Article:

Tang, P., Dong, Y., Mao, S. et al. (2 more authors) (2023) Online classification of network traffic based on granular computing. IEEE Transactions on Systems, Man, and Cybernetics: Systems. ISSN 2168-2216

https://doi.org/10.1109/TSMC.2023.3259543

Final accepted manuscript; accepted for publication on 10 March 2023

Online Classification of Network Traffic Based on Granular Computing

Pingping Tang, *Member, IEEE*, Yuning Dong, *Member, IEEE*, Shiwen Mao, *Fellow, IEEE*, Hua-Liang Wei, and Jiong Jin, *Member, IEEE,*

***Abstract*— At present, it is still a great challenge to achieve online classification of traffic flows due to the highly varying network environments, e.g., unpredictable new traffic classes, network noise and congestion. Traditional classification methods work well in stable network environments, but may not exhibit their performance in dynamic environments. To address online classification issues, a granular computing based classification model (GCCM) is developed, where the spatial and temporal flow granules are defined to make GCCM robust against variations and less sensitive to noise, and the correlations among flow granules are explored to establish the granular relation matrix (GRM). The inherent burst features between packets indicated by GRM prompt GCCM to achieve fine classification in unstable network environments. GCCM analyzes the burst features of packets without inspecting the payload information, and thus can be used to classify encrypted traffic as well as unencrypted traffic at a fast speed. In addition, GCCM model, depending on difference measurement $D(\cdot)$, is a threshold based classification, and therefore can be used to distinguish between time-varying classes. The validity of GCCM for online traffic classification is examined through theoretical results. The experimental evaluation of classification for fine and varied classes under dynamic network environments with noise and congestion also demonstrates its superiority in terms of classification accuracy and real-time performance with the state-of-the-art.**

***Index Terms*—Granular computing, granular relation matrix, network noise, online classification, traffic flows.**

I. Introduction

Network traffic is growing rapidly on a tremendous scale, with so much variety that it is indispensable to develop an effective classification systems to implement network resources management [1], provide technical support to guarantee quality of service [2], enforce differentiated services for users [3], maintain and improve the network security [4], etc. According to the 6G white paper [5], one of the first important network operations is traffic classification. Online traffic classification is necessary and has become a research focus in the fields of communications and networking [6], [7]. In recent years, flow features (e.g., duration and mean packet size) are widely used to distinguish between different traffic [8], [9]. However, the rapid development of the Internet technologies poses new challenges to online traffic classification: i) With continuous innovation of applications, new types of traffic are fed into the Internet [10]. When the number of classes is increased, the differences among classes become more subtle, which create difficulties on fine classification [11]. ii) In dynamic networks, problems such as packet loss, retransmission, and disorder of packets may occur at any time [12]. Thus there is a compelling need to deal with incomplete and noisy data for classification. iii) In ubiquitous heterogeneous environments, the target classes are frequently changed over time [13]. For example, 3GPP (the 3rd Generation Partnership Project) defines 4 target classes, including conversation, streaming, interaction, and background. There are 6 classes in ITU-T Y.1541. If a traffic flow is transformed from 3GPP to ITU-T, the target classes will be different. These investigations motivate us to develop a network traffic classification model (termed GCCM), which is expected to classify traffic into fine and time-varying classes under dynamic network environments with noise and congestion. The major contributions of this paper are summarized as follows.

- *Granular relation matrix* (GRM). GRM is proposed for the first time in this paper. It reflects the spatial and temporal correlation between granules. The existing flow features, such as mean variance and kurtosis, are just a special case of GRM (at the maximum observation scale). On the basis of holder index α, the inherent relationship between packets indicated by GRM can achieve fine classification even under congestion.
- *Flow granules*. Based on the granular computing, flow granules $\aleph_v(x)$ and $\aleph_t(y)$ are defined to make model GCCM less sensitive to incomplete and noisy data. The flow granules are generated by aggregating similar neighborhood packets. The information to be processed is aggregated packets. As a result, the missing data and incomplete information can be effectively solved in dynamic network environment.
- *Difference measurement* $D(\cdot)$, which is presented to measure the difference degree between matrices. Depending on $D(\cdot)$, GCCM use GRM to achieve a threshold based classification and thus is able to classify varied classes.

This work was supported in part by the NSFC under Grants 61271233, 62071005 and 62071252, the APNSFC under Grants 2008085MF222, 2208085MF155, and KJ2021A0124, and the NSF under Grant ECCS-1923717. This work was presented in part at IEEE INFOCOM WKSHPS [14].

P. Tang is with the School 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).

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

H. Wei is with the Department of Automatic Control and Systems Engineering (ACSE), the University of Sheffield, Sheffield, UK (e-mail: w.hualiang@sheffield.ac.uk).

