{session} \)|
| \( P^{a+k-1} \) | the \( a + k - 1 \)th packet from \( P_{session} \)|
| \( F^F \)     | the comment rate set                  |
| \( F^F \)     | the fingerprint set                   |
| \( C^F \)     | \( i \)th comment rate in \( F^F \)   |
| \( C^F \)     | \( i \)th fingerprint in \( F^F \)    |
| \( P^P \)     | the feature set calculated by \( F^F \) and \( F^P \) |
| \( F^P, F^{EST}, F^{STD} \) | the features calculated by \( F^F \) and \( F^P \) |
| \( S^{P} \)   | similarity point                      |
| \( D \)       | the number of kernels                 |
| \( M \)       | the size of feature map               |
| \( N_{in}/N_{out} \) | input/ output channel size          |
| \( I_{in}/I_{out} \) | input/ output dimensionality of fully connected layer |

**V. TRAFFIC FILTER**

In this section, we will illustrate the traffic filter to eliminate the irrelevant traffic flows caused by other applications or unpredictable user operations in the same application.

**A. INTER-APPLICATION TRAFFIC FILTER**

In order to perform preliminary traffic filter, three labels are utilized, as Server Name Indicator(SNI), ContentType, and source IP address, which still can be disclosed even after data encryption with TLS protocol.

- SNI is used to filter out irrelevant traffic from other websites
- ContentType is used to identify the TLS session
- IP address is used to ensure the continuity of TLS session

**FIGURE 8: An overview of system structure**

**FIGURE 9: The distribution of four parts in a session related to time-sync comments**
part represents the packet without any time-sync comment information. This part is mainly composed of the video feedback data and the data from browsing web pages. The head part is the first packet containing the time-sync comments, which is usually short. The middle part is a series of long packets, which are used to transmit the content of time-sync comments. The trail part usually includes 0-3 short packets to denote the end of the middle part. Secondly, a CNN based model is presented for session classification in Figure 10. The model input is a 32 dimensional vector, as most of time-sync comment sessions have no more than 25 packets. Thus, for the sessions less than 32 packets, 0 is filled up at the end of input vector. The CNN model works as a classifier to calculate the probability whether the session contains the time-sync comments. Here, we consider 1D-CNNs as an ideal choice for the network traffic classification task, which is similar to encrypted network traffic classification in [21]. Fortunately, in the intra traffic filter process, we do not have to classify the network traffic of various applications, and focus on whether session contains time-sync comments. Therefore, here we consider CNN architecture with one layer and two kernels, and we will prove that this simplified network architecture can achieve high accuracy efficiently during online eavesdropping of the victim.

Then, we use sliding windows of different size to obtain the input vector composed of packet length from the TLS session (for example, when the TLS session length is 16, the sliding window length is set from 3 to 16, and 105 vectors are generated accordingly). The input vector is first sent into the convolutional layer to extract the correlation features. Therefore, we use two convolution kernels with size of 1 * 2 and size of 1 * 3 to extract the features of the sequence. RELU is used as the activation function. After that, the maximum pooling layer is used to compress the features extracted from the two convolution layers into same length and add 0.1 drop rate. Next, the results of pooling layers are combined, and then sent to fully connected layer. The fully connected layer consists of a hidden layer with 16 neurons and an output layer with two neurons. After softmax activation function, the output represents the probability of whether the vector contains all time-sync comment packets.

The input vector is defined as $l_j = \{p_{a}^j, \ldots, p_{a+\alpha}^j\}$ (i.e., the length sequence of the packets containing the comment content in $l_j$) identified by CNN model, $a$ is the index of the first packet containing comment data in the sequence, and $\alpha$ is the number of comment packets. $l_j \in P_{session}$. We calculate the computational complexity of the model as follows:

- **Convolution layer:** $Time \sim O(\sum_{\alpha=1}^{D} M_\alpha * K_\alpha * N_{in} * N_{out})$
- **Max polling layer:** $Time \sim O(M_\alpha * N_{in})$
- **Fully connected layer:** $Time \sim O((2 * I_{in} - 1) * I_{out})$

Where $D = 2$ is the number of kernels, $M_1 = 31, M_2 = 30$ is the size of feature map, $K_1 = 2, K_2 = 3$ is the kernel size, $I_{in} = 16$ and $I_{out} = 2$ is the input or output dimensionality of the fully connected layer, $N_{in} = 1$ and $N_{out} = 1$ is the input or output channel size of the convolutional layer.

