Fine-grained Classification of Internet Video Traffic from QoS Perspective Using Fractal Spectrum

Pingping Tang, Yuning Dong, *Member, IEEE,* Jiong Jin, *Member, IEEE,* and Shiwen Mao, *Fellow, IEEE*

***Abstract*—Internet video traffic exhibits considerable variation as new video services continue to emerge. Some videos require strict real-time performance, while others may aim for a minimal packet loss rate or sufficient bandwidth. Therefore, it is important to develop fine-grained classification mechanisms to realize effective resource management and quality of service (QoS) provisioning. However, the existing methods for classifying video traffic always suffer from two problems: payload inspection and feature selection. In this paper, we propose a novel method that uses fractal characteristics to achieve traffic classification at a fine-grained level. This method requires neither payload signatures nor statistical features. Through rigorous analysis, we prove the feasibility of employing fractal characteristics for video traffic classification and further develop a theoretical framework for the proposed scheme. For the specific scenario of video flow classification, we improve the theory of fractals in terms of estimated spectrum, core domain, segmentation, and threshold setting. The results of an extensive experimental study on several real-world video traffic datasets show that the classification accuracy of the proposed scheme is higher than that of existing methods.**

***Index Terms*—Fine-grained classification, fractal characteristics, quality of service (QoS), spectrum, video traffic.**

I. Introduction

WITH the development of 4G and 5G technologies, video traffic has become one of the most popular network services, and it is growing rapidly on a tremendous scale [2], [3]. Different video traffic flows have varying requirements for quality of service (QoS) and network resources. For example, video conferencing and telemedicine applications strictly require good real-time performance, and any unexpected delay can result in a wrong decision and cause considerable economic loss [4], [5]. On the other hand, high-quality video streaming requires substantial network bandwidth to provide a good user experience [6]. Internet service providers (ISPs) are expected to allocate suitable network resources for different video flows [7], [8]. Therefore, fine-grained classification of video traffic is necessary for effective network resource management and QoS enforcement [9], [10]. For example, Liu *et al.* [11] presented a transmission delay control module to ensure the on-time arrival of various types of multimedia data, including VoIP (Voice over Internet Protocol), video streaming, and online gaming. They aimed to achieve the best transmission to satisfy diverse user demands. In the system modeling, transmission delay control is based on the initial classification of traffic into different fine-grained types. Lima *et al.* [12] formulated an algorithm, named Reallocation-based Assignment for Improved Spectral Efficiency and Satisfaction (RAISES), to solve the resource assignment problem subject to user satisfaction constraints. In their approach, similar to the method in [11], the flows must first be classified according to their different network resource and QoS requirements, and then RAISES assigns different resources for these different types of flows.

It is apparent that fine-grained classification differs from coarse-grained classification. The latter is used to classify flows into categories such as text flows, voice flows, and video flows, while the former further classifies video flows into multiple classes. An example of coarse-grained classification is the work in [13], which was devoted to distinguishing video flows from non-video flows but could not further classify the video flows into multiple classes.

A. Motivation and Challenges

From the perspective of QoS, the most effective and direct fine-grained classification is to distinguish the video traffic by quality [14]. on the basis of the video quality evaluation standard known as the mean opinion score (MOS), Canovas *et al.* [15] extracted useful traffic patterns from the peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and new quality index (NQI) to classify video flows into five types: non-critical, low critical, some critical, critical, and very critical. However, the international MOS standard only has five levels, i.e., $\{1,2,3,4,5\}$, so the number of classes is limited to five. In order to generate more classes, Yang *et al.* [16] further divided the MOS values into nine levels: $\{1,1.5,2,2.5,3,3.5,4,4.5,5\}$. However, the greater the number of classes defined, the more ambiguous the boundaries between the classes. Quality of experience (QoE) offers another kind of calibration for video quality [17], but it cannot accurately determine the boundaries of classes, either [18].

Therefore, researchers have explored many other methods to define fine-grained classes with clear boundaries, and they

This work was supported in part by the NSFC under Grant 61271233, the NSFC under Grant 61401004, the AHUNSR under Grant KJ2019A0491 and the US NSF under Grant ECCS-1923717.

P. Tang is with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China, and also with the College of Physics and Electronic Information, Anhui Normal University, Wuhu 241000, China (e-mail: tpping@ahnu.edu.cn).

Y. Dong is with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: dongyn@njupt.edu.cn).

J. Jin is with the School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne, VIC 3122, Australia (e-mail:jiongjin@swin.edu.au).

S. Mao is with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201 USA (e-mail: smao@ieee.org).

This work was presented in part at IEEE ISCIT 2017, Cairns, Australia, Sept. 2017 [1]. Manuscript received June 2, 2019; revised October 8, 2019.

have made several achievements [19]. For example, Shim *et al.* [20] proposed an application-level traffic classification method using a payload size sequence signature, which can classify each application's traffic into its respective individual application.

In general, previous works on the fine-grained classification of video flows can be grouped into two main categories: (i) classification based on payload inspection, such as deep packet inspection (DPI) [21], and (ii) classification based on statistical features with machine learning (ML). The first group requires the inspection of the packet's payload to obtain application signatures. Consequently, it has a relatively high accuracy rate [22]. However, it does not perform well for encrypted video flows [23]. The second group requires the extraction of statistical features from given flow samples [24]. Such classification methods involve feature selection, which is usually time-consuming, especially when new applications are generated irregularly [25]. In addition, some of the statistical features are particularly restricted. For example, the feature $X_{4-packet}$ (referring to the size of the first four packets) cannot be obtained if the flow is captured from the middle instead of from the beginning. The feature $X_{max-size}$ (referring to the maximum packet size) can only be obtained at the last moment after all packets have been statistically analyzed.

Furthermore, video flow is affected by a series of complex processes, such as codec design, transport layer protocol, congestion control, retransmission mechanism, and priority. These complex factors are challenging for the classification of video traffic at the fine-grained level.

Motivated by the above observations, we propose a novel method based on fractal characteristics to achieve the fine-grained classification of video traffic with high accuracy.

B. Contributions

The major contributions of this paper are summarized below.

- On the basis of the existing traffic fractal theory, we devised the flow fractal theory with rigorous theoretical proof.
- According to the fractal characteristics of flows, a novel classification method for video flows was developed at the fine-grained level. The proposed scheme addresses some of the drawbacks of existing approaches: (i) It does not require the inspection of the payload content, so it can be used to process encrypted video flows to preserve user privacy. (ii) It avoids the time-consuming process of feature extraction, which is generally required in traditional machine learning methods. (iii) Fractal characteristics are quite different from statistical features and can be obtained at any stage of the flow (in the beginning or middle of the flow).
- Fractal theory has been widely used for classification and detection in fields such as agriculture, medicine, and chemistry. With our new contributions to fractal theory on the aspects of estimated spectrum and core domain, we aim to further promote its development in these fields, in which datasets also exhibit fractal features.

C. Organization of the paper

The remainder of the paper is organized as follows. We discuss related work and introduce the fractals in Section II. The flow fractal theory is theoretically proven in Section III. The proposed classification scheme is described and analyzed in Section IV. We describe the datasets in Section V and present performance evaluation in Section VI. Section VII concludes this paper with a discussion of our future work.

II. Related Work and Preliminaries

Because of its limitations in processing encrypted flows and related privacy concerns, the DPI approach has become nearly obsolete in the classification of fine-grained video flows [26]. Most recent studies have focused on statistical features-based ML methods.

A. Statistical features based ML methods

The procedure of statistical features-based ML methods can be summarized as follows. First, flow samples are observed and analyzed; then, useful features, such as the flow size, transmission rate, duration time, packet number, and average size of packets, are extracted on the basis of statistics. Next, depending on those features, flows can be divided into different classes by ML classifiers, such as support vector machine (SVM), k-Nearest Neighbor (KNN), decision tree, and naive Bayes.

ML methods based on statistical features have been proved to be feasible. For example, Nossenson *et al.* [27] classified videos into live streaming and VOD according to the statistical features of packet length and information offset, among other characteristics. Hao *et al.* [28] investigated the classification of P2P and WWW video flows; in their research, the extracted features, such as maximum packet size and minimum packet size, were assigned suitable weights. Garcia *et al.* [13] used composite (cp) features to quickly distinguish video flows from non-video flows using only the initial 20 packets of a flow. Composite features, such as the size of the largest packet, require minimal computational effort, which contributes to an outstanding execution performance, with 1 million classifications per second. Thay *et al.* [29] provided a classification technique that used the number of peer connections (in both the incoming and outgoing direction) in a 5-minute period to classify P2P traffic in distributed applications, including BitTorrent, Skype, and SopCast. Qin *et al.* [30] aimed to identify VoIP flows in P2P applications by using packet size distribution (PSD) as a feature. However, some important issues still need to be addressed:

(i) It has been recognized that, sometimes, a large number of features can only be used to identify very few classes. For example, Cheng *et al.* [31] extracted more than 10 features from a given dataset to identify YouTube video flows from traditional streaming videos. Takeshita *et al.* [32] designated several features, including packet size and packet number, to identify HTTP video flows.

(ii) Even though such features can be used to effectively classify a specific set of flow samples, they are often not effective for the next set. Any variation in the feature set may lead to considerable computation. For example, Nair *et al.* [33] explored the behavioral patterns of P2P and non-P2P traffic, and they proposed useful features to classify P2P traffic by a decision tree classifier. However, the decision tree needs to be regenerated when features are changed, and the updating process requires a large amount of computation. Wu *et al.* [34] proposed the chain and hierarchical structure (CHS) for the fine-grained classification of network video flows. CHS combines several base classifiers to obtain superior performance and a higher accuracy rate compared with those of a single classifier. However, when the number of classes is increased, the whole classification structure of CHS must be updated; thus, the corresponding feature sets should also be updated, which necessitates enormous computation.

(iii) Generally, better performance can be achieved by adding more features, but this significantly increases the computation and storage costs [35]. On the other hand, there is evidence of a strong correlation among features. More features will lead to higher redundancy, which will greatly reduce the accuracy and efficiency of flow classification [36]. For example, Zhang *et al.* [37] classified flows with naive Bayes, assuming all features to be independent Gaussian distributions. However, the assumption of independence may not hold in the environment of a real network, and thus, the method can only ensure an accuracy rate of about 80% when used in the online environment.

Therefore, existing methods for fine-grained video flow classification may not be effective, and more research is needed to explore new methods.

B. Fractal characteristics based Classification Methods

According to fractal theory, different areas of the same fractal material generally have the same fractal characteristics. Therefore, many researchers have explored the inherent fractal characteristics of objects to distinguish them. For example, Pratiher *et al.* [38] used multifractal parameters of EEG (electroencephalogram) signals for the classification of epileptic seizures. Livi *et al.* [39] applied fractal properties to the discrimination of Parkinsonism. Hernández-Carrasco *et al.* [40] put forward a new approach to classify ocean maps at high resolution using multifractal variables. For the recognition of natural scenes, Al-Saidi *et al.* [41] proposed a new fractal descriptor to classify different land covers. In [42], Akar *et al.* presented a fractal dimension (FD)-based analysis of cerebellar tissues in magnetic resonance (MR) images to identify Chiari Malformation type-I (CM-I) patients. Allwright *et al.* [43] proposed the fractal advection-dispersion equation to achieve the classification of groundwater transport and contamination. Neto *et al.* [44] developed a method to classify the genotype of the wings of *Drosophila melanogaster* flies by combining stationary wavelet transformation, Canny filter, and fractal dimensions. In [45], on the basis of a multifractal downscaling model, the levels of soil moisture were correctly calculated and scaled for different irrigated fields (including semiarid sites, sparser agricultural districts, and temperate regions).

The above analysis demonstrates the wide use of fractal theory for classification and detection in various fields [46], such as agriculture, medicine, and chemistry. However, to date, it has never been applied to the classification of network flows.

C. Preliminaries

Fractal theory was first proposed by Mandelbrot, who recorded his findings in the book "The Fractal Geometry of Nature", published in 1983. He found that many objects in nature show the property of self-similarity. For example, a small part of a leaf is quite similar to the whole leaf.

α, called the Holder exponent or the singularity exponent, is used to describe the fractal characteristic of an object. Here, we use a simple and comprehensible description to demonstrate the calculate of α. Suppose that the sides of a large square are 1, and use a small square with the scale $r = \frac{1}{2}$ to segment the large square. To cover the large square, we need $N(r)$ small squares, that is, $N(\frac{1}{2}) = \frac{1}{(\frac{1}{2})^2} = (\frac{1}{2})^{-2}$. If $r = \frac{1}{4}$, then $N(\frac{1}{4}) = (\frac{1}{4})^{-2}$. In general, if $r = \frac{1}{k}(k = 1, 2, 3, \cdots)$, then $N(\frac{1}{k}) = (\frac{1}{k})^{-2}$.