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

TABLE I
OVERVIEW OF PREVIOUS STUDIES AND OUR WORK

Approaches	[4], [15]	[8], [9], [20], [21]	[10]	[11], [27]	[13], [17], [18], [30], [31]	[16], [31]	[19]	[22], [23]	[24]	[25],[26]	[28], [29]	GCCM
Encrypted traffic		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Throughput		✓	✓		✓		✓	✓			✓	✓
Fine classes	✓		✓	✓	✓	✓			✓	✓		✓
Varied classes	✓	✓	✓					✓	✓			✓
Noise tolerance	✓	✓					✓			✓		✓
Congestion	✓			✓	✓						✓	✓

II. RELATED WORK

A number of approaches have been proposed to implement traffic classification, e.g., the technique of deep packet inspection (DPI). DPI approaches are based on the payload to achieve classification, which are not affected by dynamic network environments (e.g., new classes and congestion) [15]. DPI is more accurate than other approaches when classifying unencrypted traffic [16]. For example, Yun *et al.* [4] exploited the semantic information in protocol message formats to identify real-world network traces. The experimental results on BitTorrent, FTP, SMTP, etc., show that the scheme has an average recall of about 97.4% and an average precision of about 98.4%. However, inspection of packet payload is time-consuming and breaches the privacy of users [9], [17]. In addition, access to the payload is often not possible since 90% of the traffic is encrypted [18]. Besides, it is not easy for DPI to identify new classes or unknown classes since the keywords, signatures, certificates, and cookies of the unknown classes are totally unknown [10]. The applicability of DPI is hence limited. Some methods for encrypted traffic exploited the fixed registered ports [6]. However, port based methods became inaccurate due to dynamic reuse of ports and new applications with unregistered or random generated ports. In [19], Zou *et al.* exploited the physical state and resource usage monitoring to implement classification of encrypted traffic. Nevertheless, the proposed phenotyping mechanism in [19] can only identify five fixed classes, including aggregation, broadcast, consensus and DGD (Distributed Gradient Descent).

At present, one of the most popular techniques for varied classes are statistical features (SFs) [8], [9]. SFs are obtained by analyzing the packet sizes and intervals, without inspecting the payload information. In contrast to DPI, it thus can be used to classify encrypted traffic as well as unencrypted traffic at a considerable speed [20], [21]. For example, Nossenson *et al.* [22] classified videos into live streaming and video on demand (VoD) based on the SFs of packet length, information offset, etc. Thay *et al.* [23] proposed a classification technique based on the number of peer connection in both incoming and outgoing directions within a 5-minute duration to classify P2P traffic, including BitTorrent, Skype, SopCast, etc. However, SFs usually do not work well for fine classification [24]. For example, the largest packets of the SD, HD, and UD video flows are all 1494 bytes. Other SFs, such as duration, mean packet size and skew, are also basically the same, which are invalid when utilized for fine classification of SD, HD, and UD video flows [25]. When the number of classes is increased, the differences in SFs between classes become subtle. It is of necessity to conduct further study and explore more effective methods. Wu *et al.* [26] proposed a chain and hierarchical structure (CHS) to make up for the defects of SFs. However, CHS has the chain effect of error propagation. When the number of classes is increased, the number of classifiers is increased, and thus the cumulative error on each classifier will be greatly increased.

Some of the explorations proposed behavior features to implement traffic classification [27], [28]. Behavior features are different from SFs. The latter suppose that the packets are independent of each other, while the former are on the basis of the close relationships among packets. For instance, Chen *et al.* [29] found that large-size messages from the server interacting with small-size messages from the client (and vice versa) are frequently observed in video or P2P traffic flows whereas rarely appear in HTTP and other types of traffic flows, and each traffic type has distinct sequential message pattern. Behavior features can be used to identify traffic flows. Compared with SFs, behavior features are more adopted for fine classification. However, behavior features usually do not work well for traffic flows with noise. In [30], the behavior features are based on the key packets from the first few seconds of the flow to achieve online classification, but they may not achieve the expected classification results if the key packets are lost. In [31], Hybrid features, i.e., SFs plus behavior features, are proposed to mitigate the shortcomings of SFs and behavior features. However, the performance of classification may not be improved by just a simple addition of features. Fine classification of noisy traffic is still hard to deal with. It is necessary to explore other avenues to overcome these obstacles for online traffic classification.

Accordingly, we presented a new model GCCM to achieve online classification of both encrypted and unencrypted traffic for fine and varied classes, with further resilience to noise: i) GCCM analyzes the burst features of packets, without inspecting the payload information, and thus can be used to classify encrypted traffic as well as unencrypted traffic at a fast speed. ii) The burst feature of GRM proposed in this paper reflects the spatial and temporal correlation between granules. The inherent relationship between packets indicated by GRM can achieve fine classification even under congestion. iii) Besides, we explored granular computing to calculate GRM, making GCCM less sensitive to incomplete and noisy data. iv) In addition, the proposed model GCCM, depending on difference measurement $D(\cdot)$, is a kind of threshold based classification, and thereby can be used to identify time-varying classes. The gaps between GCCM and other literatures can be broadly summarized in Table I.

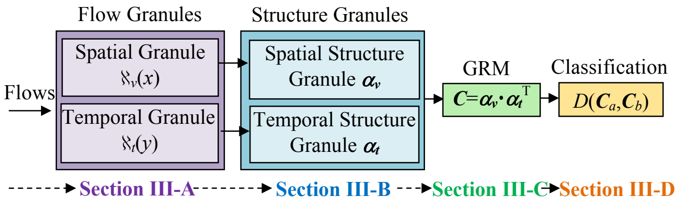


Fig. 1. Block diagram for the organization of Section III.