**VI. COMMENT RATE ESTIMATION**

Here, we analyze the relationship among the number of time-sync comments, comment length, and packet length. In each time-sync comment carries a large amount of structural formation including various auxiliary information, such as user ID, comment checksum, etc, which takes as a major part. Therefore, the length of TLS session is generally correlated to the number of comments in this session.

Figure 11 shows the relationship between the number of comments and the length of comment packets from YouTube game streaming channel. Generally, the number of the comments is proportional to the packet length. We define a linear...
relationship between the number of comments $C^c$ and the comment packet length $p^t$ as follows:

$$C^c = k \cdot \left( \sum_{m=a}^{k-1} p^t_m \right) + b$$

Where $k$ and $b$ can be computed through least square. Then, we can estimate the comment rate $F^c = \{C^c_0, C^c_1, \ldots, C^c_g\}$ for the monitored target victim. After that, we use the time stamp $t^i$ of the elements in comment rate $F^c$ to match each element in the fingerprint. If time period between $C^t$ and $t^i$ is less than 2000ms, the elements in $F^t$ that before $C^t$ are deleted, and the number of them is added to $F^p$ as a new element, as shown in algorithm 2.

Algorithm 2: Calculate the final time-sync comment fingerprint $F^p$.

**Input:**
- $F^t$ and comment rate $F^c$;

**Output:**
- fingerprint $F^p$;

1. for $i = 0$ to len($F^c$) do
2.  $x = 0$
3.  while len($F^t$) != 0 and $x < $ len($F^t$) do
4.     if abs($F^t[i] - F^c[i].t^i$) < 2000 then
5.         $F^p.append(x + 1)$
6.     end if
7.     $x = 0$
8.  end if
9.  $x += 1$
10. end while
11. end for

Furthermore, we have the comment fingerprint $F^p = \{C^p_0, C^p_1, \ldots, C^p_g\}$ as the observed fingerprint from attacker.

### VII. DELAY TOLERANT SIMILARITY MATCHING

In the previous section, we obtained the fingerprint $F^p$ from the local monitoring of attacker, and comment rate $F^c$ with the same length from the eavesdropping of the victim. Here, we will propose a delay tolerant similarity matching method for channel identification, as the arrival of time-sync comments is not strictly synchronized.

Before we implement the similarity matching between hundreds of local monitored channels and the target channel of the victim, we will have a discussion about the dislocation issue. As the viewers have different RTTs and the time-sync comments loaded in packets are not synchronized, the comment rates are different for each viewer, even if they are actually watching the same channel. For example, for 10 comments in the same channel, the comment rate observed by a viewer with higher RTT is (2, 4, 4), while that observed by another viewer with lower RTT is (3, 5, 2). It refers to the dislocation issue which violates the uniqueness of the fingerprint in a fine granularity observation, even though the trend keeps consistent in the long-term observation. To deal with this issue, we will present a sliding-window design to accommodate the local mismatch and focus on the similarity matching of the long-term trend. The formula is presented as follows:

$$C^c_{i:p} = \frac{\sum_{j=i+1}^{i+p} C^c_j}{p}, 0 \leq i, p \leq g$$

A sliding window is setup to sum the adjacent elements of the fingerprint, and elements in the new sequence will be associated with adjacent elements. The sliding window aims to extract the general trend variation of comment traffic. The comparison of original fingerprint and the processed fingerprint is shown in Figure 12. It can be seen that after preprocess of the sliding window, the similarity of the long-term trend appears more evident.

After that, a feature extraction method is proposed based on DTW. The feature set $F = \{F^D, F^{SP}, F^{SD}\}$ which contains three features are calculated and SVM classifier is used to classify the features. In other words, we calculate the feature set by pairs of comment fingerprint and comment rate, and then send the features to classifier for a two-category classification. The output result is the matching degree of the fingerprint and comment rate.