Similarly, we use $N(r)$ small boxes with scale r to cover a d-dimensional object. Then, the relationship between $N(r)$ and r is

$$N(r) = r^{-d}, \tag{1}$$

that is,

$$d = \frac{\ln N(r)}{\ln(1/r)}, \tag{2}$$

where d is the fractal characteristic α. In this case, the object only has one fractal characteristic, so we describe it as single-fractal. Some objects, such as network traffic, have several fractal characteristics, so they are multifractal. In 1993, Leland *et al.*, who analyzed captured Ethernet traffic using several statistical tools, were the first to discover that network traffic is multifractal. Consequently, they proposed the traffic multifractal theory [47].

According to the traffic multifractal theory, if each unit k has the fractal characteristic α_k, then

$$\mu_k(\varepsilon) \propto \varepsilon^{-\alpha_k}, \tag{3}$$

where $\mu_k(\varepsilon)$ represents the measurement of subset k of scale ε, and α_k is the fractal characteristic of subset k. Then, the fractal spectrum $f_G(\alpha)$ can be described as

$$N(\alpha) \propto \varepsilon^{-f_G(\alpha)}, \tag{4}$$

where $N(\alpha)$ denotes the number of subsets with a value of α under scale ε.

In general, the above Holder exponent α, known as the single-fractal, is the core concept of the fractal theory proposed by Mandelbrot. The fractal spectrum $f_G(\alpha)$ proposed by Leland *et al.* (termed multifractal) is a significant improvement in fractal theory. Broadly speaking, the single-fractal and multifractal are called fractal in this paper.

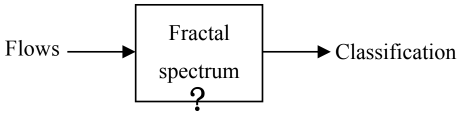

Fig. 1: Block diagram for Section III.

TABLE I: glossary of key variables and abbreviations

Symbol	Description
C_i	video classes
$c(q)$	the amount of computation caused by q order calculation
E	the target values of $c(q)$
$I = [t_1, t_2]$	time interval
IM	Instant messaging videos
F	flow
$f_G(\alpha)$	fractal spectrum
far	the false acceptance rate
frr	the false rejection rate
L	the number of classes
m	the scale of merge
M	the number of flows
$M_f(\quad)$	the fractal classification model
N	flow $F(t)$ is sampled in N resolution
$N(\alpha)$	the number of subsets with the value of α
PU	P2P unidirectional videos
MG	Massive muti-player online role-playing game
$Q\|_{(-q,+q)}$	core domain
QoS	quality of service
q	orders in fractal spectrum computation
$S_m(q)$	the sum of m-merged flow sequence
s, S	the number of segments
t, T	threshold
$v(q)$	the changing rate of $\tau(q)$ curves
V	the target values of $v(q)$
X, Y	traffic
α	the Holder exponent
$\tau(q)$	scaling function
$\Delta^k[F]$	discrete flow sequence
ε	scale in the fractal theory
ζ	infinitesimal quantity
$\rho(\quad)$	the correlation function of $\tau(q)$
σ	the mean value of $\tau(q)$
λ_j, λ_k	the weights of $v(q)$ and $c(q)$
$\mu(\varepsilon)$	the measurement of scale ε
ϕ	the coefficient of difference between spectra

III. Flow Fractal Theory

A. Assumption

Fractal characteristics are often used to distinguish materials at a fine-grained level, and this inspired us to classify flows on the basis of fractal characteristics. However, an important question is raised, as shown in Fig. 1: Are flows fractal?

To this end, although Leland *et al.* proved the fractal characteristics of **traffic**, the fractal characteristics of **flows** have never been explored. Therefore, we make an important assumption: *if flows are as multifractal as general network traffic, then the fractal characteristics of flows can be used to classify them at the fine-grained level*. If our assumption is correct, then the novel classification method based on the fractal characteristics of flows does not need to inspect the payload content or extract features through statistical analysis, thus addressing the issues in fine-grained classification methods discussed in Section I-A.

In the next subsection, we theoretically prove that flows are as multifractal as network traffic. For ease of reading, the mathematical symbols and variables used in this paper are listed in Table I.

B. Fractal Characteristics of Flows

Leland *et al.* proved that network traffic is multifractal. In this paper, we aim to prove that flows are also multifractal. The theoretical proof consists of two steps: (i) sufficiency: that is, given the condition that traffic is multifractal, we aim to prove that flow is multifractal; (ii) necessity: that is, given the condition that flow is multifractal, we aim to prove that traffic is multifractal.

i) Sufficiency

According to the traffic fractal theory, *traffic* is defined as the amount of data transmitted through a network device or a transmission medium per time unit $X = \{X(t) : t = 1, 2, 3, \cdots\}$ [47], while *flow* is defined as a set of packets with the same properties of *<Src IP, Dest IP, Src Port, Dest Port, Protocol>* [48]. In order to present the fractal spectrum of flows, we redefine flow $F = \{F(t) : t = 1, 2, 3, \cdots\}$ as the amount of data with the same properties of *<Src IP, Dest IP, Src Port, Dest Port, Protocol>* transmitted through a network device or a transmission medium per time unit.

According to multifractal theory, the fractal spectrum of traffic X is $f_X(\alpha)$. Now, we define special traffic X_s:

$$X_s = X|_{<SrcIP_s, DestIP_s, SrcPort_s, DestPort_s, Protocol_s>}. \quad (5)$$

Therefore, the fractal spectrum of traffic X_s is $f_{X_s}(\alpha)$. According to the definition of flow, traffic X_s is flow F_s. Then, flow F_s has the fractal spectrum $f_{X_s}(\alpha)$.

The sufficiency is thus proved: if traffic is multifractal: the fractal spectrum of traffic X is $f_X(\alpha)$, then flow F is also multifractal: its fractal spectrum is $f_F(\alpha)$.

ii) Necessity

Suppose that there are two flows: X and Y. The fractal spectra of flow X and Y are $f_{G_1}(\alpha)$ and $f_{G_2}(\alpha)$, respectively. X and Y are aggregated into traffic $Z = X + Y$. According to the above definitions of flow and traffic,

Flow X+Flow Y $\rightarrow$Traffic Z (not Flow Z);

Then, $f_{G_1}(\alpha) \oplus f_{G_2}(\alpha) \rightarrow ?$ Here, we use the symbol $\oplus$ to represent the possible superposition of the fractal spectra of X and Y. We are not sure what happens to the spectra of X and Y after they are aggregated. Maybe it turns out to be nothing! Now we calculate $f_{G_1}(\alpha) \oplus f_{G_2}(\alpha)$ as follows.

According to Proposition 1 (see the Appendix),

$$\inf(f_{G_1}(\alpha) \oplus f_{G_2}(\alpha)) = \frac{1}{2}(f_{G1}(\alpha) + f_{G2}(\alpha)), \quad (6)$$

$$\sup(f_{G_1}(\alpha) \oplus f_{G_2}(\alpha)) = \max(f_{G1}(\alpha), f_{G2}(\alpha)). \quad (7)$$

According to Proposition 2 (see the Appendix),

$$\frac{d(f_{G_1}(\alpha) \oplus f_{G_2}(\alpha))}{d\alpha} = \frac{f'_{G_1}(\alpha) + f'_{G_2}(\alpha)}{f_{G_1}(\alpha) + f_{G_2}(\alpha)} \max\left(f_{G_1}(\alpha), f_{G_2}(\alpha)\right). \quad (8)$$

It can be seen from (6)–(8) that the superimposed spectra of X and Y are determined by $f_{G_1}(\alpha)$ and $f_{G_2}(\alpha)$. Therefore, we provide a new definition of $f_{G_1}(\alpha) \oplus f_{G_2}(\alpha)$:

$$f_G(\alpha) = f_{G_1}(\alpha) \oplus f_{G_2}(\alpha). \quad (9)$$

Eq. (9) means that when X and Y are aggregated into traffic Z, the superimposed spectrum of Z is $f_G(\alpha)$, which is determined by $f_{G_1}(\alpha)$ and $f_{G_2}(\alpha)$. Thus, the necessity is proved.

Therefore, the two conditions—traffic is multifractal, and flow is multifractal—are verified to be mutually necessary and sufficient. Since traffic is multifractal, flow must also be multifractal.

In addition, we can derive another important conclusion from (6)–(8). When $f_{G_1}(\alpha) = f_{G_2}(\alpha) = f_G(\alpha)$, we have

$$\inf(f_{G_Z}(\alpha)) = \sup(f_{G_Z}(\alpha)) = f_G(\alpha), \tag{10}$$

$$f'_{G_Z}(\alpha) = f'_G(\alpha). \tag{11}$$

(10) and (11) indicate that, if flow X belongs to the same class as flow Y, then the aggregated flow $Z = X + Y$ will fall into the same class; if flow X is different from flow Y, then the aggregated flow Z does not belong to the flow of either X or Y. That is, each class of flow has a unique spectrum that can be used to identify it. Next, on the basis of the traffic multifractal theory, we describe the procedure for calculating the fractal spectrum of flow. As in the case of traffic $X(t)$, flow $F(t)$ is a stochastic process. A time interval $I = [t_1, t_2]$ can be divided into N subsections:

$$I^k = \left[t_1 + \frac{k}{N}(t_2 - t_1), t_1 + \frac{k+1}{N}(t_2 - t_1)\right], \tag{12}$$

where $k = 0, 1, \cdots, (N-1)$, and N is defined as the resolution. In order to simplify the calculation, we assume that $t_1 = 0$, $t_2 = 1$, and $N = 2^n$. Therefore, Eq. (12) is simplified to $I^k = [k2^{-n}, (k+1)2^{-n}]$. Thus, flow $F(t)$ is sampled and converted to a discrete sequence. An *increment process* of flow $F(t)$ involves the same sampling process, and the discrete flow sequence is

$$\Delta^k[F] = \left|F((k+1)2^{-n}) - F(k2^{-n})\right|. \tag{13}$$

On the basis of (13), the merged sequence for calculating the fractal spectrum can be described as

$$\Delta^k_{\frac{n}{m}}[F] = \left|F((k+1)2^{-\frac{n}{m}}) - F(k2^{-\frac{n}{m}})\right|, \tag{14}$$

where $m = 1, 2, \cdots, N$. According to (3) and (14), the Holder exponent α for the flow sequence can be obtained as

$$\alpha^k_{\frac{n}{m}} = -\frac{m}{n}\ln\left|F((k+1)2^{-\frac{n}{m}}) - F(k2^{-\frac{n}{m}})\right|. \tag{15}$$

On the basis of (4) and (15), the fractal spectrum of the flow sequence is derived as

$$f_G(\alpha) \overset{\Delta}{=} \lim_{\varepsilon\to 0}\lim_{n\to\infty}\frac{1}{n}\ln N(\alpha^{n}_{\frac{N}{m}})\Big|_{\alpha^{n}_{\frac{N}{m}}\in(\alpha-\varepsilon,\alpha+\varepsilon)}. \tag{16}$$

IV. Fractal Classification Model

A. Problem Statement

At present, network devices based on xFlow technology (such as Netflow and OpenFlow) can be used to easily divide bitstreams into flows. Then, these flows are grouped into different classes. ISPs will allocate suitable network resources for different classes to meet the flows' QoS requirements.

Mathematically, the classification model can be defined as

$$M_f = (F, f_G \bigcup \{C_i\}), \tag{17}$$

where $\boldsymbol{F}$ denotes a group of flows, and $\boldsymbol{f_G}$ refers to the fractal spectrum $f_G(\alpha)$. By using $\boldsymbol{f_G}$, these flows are classified into class C_i, $i = 1, 2, \cdots, L$, where L is the number of classes.

In the model of (17), it is non-trivial to employ the fractal spectrum to classify flows, as shown in Fig. 2.

(i) According to fractal theory, it is difficult to calculate the fractal spectrum $f_G(\alpha)$ by (16). Are there any alternative, more computationally effective ways to compute the fractal spectrum $f_G(\alpha)$? In Section IV-B, we explore the relationship between the scaling function $\tau(q)$ and the fractal spectrum $f_G(\alpha)$; then, the fractal spectrum can be described by $\tau(q)$ instead of $f_G(\alpha)$.

(ii) For $\tau(q)$, the range of q should be $(-\infty, +\infty)$. Can we narrow it down to speed up computation without sacrificing performance? In practical implementation, we find that when q exceeds a certain level, further increasing its value does not achieve significant gains in the results. Hence, we assert that the range of q can be reasonably narrowed. Thus, the core domain is defined as $Q|(-q, +q)$ in Section IV-C.

(iii) Overall, $\tau(q)$ is the estimated fractal spectrum of $f_G(\alpha)$, as shown in Section IV-B. How can a stable spectrum of $\tau(q)$ be obtained to achieve a stable classification? As shown in Section IV-D, we solve this problem by segmenting the flow sequence.

(iv) How can the spectra of two different flows be compared? In Section IV-E, the differences between spectra are calculated by the gray correlation, which is generally used to quantitatively measure the similarity between curves.

Before proceeding, we emphasize the following two points:

- We classify flow $F(t)$ according to its bitstream (see the definition of flow in Section III-B). Our method does not check the payload content, so it is able to deal with encrypted video flows without breaching user privacy.
- In Section III-B, we calculate the fractal spectrum during the time interval $I = [t_1, t_2]$ of $F(t)$. That is, the fractal characteristics can be obtained at any stage of the flow's lifetime (in the beginning, in the middle, and even near the end of the flow), which is quite different from statistical features.

The process of our fractal classification $M_f(\quad)$ is illustrated in Algorithm 1.

Algorithm 1: Fractal classification of traffic flows

```
Input flow F(t);
Thus achieve flow sequence Δ^k[F] by (13);
Partitioned flow sequence into S segments (see Section IV-D);
Calculate spectrum τ_s^f(q) in the core domain Q|_(-q,+q) (see
Sections IV-B and IV-C).
For each class c_l|_(l≤L):
      Compare τ_s^f(q) with typical spectrum P_l : τ_s^(p_l)(q) (see
Section IV-E );
      Difference between spectra is φ_l (see Section IV-E):
Select the minimum φ_l;
If φ_l ≤ T_l, then F and P_l are the same class.
```

B. Fractal Spectrum $f_G(\alpha)$ and Scaling Function $\tau(q)$

In accordance with the theory of fractals, it is difficult to accurately calculate the fractal spectrum $f_G(\alpha)$ with (16). Therefore, in this paper, the fractal spectrum of flow is

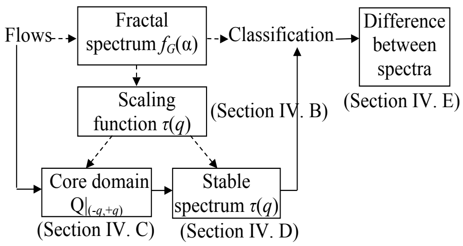

Fig. 2: Block diagram for Section IV.

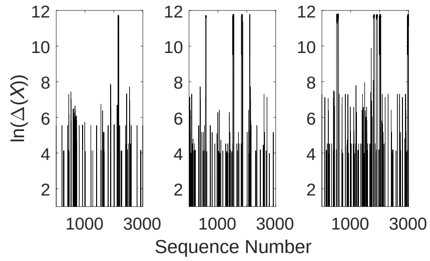

Fig. 3: Flow sequences of SD (left), HD (middle), and UD (right) by Youku. $\Delta(X)$ is the amount of bits (in Section IV-B).

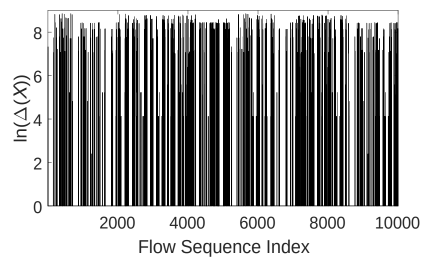

Fig. 4: Flow sequence of IM flow.

modeled by exploring the estimated spectrum on the basis of the Legendre transformation [49]. Specifically, we focus on deriving the relationship between the scaling function $\tau(q)$ and the fractal spectrum $f_G(\alpha)$. Then, $\tau(q)$, instead of $f_G(\alpha)$, is used to describe the fractal characteristics of flows. Now, we introduce the scaling function $\tau(q)$.

First, the flow sequence in (14) should be normalized to process data from different sources:

$$\Delta^{k}_{\frac{N}{m}}[\bar{F}] = \frac{\Delta^{k}_{\frac{N}{m}}[F]}{\sum_j \Delta^{j}_{\frac{N}{m}}[F]}. \tag{18}$$

We define the scaling function [49] as follows:

$$\tau(q) \stackrel{\Delta}{=} \lim_{m\to\infty} \frac{1}{m} \log S_m(q), \tag{19}$$

$$S_m(q) \stackrel{\Delta}{=} \sum_{k=1}^{\frac{N}{m}} \left| \sum_{j=1}^{m} \Delta^{(\mathrm{m}(k-1)+j)}_{\frac{N}{m}}[\bar{X}] \right|^q. \tag{20}$$

According to (15), $S_m(q)$ can be further defined as

$$S_m(q) = \sum_{k=0}^{2^n-1} 2^{-nq\alpha^{k}_{\frac{N}{m}}} \geq \sum \left(2^{-n\alpha}\right)^q = 2^{-n(\alpha q - f_G(\alpha))}. \tag{21}$$

On the basis of (19) and (21), the relationship between $\tau(q)$ and $f_G(\alpha)$ can be derived as follows:

$$\tau(q) = {f_G}^{*}(\alpha) \stackrel{\Delta}{=} \inf_{\alpha} \left(\alpha q - f_G(\alpha)\right), \tag{22}$$

where $*(\cdot)$ denotes the Legendre transformation, and $f_G(\alpha)$ is the estimated spectrum derived from the Legendre transformation of $\tau(q)$.

In accordance with the Gärtner–Ellis theorem [50], if $\tau(q)$ exists and is differential, then the estimated spectrum $f_G(\alpha)$ derived from the Legendre transformation of $\tau(q)$ is a minimum bias estimator. The estimated spectrum of $f_G(\alpha) \sim \alpha$ can be calculated from the scaling function curve of $\tau(q) \sim q$:

$$\begin{cases} f_G(\alpha) = \alpha q - \tau(q) \\ \alpha = \frac{d\tau(q)}{dq}. \end{cases} \tag{23}$$

From (23), the Legendre transformation of $\tau(q)$ can be used to represent fractal characteristics. $f_G(\alpha)$ is a convex function, $0 \leq f_G(\alpha) \leq max = f_G(\alpha_0)$. Therefore, the shape of $\tau(q)$ is a monotonically increasing curve. After taking the extremum of the derivative of $\tau(q)$ with respect to q, the extreme values of the Holder exponent α can be obtained:

$$\begin{cases} \alpha_{\min} = \lim_{q\to+\infty} \frac{d\tau(q)}{dq} \\ \alpha_{\max} = \lim_{q\to-\infty} \frac{d\tau(q)}{dq}. \end{cases} \tag{24}$$

In conclusion, $f_G(\alpha)$ is derived from the Legendre transformation of $\tau(q)$, and the $\tau(q) \sim q$ curve uniquely maps to $f_G(\alpha) \sim \alpha$. Therefore, instead of $f_G(\alpha)$, $\tau(q)$ is used to represent the fractal characteristics of flows in this paper.

C. Core Domain

As in (24), the range of q should to be $(-\infty, +\infty)$. In practice, however, we find that the workload rises exponentially with the increase of q. Especially when q exceeds a certain level, it has no significant effect on the results. Therefore, the range of q can be reasonably narrowed down to $Q|(-q,+q)$, which we call the core domain. Of course, if $Q|(-q,+q)$ is too small, leading to serious defects of curve, the fragment of curve $\tau(q)$ cannot offer sufficient details of the fractal characteristics. Therefore, $Q|(-q,+q)$ should be properly selected to reduce the workload and provide enough details of the fractal characteristics. The optimal range of q can be determined with the following procedure. First, we define $v(q)$ as the changing rate of $\Delta\tau(q)$ caused by Δq.

$$v(q) = \left| \frac{d^2\tau(\ddot{q})}{d\ddot{q}^2} \, |_{\ddot{q}=+q} \right| + \left| \frac{d^2\tau(\ddot{q})}{d\ddot{q}^2} \, |_{\ddot{q}=-q} \right|. \tag{25}$$

According to (25), $v(q)$ is an even function. From (23) and (24), when $q \to \pm\infty$, $v(q) \to 0$, but the computational complexity is exponentially increased:

$$c(q) = \theta e^q, \tag{26}$$

where $c(q)$ refers to the amount of computation. The parameters of $c(q)$ can be obtained through curve fitting.

The challenge is in obtaining a balance between $v(q)$ and $c(q)$ by tuning q. On the one hand, $c(q)$ should be as small as possible, which means q should be small. On the other hand, $v(q)$ should be small, which means q should be large. Therefore, on the basis of the weighted sum of squares (WSOS), we propose the optimization model in (27) to reach an appropriate trade-off between $v(q)$ and $c(q)$.

$$q \stackrel{\Delta}{=} \left\lceil \arg\min \left(\lambda_j (v(q) - V)^2 + \lambda_k (c(q) - E)^2\right) \right\rceil, \tag{27}$$

where V and E are the target values of $v(q)$ and $c(q)$, respectively. On the one hand, q should be large enough to ensure that $v(q)$ is as small as possible, so the value of V is an infinitesimal quantity ζ. On the other hand, q should be small enough to ensure that $c(q)$ is as small as possible. Therefore, the ideal value of E is also ζ.

In (27), λ_j and λ_k are the weights that satisfy $\lambda_j + \lambda_k = 1$ and $\lambda_j \gg \lambda_k$. We use $\lambda_j \gg \lambda_k$ because $v(q)$ and $c(q)$ have different values. In this paper, the magnitude of $v(q)^2$ is about 10^{-3}, while that of $c(q)^2$ is about 10^{-1}. Therefore, λ_j should be much larger than λ_k. For example, if λ_j = 0.99, and λ_k = 0.01, then the value of $\lambda_j v(q)^2$ is about 10^{-3}, and $\lambda_k c(q)^2$ is also about 10^{-3}. In this manner, we can strike a balance between $v(q)$ and $c(q)$ and obtain the core domain $Q|_{(-q,+q)}$ using (27).

D. Segmentation

According to the Gärtner-Ellis theorem [50], if $\tau(q)$ exists and is differentiable, then $f_G(\alpha)$ derived from the Legendre transformation of $\tau(q)$, as in (23), proves to be a minimum bias estimator. Thus, spectrum $\tau(q)$ can be regarded as a mathematical representation of complex fractal characteristics, which are estimated by (18)–(22). The estimation results in the spectra of the same class are usually slightly different. In special cases, such as when ϕ is close to the threshold, it will lead to inaccurate classification (see Section VI-C). To address this problem, we segment the flow sequence.

We first divide the flow sequence into several segments; then, we calculate $\tau(q)$ of each segment and, finally, obtain the superimposed spectrum of all the segments. Compared with the non-segmentation approach, segmentation can reduce randomness and obtain stabler classification results.

The flow sequence should be divided into S segments. However, for a given resolution N, too many segments will cause deviation of the spectrum estimation. Hence, S needs to be optimized, as shown below. A flow sequence $\{F(t)\}$, $t = 0, 1, \cdots, N-1$, is partitioned into S segments:

$$\begin{aligned} X^s = \{&\Delta^{\frac{Ns}{S}}[\bar{F}], \Delta^{\frac{Ns}{S}+1}[\bar{F}], \Delta^{k+\frac{Ns}{S}+2}[\bar{F}], \\ &\cdots, \Delta^{\frac{N(s+1)}{S}-1}[\bar{F}]\},\ s = 0, 1, \cdots, S-1. \end{aligned} \tag{28}$$

For the rate of $\Delta\tau(q)$ caused by Δq, a correlation function can be defined as

$$\begin{aligned} \rho(k,q) = \frac{\sigma^2}{2} \Bigg(&(k+1)^{2^{\frac{d^2\tau(\ddot{q})}{d\ddot{q}^2}}} - 2k^{2^{\frac{d^2\tau(\ddot{q})}{d\ddot{q}^2}}} \\ &+ (k-1)^{2^{\frac{d^2\tau(\ddot{q})}{d\ddot{q}^2}}} \Bigg)\Bigg|_{\ddot{q}=q}, \end{aligned} \tag{29}$$

where σ is the mean value of $\tau(q)$. The correlation function of segment s is

$$\begin{aligned} \rho_s(k;q) = \frac{\sigma_s^2}{2} \Bigg(&(k+1)^{2^{\frac{d^2\tau_s(\ddot{q})}{d\ddot{q}^2}}} \\ &-2k^{2^{\frac{d^2\tau_s(\ddot{q})}{d\ddot{q}^2}}} + (k-1)^{2^{\frac{d^2\tau_s(\ddot{q})}{d\ddot{q}^2}}} \Bigg)\Bigg|_{\ddot{q}=q}, \end{aligned} \tag{30}$$

where σ_s is the mean value of $\tau_s(q)$. We then construct a cost function based on the correlation information:

$$J\left(\sum_{s=0}^{S-1} \tau_s(q)\right) = \mathrm{E}\left[\|\boldsymbol{\rho}_S - \gamma\boldsymbol{\rho}\|^2\right], \tag{31}$$

where $\boldsymbol{\rho}$ represents the matrix form of $\rho(k,q)$, $\boldsymbol{\rho}_S$ represents the matrix form of $\bigcup_S \rho_s(k;q)$, and γ is a regulatory factor, which represents the degree of consistency of information carried by the original sequence and the segmented sequence.