Generally, DTW is mainly used to measure whether two time series are the same, especially for applications in speech recognition and data mining. In this section, we will design a new feature extraction method based on DTW to accommodate noises. Define a similarity point ($S^p_i$) if any of the i-th point satisfies the following formula:

$$Sgn[ \frac{C^c_{i+\gamma k} - C^c_{i+(\gamma-1)k}}{k} ] = Sgn[ \frac{C^p_{i+\gamma k} - C^p_{i+(\gamma-1)k}}{k} ]$$

$k = 1, 2$

$\gamma = 0, 1$

The signum function of a real number $x$ is defined as follows:

$$y = \begin{cases} 
-1 & \text{if } x < 0, \\
0 & \text{if } x = 0, \\
1 & \text{if } x > 0.
\end{cases}$$

We use the first derivative of comment fingerprint and comment rate to indicate their increase and decrease, where Sgn represents the sign taking function. After that, we use the average similarity point ($F^{SP}$) to measure the fingerprints of different lengths.

$$F^{SP} = S^p_i / g$$

Where $g$ is the length of comment fingerprint. In addition, we propose the Delay-Tolerant DTW (DDTW) described in Algorithm 3. In the stage of constructing the distance matrix, the distance between each pair of similar points is set to 0. The "average similarity distance" is calculated according to the number of similar points, as follows:

$$F^{SD} = (DTW - DDTW) / S^p$$
Finally, we can get the feature $F_D$ (result of DTW), $F_{SP}$ and $F_{SD}$, and merge them into feature set $F = \{F_D, F_{SP}, F_{SD}\}$. Specifically, the feature $F_D$ reflects the macro similarity, in which lower value means closer relationship. The feature $F_{SP}$ reflects the special similarity, which refers to evident change such as a sharp increase or decrease, with higher value for closer relationship. The feature $F_P$ reflects that the macro similarity with lower value for closer relationship. SVM is utilized to classify the feature set, and further find the set with the highest similarity degree according to the distance matrix.

VIII. IMPLEMENTATION AND EVALUATION

A. EXPERIMENTAL SETUP

In order to build the prototype system, we have two Amazon EC2 servers as a victim and an attacker, respectively. The server configuration is listed in Table 3. Wireshark and Fiddler is used to simulate Man-In-The Middle (MITM) attack to capture the encrypted TLS traffic of the victim. The attacker server is used to eavesdrop the live channels with time-sync comments that arrive continuously. The arrival timestamp of live comments are recorded from 3 different live channels of YouTube and BiliBili for 6 hours. Table 4 shows a summary of our dataset.

2 The data set can be found at https://1drv.ms/u/s!AnB84OgJQM04iyko73uwfV6VinKJ?e=5Y67EE

Algorithm 3 Delay-Tolerant DTW (DDTW).

Input:
- $F_P$: fingerprint
- $F_C$: comment rate

Output:
- $DDTW$: the result of DDTW

1. $m = \text{len}(F_C), n = \text{len}(F_P)$
2. $M = \text{new Int}[m][n]$
3. for $i = 0$ to $\text{len}(F_C)$
   4. for $j = 0$ to $\text{len}(F_P)$
      5. if $i == 0$
         6. $M[i][0] = MAX_{INT}$
      7. end if
      8. if $j == 0$
         9. $M[0][j] = MAX_{INT}$
      10. end if
      11. if isSimilarityPoint($i, j$)
         12. $M[i][j] = 0$
      13. else
      14. $cost = \text{dis}(F_P[i], F_C[j])$
      15. $M[i][j] = cost + \text{min}(M[i - 1, j], M[i - 1, j - 1], M[i, j - 2])$
      16. end if
      17. end for
6. end for
7. return $M[m,n]$

FIGURE 12: The comparison of original and processed fingerprint

FIGURE 13: The distribution of comment packets length in each part

VOLUME 4, 2016
B. TRAFFIC FILTER ACCURACY

In order to evaluate our proposed traffic filter method, 200 decrypted sessions are collected from YouTube with length distribution of four kinds of packets in Figure 13. It can be seen that there are significant differences in the distribution of the four kinds of packet, and the accuracy of our proposed filter model can reach 93.2%. In addition, we analyze the relationship between the sliding window size and prediction accuracy in Figure 14. It can be seen from the figure that the accuracy is the highest when the window size is 3. As the window size grows, the accuracy decreases slowly. Therefore, in the following experiments, we set the window size to 3.

C. QUALITY OF COMMENT-BASED FINGERPRINT

Generally, the popular live channels with obvious comment rate variation are easier to be identified from other live channels. Here, we will evaluate the system performance under the impact of comment rate variation in different scenarios. In Scenario A, unmatched fingerprints are all from different live channels. In Scenario B, unmatched fingerprints are all from different time periods in the same live channel. In Figure 15, when the eavesdropping time is less than 100 seconds, the accuracy of case A is low. As eavesdropping time increases, the identification accuracy is gradually improved, and can finally reach that in Scenario B. This is because when the eavesdropping time is short, the valid traffic pattern accounts only for small portion. In addition, the stable comment rate without evident change also lack unique features, and improves the difficulty of similarity matching. Secondly, we consider the influence of the distance from the features to the hyperplane on the accuracy of the identification results. In order to qualify the similarity matching precision, if a comment rate is matched to inconsistent fingerprints by mistakes, we record it as false positive (FP), and vice versa as true positive (TP). The formula of precision is as follows:

\[ \text{Precision} = \frac{TP}{TP + FP} \]

For the same set of samples, the result is shown in Figure 16. Obviously, the precision of classification has been

![FIGURE 14: The relationship between window size and accuracy](image1)

![FIGURE 15: The identification accuracy with different live streaming channels](image2)

![FIGURE 16: The impact of the distance to hyperplane on identification accuracy](image3)
TABLE 5: Threshold of DTW, P-DTW and Pearson

| Time (s) | DTW  | P-DTW | Pearson |
|---------|------|-------|---------|
| 50      | 0.18 | 0.22  | -0.25   |
| 100     | 0.49 | 0.49  | -0.25   |
| 150     | 0.36 | 0.25  | -0.25   |
| 200     | 0.23 | 0.18  | -0.25   |
| 250     | 0.18 | 0.13  | -0.25   |
| 300     | 0.14 | 0.04  | -0.25   |
| 350     | 0.13 | 0.04  | -0.25   |
| 400     | 0.11 | 0.03  | -0.25   |
| 450     | 0.05 | 0.04  | -0.25   |
| 500     | 0.05 | 0.03  | -0.25   |

improved with this method, especially when the fingerprint length is short.

D. PERFORMANCE EVALUATION AND COMPARISON

In this part, our proposed method will be compared with Pearson correlation coefficient, DTW, and improved DTW algorithm P-DTW. As shown in Figure 17, we use 100 sets of traffic data in groups of 200 seconds to calculate their fingerprints, and use DTW and P-DTW to calculate the similarity distances. We use the intersection of two false rate lines as the threshold to maximize the accuracy. It can be seen from Figure 18, 0.23 is the identification threshold to maximize the accuracy of DTW for 200 seconds. In the following comparison, we calculate 10 thresholds of 50-500 seconds to define the “accuracy” of the three methods, as shown in Table 5 (for the result of DTW and P-DTW algorithms, more than this threshold is considered a mismatch, and for Pearson correlation coefficient, less than this threshold is considered a mismatch). Figure 19 shows the accuracy comparison between our method and the above three methods when fingerprints of different lengths are used, and our accuracy can reach 98.2% when the fingerprint length is 500 seconds.

Then we compare our method with the other four state of art VBR bitrate-based methods: leaky [32], slingbox [2], beauty [3] and pdtw [19]. The test video channels are captured from 5 movies from Youtube as Titanic, Black Swan, Trainspotting, Inception and Forrest Gump. Each channel consists of 10 segments of 120 live comments sessions and 200 video clips of 200 seconds. The file downloader is implemented as interference of irrelevant traffic. As shown in Table 6, compared to bitrate-based methods, our method can achieve the highest accuracy in all test movies under interference and also does well without irrelevant traffic interference.