The optimization objective is to minimize the cost function, while the number of segments S needs to be sufficiently large:

$$S^* \triangleq \arg\min J\left(\sum_{l=0}^{\max(S)-1} \tau_s(q)\right). \tag{32}$$

As shown in (32), the minimum and maximum objective function (MMOF) results in the optimal segment number S^*.

E. Calculating Spectrum Differences using Grey Correlation

In Section IV-D, flows a and b are each divided into S^* segments. Next, spectra $\tau_s^a(q)$ and $\tau_s^b(q)$ are calculated as in (20) and (21) in the core domain $Q|_{(-q,+q)}$. In this subsection, the difference between spectra is calculated according to the *gray correlation*, which is generally used to quantitatively measure the similarity between curves [51]. We define the coefficient of difference between spectra as

$$\phi = \left(\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_{ij}\right)^{-1}, \tag{33}$$

where n is the number of samples on each curve, and

$$\gamma_{ij} = \frac{\min\limits_{1\le i\le n}\min\limits_{1\le j\le n}\{\Delta_{ij}\} + \beta\max\limits_{1\le i\le n}\max\limits_{1\le j\le n}\{\Delta_{ij}\}}{\Delta_{ij} + \beta\max\limits_{1\le i\le n}\max\limits_{1\le j\le n}\{\Delta_{ij}\}} \tag{34}$$

$$\Delta_{ij} = \sum_{s=1}^{S^*}\left|\tau_s^a(q_i) - \tau_s^b(q_j)\right|, \tag{35}$$

β is the resolution factor in $[0,1]$ and represents the proportion of the difference (generally set to 0.5), and ϕ is the coefficient of difference between spectra with the range of space $(1,+\infty)$. The smaller the value of ϕ, the greater the similarity between the two spectra. On the basis of ϕ, the typical spectrum P_l is defined as follows.

Suppose there are L classes: $\{C_l\}_{l=1}^L$. Each class has several flows: $C_l = \{\cdots, F_j, F_k, \cdots\}$). L classes correspond to L typical spectra: $\{P_l\}_{l=1}^L$. ϕ obeys the random distribution between $0 \sim 1$. Therefore, the typical spectrum P_l can be obtained as

$$P_l \triangleq \min_{F_k\in C_l}\left\{\max_{j\ne k, F_j\in C_l}\phi(F_k,F_j)\,|_{F_k}\right\}. \tag{36}$$

From (36), the coefficient of difference ϕ between P_l and all the flows $\{\cdots, F_j, F_k, \cdots\}$ in C_l is the minimum; then, this central spectrum P_l can represent class C_l.

F. Setting the Threshold

As shown in Algorithm 1, it is important to properly set the threshold for classification because it affects the performance of the entire system. Generally, recognition systems use relatively simple methods, such as the receiving operating characteristic (ROC) curve, to determine an appropriate threshold. It is assumed that the threshold is optimal when $frr = far$, that is, $T^*|_{frr=far}$, where frr is the *false rejection rate* and far is the *false acceptance rate*. However, the classification of flows at a fine-grained level requires the smallest overall false rate, that is, $T^*|_{\min(frr+far)}$. Therefore, we adopted the maximum between-class variance (Otsu) method [52] to establish an adjustment mechanism for the global optimal threshold.

$$T^* = \arg\max \sum_{i \neq j} \sigma_B^2(t; C_i \leftrightarrow C_j), \tag{37}$$

where $\sigma_B^2(t; C_i \leftrightarrow C_j)$ is the variance between classes C_i and C_j when the threshold is set to t. According to Otsu, the maximum variance between classes implies the smallest false rate. In addition, in order to increase the convergence speed, we set the termination condition of the algorithm to $\sigma(k)$, approximately equal to $\sigma(k-1)$ as in Algorithm 2. Δ is calculated by the *dichotomy method* [53], and thus, the iterative calculation of T^* is linearly convergent. The size of the convergence step is 0.5, which means the interval will shrink by a ratio of 0.5 in each iteration.

Algorithm 2: Setting the threshold

Initialization: $k = 1$, $\sigma(k) = \infty$, $t(k) = t_0$;
Do { $\sigma(k+1) = \sum_{i \neq j} \sigma_B^2(t(k); i \leftrightarrow j)$;
If $(\sigma_{k+1} > \sigma_k)$ $t(k+1) = t(k) + \Delta$;
Else If $(\sigma_{k+1} < \sigma_k)$ $t(k+1) = t(k) - \Delta$;
Else $t(k+1) = t(k)$;
$k++$;
} While $(\sigma(k) - \sigma(k-1) > \varepsilon)$;
Output: $T^* = t(k)$;

G. Computational and Space Complexity

Our proposed method groups flows into different classes on the basis of fractal characteristics. Flow is defined as discrete sequence $F(t)(t = 0, 1, \cdots, (N-1))$, where N is the resolution. Both the core domain $Q|_{(-q,+q)}$ and threshold are determined in the training phase, and the complexity of the training is $O(\max(M\log(\varepsilon^{-1}), MNQ\log(N)))$, where M is the number of flows, Q is the boundary value of the core domain, and ε is the termination criterion of the threshold iterations.

In the testing phase, as shown in Algorithm 1, the calculation of classification contains nine steps.

Steps 1–2: From flow $F(t)$, we obtain the flow sequence $\Delta^k[F]$ by (13), where N is the resolution. The time complexity is $O(N)$.

Steps 3–4: On the basis of the above sequence $\Delta^k[F]$, we use (19) to compute the spectrum $\tau_s^f(q)$ in the core domain $Q|(-q,+q)$ for each segment S_i. The time complexity mainly lies in the calculation of $S_m(q)$, so it is $O(SN\log(N))$.

Steps 5–7: For each class c_l, we compare $\tau_s^f(q)$ with the typical spectrum P_l. The time complexity of this comparison is $O(Q^2S)$. Note that there are L classes, so the time complexity is $O(LSN\log(N) + Q^2SL)$.

Steps 8–9: The time complexity is $O(L)$.

Therefore, the complexity of Algorithm 1 is $O(N + SLN\log(N) + Q^2SL + L)$. Since $N = 10000$ is significantly greater than $Q = 15$, $S = 8$, and $L = 23$, the time complexity can be simplified to $O(SLN\log(N))$. Here, we can also see that the total time complexity mainly lies in the calculation of $S_m(q)$. When M flows are classified, the complexity in the testing phase is $O(MSLN\log(N))$. Note that segmentation only occurs when ϕ is extremely close to the threshold. For most of the flows (more than 95%) in Section VI, we do not need to implement the segmentation, so S=1. Then, the time complexity can be further simplified to $O(MLN\log(N))$.

Just as the time complexity mainly lies in the calculation of $S_m(q)$, the space complexity of Algorithm 1 mainly focuses on the storage of $S_m(q)$. For each segment, we need space $O(N/S\log(N/S))$. Therefore, for S segments, the total space complexity is $O(N\log(N/S)) \approx O(N\log(N))$. When M flows are classified into L classes, the overall space complexity is $O((M+L)N\log(N))$.

V. Video Classes and Datasets

In the field of traffic classification, the first important issue is defining the classes, such as port-based classes (e.g., VPN), protocol-based classes (e.g., HTTP), quality-based classes (e.g., 5 levels of MOS), application-based classes (e.g., YouTube)$\cdots$ What a variety! In particular, we aim to classify flows at a fine-grained level. At present, application-based classes are considered to be the finest-grained. However, after carefully observing the datasets, we found the following:

(i) Some applications, such as QQ and WeChat, were developed with similar mechanisms and thus have similar network resource requirements (e.g., buffer, priority) during scheduling and transmission, often generating similar types of video bitstreams.

(ii) One application can generate different types of flows. For example, Youku can basically generate three flow types: SD, HD, and UD, which refer to three different resolutions: SD ($\leq 480p$), HD ($720p$), and UD ($\geq 1080p$). SD/HD/UD video requires a bitrate of 1/1.5/3.5 Mbps for H.264 and 2/3/5 Mbps for MPEG-4. Achieving ideal playback qualities of SD, HD, and UD requires that service providers and network operators implement different transmission strategies and protocols. Consequently, there are basically three types of bitstreams for Youku: SD, HD, and UD, as shown in Fig. 3:

Youku $\rightarrow$ SD, HD, UD

(iii) One application can freely switch back and forth between encrypted and unencrypted patterns. The fact that we did not find such videos in our datasets does not mean they do not exist. We can foresee the flows generated by this application must be changed, including the variance of packet sizes, skew of packet sizes, number of bytes, etc.

Similar to SD, HD, and UD, encrypted and unencrypted flows can substantially differ. We can define two subclasses for each application that can switch between encrypted and unencrypted patterns as follows.

Application A $\rightarrow$ encrypted and unencrypted

In summary, different applications may generate similar types of bitstreams, while the same application may generate different types of bitstreams. The type of bitstream is affected by a series of complex factors, including the codec design, transport layer protocol, congestion control mechanism, retransmission of lost packets, and priority, which form the

TABLE III: DATASETS

Dataset	Traces	Flows	Label of classes
NJUPT	NJUPT	106k	3,7,8,11,12,20
IU	ISP+UNB	138K	1,2,3,7,8,11,12,13,14,15,16, 17,18,19,20
NIUI	NJUPT+ISP +UNB+UNIBS	180K	1,2,3,4,5,6,7,8,9,10,11,12, 13,14,15,16,17,18,19,20

NRQ (Network Resource and QoS Requirement) classes. The mapping relationships between the labels, NRQ classes, and applications are shown in Table II.

Moreover, we cannot provide QoS restrictions for each NRQ class because some of the NRQ classes are not differentiated according to QoS provisioning but rather their transmission mechanisms, such as P2P unidirectional videos and P2P bidirectional videos or encrypted and unencrypted flows.

Four types of network traffic traces were used in this study:

- The *NJUPT traces* were captured by Wireshark in the campus network of Nanjing University of Posts and Telecommunications. The traces were preprocessed by using Linux shell scripts and divided into five-tuple flow sequences, as described in Section III.
- The *ISP traces* were collected at a leading ISP of China located in City A in southern China (the name is omitted for commercial confidentiality), and they contained important monitoring and conferencing videos, such as Ezviz and Gotomeeting.
- The *UNB ISCX Network Traffic (VPN-nonVPN) traces* contained a lot of network applications, such as Vimeo, YouTube, ICQ, Skype, Facebook, and BitTorrent. ISCXFlowMeter [54] was used to read the full payload trace (a total of 28 GB) and create the csv file using selected features.
- The *UNIBS-2009 traces* [55] were collected from the edge router of the campus network of the University of Brescia, and they included the applications Edonkey, Skype, and BitTorrent.

From the above traces, we obtained several datasets, as summarized in Table III.

VI. Performance Evaluation

In this research, we explored the fractal spectra of flows to achieve the fine-grained classification of video traffic. Thus, in this section, we first demonstrate the fractal spectrum of a flow. After that, we discuss key parameters, such as the core domain Q and segmentation S used in our proposed scheme. Then, we compare the classification results with several state-of-the-art methods, and finally, we analyze the computational and space complexity from both experimental and theoretical perspectives.

A. Evaluating the $\tau(q)$ Spectrum of a Single Flow

In this experiment, we used instant messaging video flows (IM flows) generated by applications such as QQ and WeChat. The flow sequence is shown in Fig. 4. The duration was set to 100 s, and the resolution N was set to 10000. Therefore, the maximum of $\ln(m)$ is $\ln(10000) = 9.21$, which is sufficient to achieve a reliable estimation of $\tau(q) \sim q$ from $\ln(S_m(q)) \sim \ln(m)$, as shown in Fig. 5.

Fig. 5 shows approximately straight lines with different slopes according to different values of m and q. If the flow sequence is not fractal, then there will be no such straight lines. It is the slopes of these lines that form the scaling function space. Then, according to (19), the curve of $\tau(q) \sim q$ can be plotted with the least square method (LSM), as shown in Fig. 6. Here, note that:

- When $q \to 1$, the line of $\ln(S_m(q)) \sim \ln(m)$ is parallel to the horizontal coordinate axis.
- Regardless of the value that q takes, when $m \to N$, $\ln(S_m(q)) \to 0$. In addition, the slope of the line is positive when q is positive, and the slope of the line is negative when q is negative.
- Regardless of the type of flow, when $q = 0$, the slope of the line $\ln(S_m(q)) \sim \ln(m)$ is the same. In other words, the curves of $\tau(q) \sim q$ of all flows intersect at $q = 0$.