E. IMPACT OF NETWORK VARIATION

In this section, we will evaluate the accuracy of identification in network variation environment. The network emulator for windows toolkit is used for network emulation, which can generate various network conditions, such as packet loss, packet error, delay, bandwidth limit, number of closed connections, etc. In order to verify the impact of bandwidth, we set the upper bound bandwidth...
TABLE 6: Comparison of different methods

| Interference from file downloading | No interference |
|-----------------------------------|-----------------|
|                                   | Titanic | Black Swan | Trainspotting | Inception | Forrest Gump | Titanic | Black Swan | Trainspotting | Inception | Forrest Gump |
| leaky                             | 4.1%    | 4.4%       | 4.2%          | 5.7%      | 3.1%        | 78.8%   | 79.7%       | 78.0%        | 73.9%     | 77.4%        |
| slingbox                          | 11.3%   | 6.5%       | 11.3%         | 18.4%     | 9.2%        | 93.4%   | 60.2%       | 93.5%        | 85.3%     | 61.8%        |
| beauty                            | 15.5%   | 5.2%       | 6.3%          | 11.3%     | 6.3%        | 94.7%   | 91.1%       | 98.1%        | 93.5%     | 93.2%        |
| pdtw                              | 7.9%    | 10.7%      | 13.7%         | 8.1%      | 15.6%       | 90.2%   | 90.0%       | 89.3%        | 90.2%     | 88.9%        |
| our method                        | 92.1%   | 91.8%      | 91.2%         | 92.1%     | 91.0%       | 92.0%   | 91.3%       | 91.6%        | 90.7%     | 91.1%        |

FIGURE 20: Accuracy under bandwidth limitation

FIGURE 21: Accuracy under interferences

In order to analyze the impact of irrelevant traffic flows, we run other applications while playing YouTube live video to emulate the real network usage of users. As we can see from Figure 21, the identification accuracy of our proposed strategy does not deteriorate when the viewer watches YouTube and Bilibili live streaming at the same time, or performs file download while watching YouTube live streaming.

**IX. CONCLUSIONS & FUTURE WORK**

In this paper, we proposed a channel identification method for encrypted traffic flow, which uses the traffic features of time-sync comments. A real dataset was captured from three YouTube live channel and a prototype system was presented for performance evaluation. Through extensive experiments, we proved that even in the complex network environment, our proposed solution can reach 98.2% accuracy after 500 seconds traffic eavesdropping.

With the prevalence of time-sync comments in nowadays video streaming system, our work provides the full set of process channel identification from traffic filtering to similarity matching. We have shown that the time-sync comments accompanied in the video channels could be a fine-gained fingerprint after preprocess and resilient to network condition variations. In our future work, the system performance with respect to balancing accuracy and efficiency can be further considered from the following two aspects. First, an end-to-end training model can be considered to merge the feature engineering and channel classification. Second, an online adaptive design can help to select the candidates and reduce the similarity matching complexity as time goes.

**REFERENCES**

[1] https://sullygnome.com/.
[2] T. S. Saponas, J. Lester, C. Hartung, S. Agarwal, and T. Kohno, “Devices that tell on you: Privacy trends in consumer ubiquitous computing,” in Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium, SS’07, (USA), USENIX Association, 2007.
[3] R. Schuster, V. Shmatikov, and E. Tromer, “Beauty and the burst: Remote identification of encrypted video streams,” in 26th USENIX Security Symposium (USENIX Security 17): 1357–1374, 2017.
[4] H. Wu, Z. Yu, G. Cheng, and S. Guo, “Identification of encrypted video streaming based on differential fingerprints,” in IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2020.
[5] Z. K. Zhang, M. C. Y. Cho, C. W. Wang, C. W. Hsu, C. K. Chen, and S. Shieh, “IoT security: ongoing challenges and research opportunities,” in 2014 IEEE 7th international conference on service-oriented computing and applications, 2014.
[6] Z. Tang, G. Zhao, and T. Ouyang, “Two-phase deep learning model for short-term wind direction forecasting,” Renewable Energy, 2021.
[7] S. Zhang, H. Xian, Z. Li, and L. Wang, “Secdedup: Secure encrypted data deduplication with dynamic ownership updating,” IEEE Access, 8: 186323–186334, 2020.
[8] X. Lu and X. Cheng, “A secure and lightweight data sharing scheme for internet of medical things,” IEEE Access, 8: 5022–5030, 2019.
[9] X. Ge, J. Yu, H. Zhang, J. Bai, J. Fan, and N. N. Xiong, “Spps: A search pattern privacy system for approximate shortest distance query of encrypted graphs in IoT,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 1–15, 2021.

[10] X. Lu, Z. Pan, and H. Xian, “An integrity verification scheme of cloud storage for internet-of-things mobile terminal devices,” Computers & Security, 92: 101686, 2020.