The monotone curves $\tau(q) \sim q$ of IM flows, P2P unidirectional video flows (PU), and MMORPG game flows (MG) are plotted in Fig. 6. These curves are significantly different from each other. The curves of SD, HD, and UD of video streaming flows are plotted in Fig. 7, and their corresponding flow sequences are shown in Fig. 3. The slope of the curve $\tau(q) \sim q$ at each point is the Holder exponent α, which represents the degree of data mutation. The minimum α_{min} is obtained when $q \to +\infty$, while the maximum α_{max} is achieved when $q \to -\infty$. In this paper, we use $\tau(q) \sim q$ curves to represent the fractal characteristics of flows for classification.

B. Calculating the Core Domain $Q|_{(-q,+q)}$

As discussed in Section IV-C, the workload increases exponentially with the increase in q. However, when q exceeds a certain level, it has no significant effect on the curve $\tau(q) \sim q$. Here, we still use the above IM, PU, and MG flows as an example. As shown in Fig. 8, the changing rate of $\Delta\tau(q)$ is gradually stabilized when $|q|$ increases from 10 to 20. The substantial increase in $|q|$ to 30 does not result in any change in $\Delta\tau(q)$. Therefore, the range of q can be reasonably narrowed, but it cannot be too small. As shown in Fig. 8, when $|q|$ is smaller than 10, the rate of $\Delta\tau(q)$ caused by Δq changes drastically, which will result in serious curve defects. Therefore, the core domain Q should be properly selected by using the WSOS method according to (27).

For IM flows, the optimal value of Q is 15. As shown in Fig. 9, when $Q = 15$, the variation coefficient of spectrum $\tau(q)$ is around 0.02, which means that the spectral difference between inside and outside the core domain is 0.02. When $Q = 10$, the variation coefficient is about 0.2, and the difference becomes more pronounced. With the continuous decrease in Q, the variation coefficient increases greatly, and therefore, the difference is more significant.

We repeatedly calculated the optimal value of Q with other classes of flows. Q varies from 13 to 16. Note that the smaller the value of Q, the lower the classification accuracy. On the

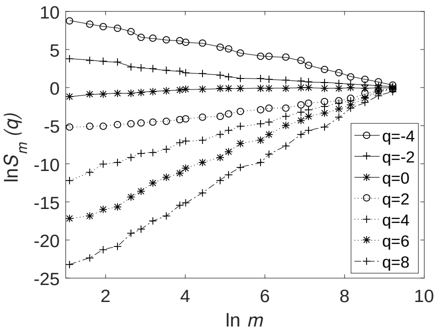

Fig. 5: The $\ln(S_m(q)) \sim ln(m)$ curves.

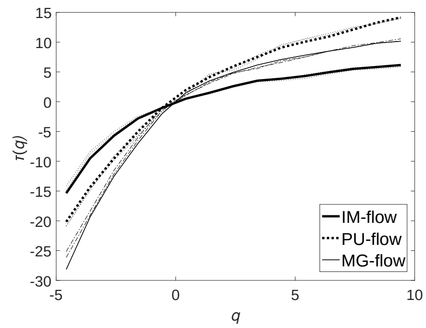

Fig. 6: The $\tau(q) \sim q$ curves.

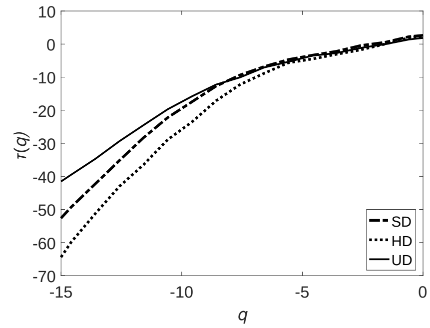

Fig. 7: The $\tau(q) \sim q$ curves of SD, HD, and UD in video streaming.

TABLE II: CLASSES OF DIFFERENT GRANULARITY

Coarse-grained classes	Medium-grained classes	Fine-grained NRQ classes	Class label	Applications
Video	Video conversation	Video Conferencing	1	Gotomeeting
		Telemedicine	2	FsMeeting
		Instant messaging videos (IM)	3	QQ, Wechat
		Group chat videos	4	Skype, MSN messenger, AOL Messenger
		Instant videos	5	eBuddy
		E-commerce	6	Direct Connect
	P2P video	Unidirectional videos (PU)	7	PPlive, PPStream
		Bidirectional videos (PB)	8	TVant, Lime Wire, Gnutella, Fast Track
		multidirectional videos,	9	Bittorrent, Edonkey, Kazaa
		BT video on demand	10	Jjvod
	Video streaming	SD (VSS)	11	
		HD (VSH)	12	Youku, Tudou, iQIYI
		UD	13	
		$\leq 480p$	14	
		720P	15	YouTube, Vimeo, etacafe, YahooVideo, Netflix
		≥ 1080	16	
	Live Video	Video broadcast	17	UUSee, LETV
		Video surveillance	18	Ezviz
Web browsing				Baidu, Blogger, Hotmail, Redbook, eCook
Audio	Audio conversation			QQ, Wechat
	P2P audio			Peergine, MusicTorrent, Xiami
	Online music			TTplayer, Kugou, QQmusic
	Audio broadcasting			GoldenRadio, Qingting
Game	Web game	Console game	19	SDO
		MMORPG (MG)	20	DotA

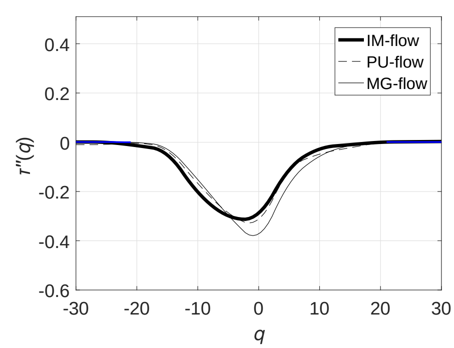

Fig. 8: Curves $\tau''(q)$ of IM, PU, and MG flows.

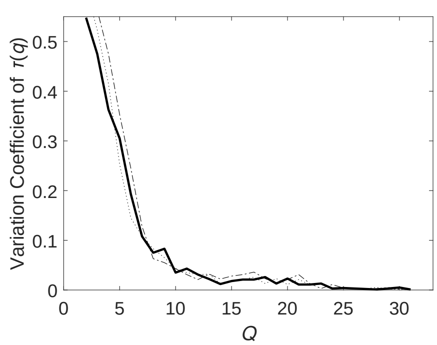

Fig. 9: Coefficient of variation for different Q values.

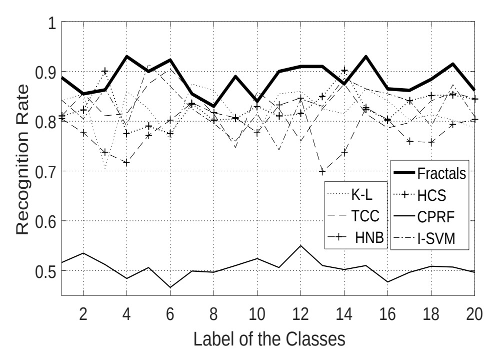

Fig. 10: Statistics of tests on the ability of responsiveness.

other hand, the larger the value of Q, the more computations it requires. Since similar results are obtained with $Q = 15$ and 16, the optimal value of Q is set to 15 to strike a balance between classification accuracy and computation complexity.

C. Effect of Segmentation

On the basis of the previous two experiments, the aim of this experiment was to compare the effects of segmentation and non-segmentation by calculating ϕ between IM and P2P unidirectional (PU) video flows. In order to simplify the calculation, we set the threshold to $T = 1.16$ and the segmentation to $S = 8$.

As shown in Table IV, in the case of non-segmentation, ϕ varies from 1.159 to 1.164. When ϕ is close to the threshold, the random variation in ϕ will lead to unstable classification. Sometimes, the classification result is Y (Yes), and the flows are identified as the same class; sometimes, the classification result is N (No), and the flows are identified as different

TABLE IV: DIFFERENCES BETWEEN SEGMENTATION AND NON-SEGMENTATION

Serial number	ϕ without segmentation	Classification result	ϕ with segmentation	Classification result
1	1.163	N	1.162	N
2	1.161	N	1.161	N
3	1.159	**Y**	1.161	N
4	1.162	N	1.161	N
5	1.164	N	1.162	N
6	1.163	N	1.162	N
7	1.162	N	1.161	N
8	1.159	**Y**	1.161	N
9	1.162	N	1.162	N
10	1.161	N	1.161	N

classes. In the case of segmentation, ϕ is relatively stable, so the classification result is also relatively stable. The above results are obtained by setting the threshold to $T = 1.16$. If the threshold is set to $T = 1.1$ or $T = 1.2$, then there is no difference between segmentation and non-segmentation.

According to the related statistics, segmentation helps to obtain stable ϕ, and more segments of flows indicate higher stability of classification. However, under given resolution N, excessive segmentation will result in deviation of spectrum estimation. According to the MMOF function (32), the optimal segmentation is $S^* = 8$ in terms of IM and PU flows in this experiment. We repeatedly calculate the optimal segmentation S^* with other classes, and find S^* varies from 8 to 9. So we set S^* at 8 in this paper, which is proved to be appropriate considering both stability of flows classification and deviation of spectrum estimation.

D. Performance of Classification

In this experiment, 3000 flows were randomly selected from the NJUPT dataset for IM flows, PU flows, MG flows, video streaming SD flows (VSS), video streaming HD flows (VSH), and P2P bidirectional videos (PB), with 500 flows for each class. Here, two questions are addressed.

(i) Why did we select 500 flows for each class? We selected 500 flows because there is a typical imbalance in our datasets; the number of flows for different classes is quite different, which is typical for most datasets (since it is impossible to guarantee the number of users of a campus network to generate a similar number of flows for each class during the data collection period). For example, in the NJUPT dataset, classes 3, 11, and 12 account for more than 80% of the flows. In the IU dataset, fewer than 900 flows were generated by the application FsMeeting. Of course, there have been many studies on such imbalanced classes. Several studies have used relatively simple resampling methods, including random undersampling and random oversampling. In this experiment, we adopted the random undersampling method; that is, we randomly selected 500 flows for each class.

(ii) Why did we not use all the classes? In this study, we aimed to classify flows at a fine-grained level, so it is important to observe changes in classification performance with the increase in L (the number of classes). For example, when a dataset contains only 6 classes, CHS achieves excellent performance with an overall accuracy of 97.35%, which is much higher than the accuracies of all the other existing methods. However, when L is increased, its classification performance declines greatly. We wanted to observe the changes in classification performance and analyze the important factors that influence the performance to predict its response when L continues to grow.

TABLE V: CONFUSION MATRIX (%)

	IM	PU	MG	VSS	VSH	PB
IM	92.35	1.45	1.36	1.32	1.47	2.05
PU	1.41	91.76	1.79	2.21	1.22	1.61
MG	1.98	1.73	90.47	1.56	2.54	1.72
VSS	1.52	2.11	1.53	91.6	1.68	1.56
VSH	1.88	1.91	2.05	1.81	90.68	1.67
PB	1.92	1.45	1.73	1.96	1.35	91.59

Therefore, in the evaluation of classification, we first tested the classification performance for L = 6, with classes 3, 7, 8, 11, 12, and 20 from the NJUPT dataset. Next, L was increased to 12, with classes 1, 2, 3, 7, 8, 11, 12, 13, 17, 18, 19, and 20 from the IU dataset. Finally, we continued to increase L to 20, with all 20 classes from the NIUI dataset, as shown in Tables II and III. We certainly could omit the experiments on 6 classes and 12 classes and show only the results of experiments on 20 classes. However, it would be hard to show the impact of L with this approach. When L increases, different methods respond in unique ways. We aim to show these changes to explore the main issues that can occur during classification and analyze the major causes.

Two-fold cross-validation was carried out on these flows. That is, the flows were randomly and equally divided into two groups: one group comprised training samples, and the other comprised testing samples. The final result was obtained by averaging the results of 20 runs, and it is presented in a confusion matrix in Table V. The rate of correctly identifying IM flows is 92.35%, and the rates of misidentifying IM flows as PU, MG, VSS, VSH, and PB are 1.45%, 1.36%, 1.32%, 1.47%, and 2.05%, respectively; the rate of correctly identifying PU flows is 91.76%, and the rates of misidentifying PU flows as IM, MG, VSS, VSH, and PB are 1.41%, 1.79%, 2.21%, 1.22%, and 1.61%, respectively.

From Table V, we can compute the *frr* of IM, PU, MG, VSS, VSH, and PB flows as 7.65%, 8.24%, 9.53%, 8.4%, 9.32%, 8.41%, respectively, and the *far* for the six types of flows is 8.71%, 8.65%, 8.46%, 8.86%, 8.26%, and 8.61%, respectively. These results are consistent with the Otsu scheme given in (37), which claims to establish a global optimization and avoid the local worst case.

E. Comparison with Benchmarks

Based on the experiment in Section VI-D, we further train and test several state-of-the-art schemes, including CHS, CPRF, I-SVM, K-L, TCC, and HNB. We provide the details of these baseline methods as follows:

In [34], Wu *et al.* proposed a novel classification structure called the chain and hierarchical structure (CHS), for fine-grained classification of network video flows. CHS combines several base classifiers to achieve a better performance and higher accuracy rate than a single classifier. Garcia *et al.* [13]

developed a method, which we call CPRF, to rapidly distinguish video flows from non-video flows using only the initial 20 packets of a flow. The authors found that some of the statistical features, such as the deviation and kurtosis of packet sizes, are highly computationally expensive to extract. Thus, composite (cp) features, which require minimal computational effort, were introduced to achieve an outstanding runtime performance, with 1 million classifications per second. I-SVM [28] is a representative supervised machine learning method (SVM). In the traditional SVM network traffic classification, all features are treated equally. Then, Hao *et al.* proposed the I-SVM method, which employs a weight-learning algorithm to assign a weight to a feature according to its importance in classification. Kim *et al.* [9] used the Kullback-Leibler divergence to measure the divergence between the Markov models of the flows. A test flow is classified as the application whose Markov model has the smallest divergence. Zhang *et al.* [24] proposed the bag of flow (BoF) concept and improved the near neighbor (NN) classifier, which can effectively improve classification performance by incorporating correlated information into the classification process. Their network traffic classification using correlation information is abbreviated to TCC. In [26], Ghofrani *et al.* applied a hidden naive Bayes (HNB) structure for traffic applications using a supervised discretization method, which is different from traditional classification methods because it assumes the independence of all features. The classification of new flows is based on the maximum likelihood of the HNB structure, and the model yields a posteriori probability for the given features.

According to [9], [13], [24], [26], [28], [34], the features adopted by the CHS, CPRF, I-SVM, K-L, TCC, and HNB schemes are summarized in the first column (feature set A) of Table VI. The two-fold cross-validation is also shown here, with 3000 flows (the flows reported in Section VI-D) as the training samples and testing flows. They were randomly selected from six classes of the NJUPT dataset, namely, IM, PU, MG, VSS, VSH, and PB. The final results were obtained by averaging the results of 20 runs.

The classification results are presented in Table VII. It can be seen that CPRF is barely able to classify the PU and PB flows. The cp features are effective for classifying video flows from non-video flows, but they fail to further classify them since most of the video flows have a similar duration, size of the largest packet, etc. The recognition accuracy rate of I-SVM for IM flows is considerably high, but it is unsatisfactory for PU flows and VSS flows. The accuracy rate of K-L does not achieve satisfactory results. In K-L, four packet patterns are defined for flows from two different applications. However, it may need more packet patterns to classify flows at the fine-grained level. The accuracy rates of HNB for the six classes are around 80%, and the accuracy rate of TCC is slightly higher than that of HNB. By contrast, the accuracy rate of the CHS scheme is as high as 94.73%. CHS combines several base classifiers and thus has better performance and a higher accuracy rate than those of a single classifier.

TABLE VI: FEATURE SET

Method	Feature set A	Feature set B
HS [34]	Number of packets Number of Bytes Size of largest packet Variance of packet size Mean packet size Variance of packet arrival time Mean packet inter arrival time	Number of packets Number of Bytes Minimum of packet size Mean packet size Variance of packet size SD of packet sizes Variance of arrival time Mean arrival time SD of arrival time
CPRF [13]	Number of Bytes Number of packets Mean packet size Flow duration Number of non full packets Mean packet inter arrival time Size of largest packet Fraction of packets	Number of Bytes Number of packets Mean packet size Variance of packet size Skew of packet size Kurtosis of packet sizes Flow duration SD of packet sizes
I-SVM [28]	Maximum of packet size Minimum of packet size Minimum segment size Maximum segment size Total number of bytes	Maximum of packet size Minimum of packet size Minimum segment size Segment size Total number of bytes Port number Number of SC packets Number of CS packets Average segment size
K-L [9]	Packet patterns Size of the first four packets Direction of flows	Packet patterns Size of the first four packets Direction of flows Average packet size Variance of packet size Transmission rate Packet number Port number
TCC [24]	Average packet size Variance of packet size Transmission rate Packet number Average packet interval	Packet size Variance of packet size Transmission rate Packet number Average packet interval Maximum of packet size Minimum of packet size Transmission rate
HNB [26]	Number of CS packets Maximum CS packet size Minimum packet size Variance of packet size Number of SC packets Maximum SC packet size	Number of CS Packets Maximum CS packet size Packet size Variance of packet size Average packet size Variance of packet size Transmission rate Packet number Average packet interval

TABLE VII: COMPARISON OF RECOGNITION RATES WITH FOUR BENCHMARKS (%)

	IM	PU	MG	VSS	VSH	PB
Fractals	92.35	**91.76**	90.47	91.6	90.68	91.59
CHS [34]	**97.08**	90.33	**95.82**	**98.09**	**97.95**	89.12
CPRF [13]	91.28	45.87	79.61	67.83	71.54	47.86
I-SVM [28]	94.61	78.49	85.33	79.84	96.4	**98.68**
K-L [9]	83.82	76.44	93.76	84.47	82.28	88.35
TCC [24]	86.74	90.68	82.89	**93.64**	85.63	89.43
HNB [26]	79.28	83.63	86.24	77.35	81.65	83.52

F. Adaptability to Dynamic Flows

In order to check whether these schemes can adapt to varying classes, we randomly selected 12 classes of new

TABLE VIII: AVERAGE RECOGNITION RATES WITH DIFFERENT DATA AND FEATURE SETS (%)

	6 lasses Feature set A	12 classes 12 Feature set A	12 classes Feature set B
Fractals	91.41	**90.16**	**90.16**
CHS [34]	**94.73**	84.39	88.71
CPRF [13]	67.33	51.43	85.34
I-SVM [28]	88.89	84.61	87.81
K-L [9]	84.85	81.83	83.46
TCC [24]	87.91	83.29	87.08
HNB [26]	81.94	77.52	80.27

flows (500 flows for each class) from the IU dataset. The 12 classes were video conferencing flows, telemedicine flows, instant messaging video flows, video streaming SD, video streaming HD, video streaming UD, P2P unidirectional videos, P2P bidirectional videos, video broadcasts, video surveillance, console games, and MMORPG games. Their labels correspond to 1, 2, 3, 7, 8, 11, 12, 13, 17, 18, 19, 20, respectively, as shown in Table III.

As shown in Table VIII, when the number of classes increases from 6 to 12, the average accuracy rates of CHS, CPRF, K-L, I-SVM, HNB, and TCC markedly decrease. The accuracy rate of the CPRF scheme declines considerably. When the number of classes increases, the differences between classes decrease. Therefore, cp features (flow duration, size of largest packet, etc.) are almost unable to classify video flows. When we replaced these cp features with feature set B, the average accuracy rate rose to 85.34%. It can be seen that the selected features have a significant impact on classification performance. In general, statistical features, which can be used to effectively identify the previous set of flow samples, do not work well for the new set. Therefore, feature set A should be updated to regain ideal classification performance.

In contrast, for Fractals, the average accuracy rate of 12 classes is close to that of 6 classes. Fractal characteristics are different from statistical features in that they reflect the essence of things (as a fingerprint or iris) that do not change when the classes of flows increase or other variables of external environments change. As shown in Table IX, our proposed scheme consistently achieves a higher accuracy rate for all 12 classes of video flows and shows stable classification performance.

We continued to increase the number of classes from 12 to 20. In Fig. 10, the x-axis represents the label of 20 classes, and the y-axis represents the recognition accuracy rate. The accuracy rates of Fractals are consistently higher than that of other methods, and the average accuracy rate reaches around 90%. In addition, the Fractals method shows stable classification performance for all classes, while the CHS, CPRF, K-L, I-SVM, HNB, and TCC methods exhibit poor performance.

The CPRF method, in particular, lost the ability to classify video flows. When the number of classes increases to 20, the differences between classes are further decreased. Thus, the cp features of CPRF do not work at all. In addition, some issues remain with CHS. In particular, when the number of classes increases, its recognition accuracy rate declines greatly. We analyzed the reasons and found that CHS has a chain effect of error propagation. Specifically, the error in classifier 1 can spread backward to classifiers 2, 3, and 4. Similarly, the error in classifier 2 can spread backward to classifiers 3 and 4. When the number of classes increases, the number of classifiers also increases; thus, the accumulative error for each classifier increases considerably. As a result, for a large dataset with more classes, CHS cannot provide satisfactory performance.

In contrast to the state-of-the-art schemes, our scheme based on fractal characteristics does not require application signatures or statistical features, so it can achieve better performance in response to increased classes.

G. Computational and Space Complexity

For real-time applications, the classification of video flows should ensure not only high recognition accuracy but also low time and space complexity. Time complexity involves the learning time, storage time, and classification time. Compared with the supervised methods (CHS, I-SVM, and HNB), CPRF, K-L, TCC, and Fractals have no additional learning procedure. Furthermore, the same set of flows was used for all methods, so they have almost the same storage time. Thus, the performance of time complexity is differentiated mainly on the basis of the classification time.

In accordance with international practice, we used the special length of flows to compute the classification time. The baseline methods CHS, CPRF, I-SVM, K-L, TCC, and HNB all have different requirements for the duration of flows. For example, CPRF was designed to only employ the 20 initial packets to extract the features, and would hence require on average less than 1 second of flow duration. However, for Fractals, the duration of the flows was set to 100 s, as described in Section VI-A. If the flow is shorter than that, then we cannot obtain enough data to compute the fractal characteristics. Hence, the duration of the flows was set to 3 minutes. As a result of such restrictions on duration, Fractals cannot be applied to classify certain flows, such as the Web browsing data in Table II. Our scheme only shows significant superiority for the fine-grained classification of video flows.

In this experiment, 100 flows were randomly selected from the NJUPT (6 classes), IU (12 classes), and NIUI (20 classes) datasets to evaluate the classification time. As shown in Fig. 11, the Fractals method took 1.851 s for the NJUPT dataset, 1.88 s for the IU dataset, and 1.924 s for the NIUI dataset.

Table X presents a theoretical analysis of time and space complexities. From Fig. 11 and Table X, it can be seen that the time and space complexities of CPRF are extremely low. In CPRF, the random forest is established to classify video flows according to cp features. Therefore, the time and space complexities of CPRF are $O(M_1LJ)$ and $O(M_1 + LJN_1)$. In this experiment, the testing time of CPRF did not achieve the results reported in [13] for two main reasons. (i) Data preprocessing was not taken into account in [13]. Garcia *et al.* classified flows by using cp features and measured only the classification time, not considering the time required to obtain

TABLE IX: CONFUSION MATRIX (%)

labels	1	2	3	7	8	11	12	13	18	19	20
1	91.21	1.27	1.21	1.43	0.75	0.94	0.67	0.65	0.81	0.35	0.27
2	2.73	88.74	2.16	0.91	0.6	0.61	0.72	0.83	0.91	0.51	0.56
3	1.71	2.23	89.68	0.76	0.87	0.4	0.62	0.67	0.82	0.74	0.69
7	0.84	1.31	1.14	87.82	2.15	1.52	0.89	0.94	0.71	0.75	1.11
8	1.02	0.54	0.83	2.81	88.31	1.35	0.77	1.01	1.05	0.74	0.91
11	0.63	0.41	0.53	0.91	0.37	91.56	1.81	1.28	0.83	0.67	0.45
12	0.41	0.61	0.55	0.91	0.63	2.17	90.22	1.64	0.69	0.76	0.54
13	0.69	0.84	0.75	0.52	0.58	1.93	2.37	89.15	0.73	0.91	0.75
18	0.44	0.75	0.52	0.69	0.71	0.93	0.64	0.56	91.14	2.22	0.48
19	0.31	0.37	0.39	0.61	0.65	0.46	0.82	0.47	2.13	92.72	0.58
20	0.52	0.51	0.63	0.57	0.81	0.43	0.75	0.89	0.46	0.57	91.73

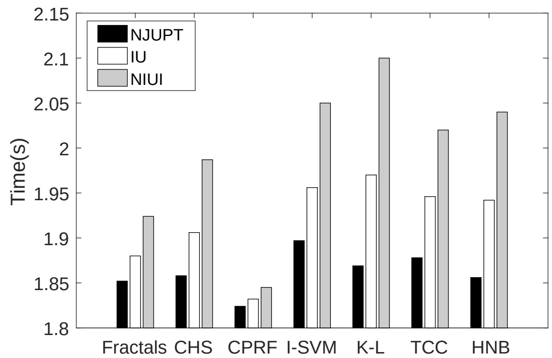

Fig. 11: Comparison of classification time.