[11] H. Zhang, J. Yu, M. S. Obaidat, P. Vijayakumar, L. Ge, J. Lin, J. Pan, and R. Hao, “Secure edge-aided computations for social internet-of-things systems,” IEEE Transactions on Computational Social Systems, 2020.

[12] H. Zhang, J. Yu, C. Tian, P. Zhao, G. Xu, and J. Lin, “Cloud storage for electronic health records based on secret sharing with verifiable reconstruction outsourcing,” IEEE Access, 6: 40713–40722, 2018.

[13] J. Yu and R. Hao, “Comments on sepdp: Secure and efficient privacy preserving provable data possession in cloud storage,” IEEE Transactions on Services Computing, 2019.

[14] H. Zhang, L. Tong, J. Yu, and J. Lin, “Blockchain aided privacy-preserving outsourcing algorithms of bilinear pairings for internet of things devices,” arXiv preprint arXiv:2101.02341, 2021.

[15] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, “IoT security techniques based on machine learning: How do IoT devices use ai to enhance security?,” IEEE Signal Processing Magazine, 35(5): 41–49, 2018.

[16] Y. Zhang, H. Xian, and A. Yu, “Csm: Password guessing method based on chinese syllables and neural network,” Peer-to-Peer Networking and Applications, 13(6): 2237–2250, 2020.

[17] Y. Liu, S. Zhang, J. Zhang, L. Tang, and Y. Bai, “Assessment and comparison of six machine learning models in estimating evapotranspiration over croplands using remote sensing and meteorological factors,” Remote Sensing, 13(19): 3838, 2021.

[18] G. Aceto, G. Bovenzi, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapé, “Characterization and prediction of mobile-app traffic using markov modeling,” IEEE Transactions on Network and Service Management, 18(1): 907–925, 2021.

[19] J. Gu, J. Wang, Z. Yu, and K. Shen, “Traffic-based side-channel attack in video streaming,” IEEE/ACM Transactions on Networking, 27(3): 972–985, 2019.

[20] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian, “Deep packet: A novel approach for encrypted traffic classification using deep learning,” Soft Computing, 24(3): 1999–2012, 2020.

[21] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, “Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges,” IEEE Transactions on Network and Service Management, 16(2): 445–458, 2019.

[22] J. Zhang, X. Chen, Y. Xiang, W. Zhou, and J. Wu, “Robust network traffic classification,” IEEE/ACM transactions on networking, 23(4): 1257–1270, 2014.

[23] S. Hao, J. Hu, S. Liu, T. Song, J. Guo, and S. Liu, “Improved svm method for internet traffic classification based on feature weight learning,” in 2015 International Conference on Control, Automation and Information Sciences (ICCAIS), 2015.

[24] T. W. Liao, “Clustering of time series data - a survey,” Pattern recognition, 38(11): 1857–1874, 2005.

[25] L. Chen and R. Ng, “On the marriage of lp-norms and edit distance,” in Proceedings of the Thirtieth international conference on Very large data bases-Volume 30: 792–803, 2004.

[26] M. Müller, “Dynamic time warping,” Information retrieval for music and motion: 69–84, 2007.

[27] T. Eiter and H. Mannila, “Computing discrete fréchet distance,” tech. rep., Citeseer, 1994.

[28] B. Lin and J. Su, “One way distance: For shape based similarity search of moving object trajectories,” GeoInformatica, 12(2): 117–142, 2008.

[29] X. Ding, K. Hao, X. Cai, X. S. Tang, L. Chen, and H. Zhang, “A novel similarity measurement and clustering framework for time series based on convolution neural networks,” IEEE Access, 8: 173158–173168, 2020.

[30] F. Karim, S. Majumdar, H. Darabi, and S. Chen, “Lstm fully convolutional networks for time series classification,” IEEE access, 6: 1662–1669, 2017.

[31] B. Wang, J. Zhang, Z. Zhang, L. Pan, Y. Xiang, and D. Xia, “Noise-resistant statistical traffic classification,” IEEE Transactions on Big Data, 2017.

[32] A. Reed and B. Klimkowski, “Leaky streams: Identifying variable bitrate dash videos streamed over encrypted 802.11 a connections,” in 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC), 2016.