TABLE X: COMPARISON OF TIME AND SPACE COMPLEXITY

	Time complexity	Space complexity
Fractals	$O(M_1 L N_0 \log(N_0))$	$O((M_1 + L) N_0 \log(N_0))$
CHS [34]	$O(M_1 \log(M_0) L J^2 N_1^2)$	$O((M_0 + M_1) L J N_1)$
CPRF [13]	$O(M_1 L J N_1)$	$O(M_1 + L J N_1)$
I-SVM [28]	$O(M_1 L (J N_1^2 + M_0 J N_1))$	$O((M_0 + M_1) J N_1^2)$
K-L [9]	$O(M_1 L J^3 N_1)$	$O(M_1 L J^2 N_1)$
TCC [24]	$O(M_0 M_1 L J N_1^2)$	$O((M_0 + M_1) J N_1)$
HNB [26]	$O(M_1 (M_0 J^2 + L (J N_1)^2))$	$O(M_1 L (J N_1)^2)$
Parameters	M_0: no. of sample flows N_0: resolution of fractals J: no. of features	M_1: no. of testing flows N_1: no. of feature values L: no. of classes

these cp features. (ii) The evaluation in [13] considered only the first 20 packets in a flow, while the present evaluation was performed using up to 60 seconds of flow data. For the I-SVM method, Hao *et al.* [28] proposed a weight-learning algorithm to assign each feature a weight. Therefore, compared with the traditional SVM method, I-SVM requires additional time to calculate the weights, as shown in Table X. For K-L [9], the multiplication of three transition probability matrices of the Markov models (the matrix order is J) results in high time and space complexities. The TCC method [24] adopts the nearest neighbor rule, which requires the storage for all training data samples, and the space complexity is $O((M_0 + M_1)JN_1)$. It compares all flows in one KNN classifier, which results in $O(M_0 M_1)$ comparisons. CHS combines several KNN classifiers to implement classification. It divides the sample flows into several KNN classifiers, thus reducing the number of comparisons to $M_1 \log(M_0)$. Ghofrani *et al.* [26] proposed a structure of HNB to achieve classification, and it must consider each pair of parent and child features within each class. Therefore, the time and space complexities of HNB are sensitive to J^2 (more details can be found in [26]).

According to the previous experiments, the parameters M_0, M_1, and N_0 were fixed. As the video classes (L) increase from 6 to 12 and 20, J increases as a result. Here, we only focus on these variable parameters. As shown in Table X, the time and space complexities of Fractals depend only on L, while those of the other methods depend not only on L but also on other factors, such as J and N_1.

In general, our proposed method Fractals relies neither on application signatures (obtained by inspecting the payload content) nor on statistical features (extracted from given flow samples through a long-term statistical analysis), and as a result, it has superior performance in the classification of flows at a fine-grained level.

VII. Conclusions

In this paper, we investigate the classification of Internet video traffic at the fine-grained level. To mitigate the limitations of existing techniques based on application signatures and statistical features, we introduce the fractal characteristics of flows as a new concept and propose the use of unique characteristics for accurate classification. We first prove the fractal characteristics of flows through rigorous analysis, and we then present a theoretical classification framework for the proposed scheme on the basis of multifractal theory.

In our Fractals method, fractal characteristics can be obtained at any stage of the flows, which are quite different from statistical features. Moreover, our method does not require payload inspection, and thus, it can be used to process encrypted flows. It also avoids the time-consuming process of feature extraction and shows robustness to varying classes. In general, the proposed scheme demonstrates superior performance for the fine-grained classification of video traffic.

However, there are some issues that need to be further explained and explored in the future.

(i) We will propose a complete framework for classification. We found that coarse-grained classification methods (e.g., CPRF) show obvious superiority for coarse-grained classes, but they do not work well for fine-grained classes, as shown in Section VI. Although our proposed Fractals scheme shows excellent performance for fine-grained video classes, it does not work for certain coarse-grained class (e.g., Web browsing data). In order to address the above shortcoming, we propose a complete framework for classification. In this framework, a coarse-grained classification method is used to classify the flows into text flows, voice flows, and video flows, among other types. Next, by using the fine-grained classification method, video flows are further classified into categories such as video conference, telemedicine system, and electronic commerce.

(ii) In this paper, the estimated spectrum $\tau(q)$ is used to represent the fractal characteristics of flows. In our future research, other estimation spectra will be explored to further improve accuracy and reduce complexity.

References

[1] P. Tang, Y. Dong, Z. Wang, and L. Yang, "Classification of Internet video traffic using multi-fractals," in *Proc. IEEE Int. Symp. Commun. Inf. Technol. (ISCIT)* , Cairns, Australia, Sept. 2017, pp. 1–5.

[2] C. Long, C. Yang, and J. Tao, "Edge computing framework for co-operative video processing in multimedia IoT systems," *IEEE Trans. Multimedia,* vol.20, no.5, pp. 1126–1139, Mar. 2018.

[3] M. Li, H. L. Chen, "Energy-efficient traffic regulation and scheduling for video streaming services over LTE-A networks," *IEEE Trans. Mobile Comput.,* vol.18, no.2, pp. 334–347, Feb. 2019.

[4] Y. Liu, N. Di, and B. Li, "Delay-optimized video traffic routing in software-defined interdatacenter networks," *IEEE Trans. Multimedia,* vol.18, no.5, pp. 865–878, May 2016.

[5] A. O. Al-Abbasi, V. Aggarwal, M. R.Ra, "Multi-tier caching analysis in CDN-based over-the-top video streaming systems," *IEEE/ACM Trans. Netw.,* vol.20, no.1, pp. 1–13, Feb. 2019.

[6] G. Gao, H. Zhang, and H. Hu, "Optimizing quality of experience for adaptive bitrate streaming via viewer Interest inference," *IEEE Trans. Multimedia,* vol.20, no.12, pp. 3399–3413, Dec. 2018.

[7] Z. Wang, S. Mao, L. Yang, and P. Tang, "A survey of multimedia big data," *IEEE/CIC China Commun.*, vol.15, no.1, pp. 155–176, Jan. 2018.

[8] J. Dong, X. Li, and C. Snoek, "Predicting visual features from text for image and video caption retrieval," *IEEE Trans. Multimedia,* vol.20, no.12, pp. 3377–3388, Dec. 2018.

[9] J. Kim, J. Hwang, and K. Kim, "High-performance Internet traffic classification using a Markov model and Kullback-Leibler Divergence," *Mobile Inf. Syst.,* vol.2016, Article ID 6180527, pp. 1–13, Jan. 2015.

[10] N. Carlsson, D. Eager, and V. Krishnamoorthi, "Optimized adaptive streaming of multi-video stream bundles," *IEEE Trans. Multimedia,* vol.19, no.7, pp. 1637–1653, Jul. 2017.

[11] Q. Liu, Chen, Chang Wen, "Smart downlink scheduling for multimedia streaming over LTE networks with hard hand-off," *IEEE Trans. Circuits Syst. Video Technol.,* vol.25, no.11, pp. 1–14, Nov. 2015.

[12] F. R. M Lima, T. F. Maciel, and W.C. Freitas, "Resource assignment for rate maximization with QoS guarantees in multiservice wireless systems," *IEEE Trans. Veh. Technol.,* vol.61, no.3, pp. 1318–1332, Mar. 2016.

[13] J. Garcia, T. Korhonen, R. Andersson, and F. Västlund, "Towards video flow classification at a million encrypted flows per second," *Proc. IEEE Int. Conf. Inform. Netw. Appl. (AINA),* Cracow, Poland, May. 2018, pp. 358–365.

[14] W. Abbessi, H.Nabli, "General approach for video traffic: from modeling to optimization," *Multimedia Systems,* vol.25, no.3, pp. 177–193, Jun. 2018.

[15] A. Canovas, J. M. Jimenez, O. Romero, and J. Lloret, "Multimedia data flow traffic classification using intelligent models based on traffic patterns," *IEEE Network,* vol.32, no.6, pp. 100–107, Nov. 2018.

[16] L. Yang, Y. Dong, and M. S.Rana, "Fine-grained video traffic classification based on QoE values," *Wireless Personal Communications,* vol.103, no.4, pp. 1–18, May. 2018.

[17] S. Petrangeli, J. V. D. Hooft, and T. Wauters, "Quality of experience-centric management of adaptive video streaming: services status and challenges," *Acm Trans. Multimedia Comput. Commun. Applicat.,* vol.14, no.2s, pp. 1–29, May. 2018.

[18] T. Kimura, M. Yokota, and A. Matsumoto, "QUVE: QoE maximizing framework for video-streaming," in *IEEE J. Sel. Top. Sign. Proces.,* vol.11, no.1, pp. 138C153, Mar. 2017.

[19] A. Get'man, Y. V. Markin, and E. F. Evstropov, "A survey of problems and solution methods in network traffic classification," *Proc. Isp. Ras,* vol.29, no.3, pp. 117–150, Jan. 2017.

[20] K. S. Shim, J. H. Ham, and B. D. Sija, "Application traffic classification using payload size sequence signature," *International Journal of Network Management,* vol.27, no.5, e1981, Jun. 2017.

[21] X. Yun, Y. Wang, Y. Zhang, and Y. Zhou, "A semantics-aware approach to the automated network protocol identification," *IEEE/ACM Trans. Netw.,* vol.24, no.1, pp. 583–595, Feb. 2016.

[22] J. Garcia, and A. Brunstrom, "Clustering-based separation of media transfers in DPI-classified cellular video and VoIP trafficin," in *Proc. IEEE Int. Conf. Commun. Netw. (WCNC),* Barcelona, Spain, Apr. 2018, pp. 1-6.

[23] A. Tejero-De-Pablos, Y. Nakashima, and T. Sato, "Summarization of user-generated sports video by using deep action recognition features," *IEEE Trans. Multimedia,* vol.20, no.8, pp. 2000–2011, Aug. 2018.

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

[25] J. Kornycky, O. Abdul-Hameed, A. Kondoz, "Radio frequency traffic classification over WLAN," *IEEE Trans. Parallel Distrib. Syst.,* vol.25, no.1, pp. 56–68, Feb. 2017.

[26] F. Ghofrani, A. Jamshidi, and A. Keshavarz-Haddad, "Internet traffic classification using Hidden Naive Bayes model," in *Proc. IEEE Int. Conf. Electr. Eng. (ICEE),* Tehran, Iran, May 2015, pp. 235–240.

[27] R. Nossenson and S. Polacheck, "On-line flows classification of video streaming applications," in *Proc. IEEE Int. Symp. Network Comput. Appl. (NCA),* Cambridge, MA, Sept. 2015, pp. 251–258.

[28] S. Hao, J. Hu, and S. Liu, "Improved SVM method for internet traffic classification based on feature weight learning," in *Proc. Int. Conf. Control, Automat. Inform. Sci. (ICCAIS),* Changshu, China, Oct. 2015, pp. 102–106.

[29] C. Thay, V. Visoottiviseth, and S. Mongkolluksamee, "P2P traffic classification for residential network," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. (ICSEC),* Chiang Mai, Thailand, Feb. 2016, pp. 23–26.

[30] T. Qin, L. Wang, and Z. Liu, "Robust application identification methods for P2P and VoIP traffic classification in backbone networks," *Knowledge-Based Syst.,* vol.82, no.C, pp. 63–97, Jul. 2015.

[31] X. Cheng, J. Liu, and C. Dale, "Understanding the characteristics of Internet short video sharing: A YouTube-Based Measurement Study," *IEEE Trans. Multimedia* vol.15, no.5, pp. 1184–1194, Jun. 2013.

[32] K. Takeshita, T. Kurosawa, and M. Tsujino, "Evaluation of HTTP video classification method using flow group information," in *Proc. IEEE Int. Conf. Telecommun. Network Strategy and Planning Symp.,* Warsaw, Poland, Sep. 2010, pp. 1–6.

[33] L. M. Nair and G. P. Sajeev, "Internet traffic classification by aggregating correlated Decision Tree classifier," in *Proc. IEEE Int. Conf. Comput. Intell. (ICACI),* Kuantan, Malaysia, Jul. 2015, pp. 119–126.

[34] Z. Wu, Y. Dong, L. Yang, and P. Tang, "A new structure for Internet video traffic classification using machine learning," *in Proc. IEEE Int. Conf. Adv. Cloud Big Data (CBD),* Lanzhou, China, Aug. 2018, pp. 322–327.

[35] Y. Fu, H. Xiong, and X. Lu, "Service usage classification with encrypted Internet traffic in mobile messaging Apps," *IEEE Trans. Mobile Comput.,* vol.15, no.5, pp. 1–11, Jan. 2016.

[36] A. Dainotti, A. Pescap, and K. C. Claffy, "Issues and future directions in traffic classification," *IEEE/ACM Trans. Netw.,* vol.26, no.1, pp. 35–40, Jan. 2012.

[37] J. Zhang, C. Chen, and Y. Xiang, "Internet traffic classification by aggregating correlated naive bayes predictions," *IEEE Trans. Inf. Forensics Security,* vol.8, no.1, pp. 5–15, Jan. 2013.

[38] S. Pratiher, S. Patra, and P. Bhattacharya, "On the marriage of Kolmogorov complexity and multi-fractal parameters for epileptic seizure classification," in *Proc. IEEE Int. Conf. Contemporary Comput. Informat. (IC3I),* Noida, India, May 2017, pp. 831–836.

[39] L. Livi, A. Sadeghian, and H. Sadeghian, "Discrimination and characterization of Parkinsonian rest tremors by analyzing long-term correlations and multifractal signatures," *IEEE Trans. Biomed. Eng.,* vol.63, no.11, pp. 2243–2249, Nov. 2016.

[40] I. Hernández-Carrasco, V. Garcon, and J. Sudre, "Increasing the resolution of ocean pCO2 maps in the south eastern Atlantic ocean merging multifractal satellite-derived ocean variables," *IEEE Trans. Geosci. Remote.,* vol.56, no.11, pp. 2243–2249, Nov. 2018.

[41] N. M. G. Al-Saidi and H. Y. Abdul-Wahed, "Classification of remote sensing images via fractal discriptores," in *Proc. IEEE Int. Conf. Advance of Sustainable Eng. Appl. (ICASEA),* Wasit, Iraq, Mar. 2018, pp. 99–104.

[42] E. Akar, S. Kara, and H. Akdemir, "Fractal analysis of MR images in patients with chiari malformation: The importance of preprocessing," *Biomedical Signal Processing & Control,* vol.2017, no.31, pp. 63–70, Jan. 2017.

[43] A. Allwright, A. Atangana, "Fractal advection-dispersion equation for groundwater transport in fractured aquifers with self-similarities," *The European Physical Journal Plus,* vol.133, no.2, pp. 48–54, Feb. 2018.

[44] F. G. M. Neto, I. C. De Paula, "Fractal dimension-based drosophila melanogaster wing genotype classification," in *Proc. IEEE Brazilian Conf. Intell. Syst. (BRACIS),* Uberlandia, MG, Brazil, Oct. 2017, pp. 258–263.
[45] A. Ko, G. Mascaro, and E. R. Vivoni, "Irrigation impacts on scaling properties of soil moisture and the calibration of a multifractal downscaling model," *IEEE Trans. Geosci. Remote.,* vol.54, no.6, pp. 3128–3142, Nov. 2016.
[46] Z. Wu, L. Zhang, and Y. Meng, "Low-rate DoS attacks detection based on network multifractal," *IEEE Trans. Depend. Secure.,* vol.13, no.5, pp. 559–567, Feb. 2016.
[47] W. E. Leland, M. S. Taqqu, and W. Willinger, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Trans. Netw.,* vol.2, no.1, pp. 1–15, Feb. 1994.
[48] C. Barakat, P. Thiran, and G. Iannaccone, Modeling Internet backbone traffic at the flow level, *IEEE Trans. Sig. Process.,*, vol.51, no.8, pp. 2111–2124, Aug. 2003.
[49] R.H. Riedi, M.S. Crouse, and V.J. Ribeiro, "A multifractal wavelet model with application to network traffic," *IEEE Trans. Inf. Theory.,* vol.45, no.3, pp. 992–1018, Apr. 1999.
[50] P. Chen, "Generalization of Gärtner-Ellis Theorem," *IEEE Trans. inform. Theory,* vol.46, no.7, pp. 2752–2760, Nov. 2000.
[51] X. Jia, H. An, W. Fang, X. Sun, and X. Huang, "How do correlations of crude oil prices co-move? A grey correlation-based wavelet perspective," *Elsevier Energy Economics*, vol. 49, pp. 588–598, May 2015.
[52] N. Otsu, "A threshold selection method from gray-Level histogram," *IEEE Trans. Syst. Man, Cybern. B, Cybern,* vol.9, no.1, pp. 94–98, Jan. 1979.
[53] S. Kuzuoka and S. Watanabe, "A dichotomy of functions in distributed coding: An information spectral approach," *IEEE Trans. Inf. Theory*, vol.61, no.9, pp. 5028–5041, Sept. 2015.
[54] "UNB ISCX VPN-nonVPN traffic dataset," [online] Available: http://www.unb.ca/cic/research/datasets/vpn.html, as of Apr. 2016.
[55] "UNIBS-2009 payload traces," [online] Available: http://netweb.ing.unibs.it/ ntw/tools/traces/index.php, as of Sept. 2009.

APPENDIX

Define two important functions:

$$X = N_1(\alpha) \propto \varepsilon^{-f_{G_1}(\alpha)}, \tag{38}$$

$$Y = N_2(\alpha) \propto \varepsilon^{-f_{G_2}(\alpha)}, \tag{39}$$

where $N_1(\alpha)$ and $N_2(\alpha)$ are > 0 (see Eq. (4)).

Proposition 1. *If $Z = X + Y$, then the boundaries of Z is determined by $f_{G_1}(\alpha)$ and $f_{G_2}(\alpha)$.*

Proof: According to (38) and (39), we can obtain:

$$f_{G_1}(\alpha) = \lim_{\varepsilon\to 0} \frac{\ln N_1(\alpha)}{\ln \varepsilon}, \tag{40}$$

$$f_{G_2}(\alpha) = \lim_{\varepsilon\to 0} \frac{\ln N_1(\alpha)}{\ln \varepsilon}. \tag{41}$$

Note that $Z = X + Y = N_1(\alpha) + N_1(\alpha)$, and thus we define a new function $f_{G_Z}(\alpha)$ as:

$$f_{G_Z}(\alpha) = \lim_{\varepsilon\to 0} \frac{\ln(N_1(\alpha) + N_2(\alpha))}{\ln \varepsilon}. \tag{42}$$

It yields,

$$\begin{aligned}
\inf(f_{G_Z}(\alpha)) &= \lim_{\varepsilon\to 0} \frac{\ln \sqrt{2N_1(\alpha)N_2(\alpha)}}{\ln \varepsilon} \\
&= \frac{1}{2}\left(\lim_{\varepsilon\to 0} \frac{\ln 2}{\ln \varepsilon} + \lim_{\varepsilon\to 0} \frac{\ln\left(N_1(\alpha)N_2(\alpha)\right)}{\ln \varepsilon}\right) \\
&= \frac{1}{2}\left(\lim_{\varepsilon\to 0} \frac{\ln N_1(\alpha)}{\ln \varepsilon} + \lim_{\varepsilon\to 0} \frac{\ln N_2(\alpha)}{\ln \varepsilon}\right) \\
&= \frac{1}{2}(f_{G1}(\alpha) + f_{G2}(\alpha)). \\
\sup(f_{G_Z}(\alpha)) &= \lim_{\varepsilon\to 0} \frac{\ln(2\max(N_1(\alpha), N_2(\alpha)))}{\ln \varepsilon} \\
&= \max(f_{G1}(\alpha), f_{G2}(\alpha)).
\end{aligned}$$

■

Proposition 2. *If the derivatives of $f_{G_1}(\alpha)$, $f_{G_2}(\alpha)$ are $f'_{G_1}(\alpha)$, and $f'_{G_2}(\alpha)$, respectively, then $f'_{G_Z}(\alpha)$ is determined by $f'_{G_1}(\alpha)$ and $f'_{G_2}(\alpha)$.*

Proof: Based on the known conditions described above, we have:

$$\begin{aligned}
&f'_{G_Z}(\alpha) = \lim_{\Delta\alpha\to 0} \frac{1}{\Delta\alpha}(f_{G_Z}(\alpha+\Delta\alpha) - f_{G_Z}(\alpha)) \\
&= \lim_{\Delta\alpha\to 0} \frac{1}{\Delta\alpha} \lim_{\varepsilon\to 0} \left\{ \frac{\ln(N_1(\alpha+\Delta\alpha) + N_2(\alpha+\Delta\alpha))}{\ln \varepsilon} \right. \\
&\quad \left. - \frac{\ln(N_1(\alpha) + N_2(\alpha))}{\ln \varepsilon} \right\} \\
&= \lim_{\varepsilon\to 0} \frac{1}{\ln \varepsilon} \ln\left(\lim_{\Delta\alpha\to 0} \frac{1}{\Delta\alpha} \frac{N_1(\alpha+\Delta\alpha) + N_2(\alpha+\Delta\alpha)}{N_1(\alpha) + N_2(\alpha)} \right) \\
&= \lim_{\varepsilon\to 0} \frac{1}{\ln \varepsilon} \ln\left(\frac{N'_1(\alpha) + N'_2(\alpha)}{N_1(\alpha) + N_2(\alpha)} + \lim_{\Delta\alpha\to 0} \frac{1}{\Delta\alpha} \right). \\
&= \lim_{\varepsilon\to 0} \frac{1}{\ln \varepsilon} \ln\left(\frac{N'_1(\alpha) + N'_2(\alpha)}{N_1(\alpha) + N_2(\alpha)} + \lim_{\varepsilon\to 0} \frac{N'(\alpha)}{\ln \varepsilon f'_G(\alpha) N(\alpha)} \right) \\
&= \lim_{\varepsilon\to 0} \frac{\ln\left(\frac{N'_1(\alpha)+N'_2(\alpha)}{N_1(\alpha)+N_2(\alpha)}\right)}{\ln \varepsilon} + \lim_{\varepsilon\to 0} \frac{1}{\ln \varepsilon} \ln\left(1 + \frac{1}{\ln \varepsilon}(f'_G(\alpha))^{-1}\right) \\
&= \lim_{\varepsilon\to 0} \frac{1}{\ln \varepsilon} \ln\left\{ \frac{f'_{G_1}(\alpha)\varepsilon^{-f_{G_1}(\alpha)} + f'_{G_2}(\alpha)\varepsilon^{-f_{G_2}(\alpha)}}{\varepsilon^{-f_{G_1}(\alpha)} + \varepsilon^{-f_{G_2}(\alpha)}} \right\} \\
&= \frac{f'_{G1}(\alpha) + f'_{G2}(\alpha)}{f_{G1}(\alpha) + f_{G2}(\alpha)} \lim_{\varepsilon\to 0} \frac{f_{G1}(\alpha)\varepsilon^{-f_{G1}(\alpha)} + f_{G2}(\alpha)\varepsilon^{-f_{G2}(\alpha)}}{\varepsilon^{-f_{G1}(\alpha)} + \varepsilon^{-f_{G2}(\alpha)}} \\
&= \cdots \left\{ f_{G_1}(\alpha) + f_{G_2}(\alpha) - \lim_{\varepsilon\to 0} \frac{f_{G_1}(\alpha)f_{G_2}(\alpha)(\varepsilon^{-f_{G_1}(\alpha)} + \varepsilon^{-f_{G_2}(\alpha)})}{f_{G_1}(\alpha)\varepsilon^{-f_{G_1}(\alpha)} + f_{G_2}(\alpha)\varepsilon^{-f_{G_2}(\alpha)}} \right\} \\
&= \frac{f'_{G_1}(\alpha) + f'_{G_2}(\alpha)}{f_{G_1}(\alpha) + f_{G_2}(\alpha)} \max\left(f_{G_1}(\alpha), f_{G_2}(\alpha)\right).
\end{aligned}$$

■

See the last line in the proof of Proposition 2, where a special limit is used. Now we prove it as follows.

Proposition 3.

$$a + b - \lim_{e\to 0} \frac{ab(e^{-a} + e^{-b})}{ae^{-a} + be^{-b}} = \max(a, b).$$

Proof:

$$\begin{aligned}
&a + b - \lim_{e\to 0} \frac{ab(e^{-a} + e^{-b})}{ae^{-a} + be^{-b}} \\
&= \lim_{e\to 0} \frac{(a+b)(ae^{-a} + be^{-b}) - ab(e^{-a} + e^{-b})}{ae^{-a} + be^{-b}} \\
&= \lim_{e\to 0} \frac{a^2e^{-a} + b^2e^{-b}}{ae^{-a} + be^{-b}}.
\end{aligned}$$

Here, if $a > b$, then,

$$\begin{aligned}
&a + b - \lim_{e\to 0} \frac{ab(e^{-a} + e^{-b})}{ae^{-a} + be^{-b}} \\
&= \lim_{e\to 0} \frac{a^2 + b^2e^{-b+a}}{a + be^{-b+a}} = \frac{a^2}{a} = a.
\end{aligned}$$

If $a < b$, then,

$$\begin{aligned}
&a + b - \lim_{e\to 0} \frac{ab(e^{-a} + e^{-b})}{ae^{-a} + be^{-b}} \\
&= \lim_{e\to 0} \frac{a^2e^{-a+b} + b^2}{ae^{-a+b} + b} = \frac{b^2}{b} = b.
\end{aligned}$$

Therefore,

$$a + b - \lim_{e \to 0} \frac{ab(e^{-a} + e^{-b})}{ae^{-a} + be^{-b}} = \max(a, b).$$

■