III. The Proposed Model GCCM

A group of scientists explored the manner of human thinking and learning, and proposed a new mechanism called *granular computing* [32]–[34]. By studying the process of human recognition, Zadeh [35] found that human divided an object into granules for analysis. Pal *et al.* [36] also pointed out that the contents of information that human observe, measure, and reason are all granules. Granular computing reasons and analyzes the relationships between granules, which can filter out interference and noise, and handle missing or incomplete data. Based on the principles and mechanisms of granular computing, the framework of GCCM basically consists of four steps as shown in Fig. 1: i) define flow granules: spatial granule $\aleph_v(x)$ and temporal granule $\aleph_t(y)$; ii) explore the relationships between granules (i.e., structure granules); iii) establish the novel flow feature of GRM; iv) achieve classification of traffic flows on the basis of GRM and $D(\cdot)$.

A. Flow Granules

In this study, the basic granules (i.e., flow granules) are defined based on the concept of neighborhood granules. Before proceeding, an accurate definition of flow is first provided as follows. Traffic is composed of flows, and the flows aggregate into traffic. Some flows are unidirectional (e.g., uplink or downlink), while others are bidirectional. The characteristics of uplink and downlink packets are often quite different, which should be calculated separately. Therefore, the kth flow F_k is defined as a set of packets with the same five-tuple. The five-tuple refers to $\{SrcIP, DestIP, SrcPort, DestPort, Protocol\}$, where $SrcIP$, $DestIP$, $SrcPort$ and $DestPort$ denote the source IP address, destination IP address, source port, and destination port, respectively. The flow sequence is described as

$$F_k \triangleq \{(P_i, T_i)\,|_{i=1,2,\ldots,N_r}\}, \tag{1}$$

where P_i refers to the size of the ith packet, T_i is the inter-arrival time between the ith packet and the previous packet, and resolution N_r refers to the number of packets in F_k. Based on the concept of neighborhood granules proposed by Pal *et al.* [36], two types of flow granules are presented: spatial and temporal granules. The former is defined as

$$\aleph_v(x) = \bigcup_{i=k}^{j} P_i \in U \tag{2}$$

$$\text{s.t.} \quad |P_i - P_{i+1}| < Thr_v, \tag{3}$$

where symbols j and k are the sequence numbers of the packets, and the values of j and k depend on the flow data. U refers to the complete set. If the neighborhood packets have a similar packet size (Thr_v is the threshold, and the details about the settings of Thr_v refer to Section IV-B), they will be aggregated into the same granule $\aleph_v(x)$, and thus $\{\aleph_v(x)\}|_{x=1,2,\cdots,X}$ can be obtained, where X is the number of the spatial granules. Take an email flow as an example, which is captured by packet capture software (e.g., Wireshark). $\{P_i\}$ is obtained as $\{60, 76, 60, 239, 84, 76, 90, 67, 83, 67, 460, \cdots\}$. If Thr_v is set to 100, the spatial granules are

$$\aleph_v(1) = \bigcup_{i=1}^{3} P_i = \{P_1, P_2, P_3\} = \{60, 76, 60\},$$

$$\aleph_v(2) = \bigcup_{i=4}^{4} P_i = \{P_4\} = \{239\},$$

$$\aleph_v(3) = \bigcup_{i=5}^{10} P_i = \{84, 76, 90, 67, 83, 67\}, etc.$$

Similar to spatial granule, temporal granule is defined as

$$\aleph_t(y) = \bigcup_{i=k}^{j} T_i \in U \tag{4}$$

$$\text{s.t.} \quad |T_i - T_{i+1}| < Thr_t. \tag{5}$$

From (5), if the neighborhood packets have similar inter-arrival time (Thr_t is the threshold, and the details about the settings of Thr_t refer to Section IV-B), they will aggregate into the same granule, and thus $\{\aleph_t(y)\}|_{y=1,2,\cdots,Y}$ is obtained, where Y is the number of the temporal granules. The members in granules $\aleph_v(x)$ and $\aleph_t(y)$ are similar neighborhood packets (i.e., the neighborhood packets have similar size or similar inter-arrival time), so the calculation model is less sensitive to missing data and can remove the noisy data as well, which is one of the basic ideas of granular computing.

B. Structure Granules

By exploring the human reasoning patterns, Zadeh [35] found that human analyze issues from various perspectives, and can shuttle up and down at these perspectives to make a synthetic diagnosis. Imitating such patterns, granular computing decomposes or merges the granules from different perspectives or levels (scales) to obtain structure granules. Granular computing studies the inherent relationship between granules at different perspectives or levels (scales). However, the idea of analyzing the changes of a process in different scales is not new. In fact, date back to at least the late 1960s, Mandelbrot used the concept of scales to study the traits of objects [37]. Suppose $\{F(t)\}$ is a stochastic process, and the measurement $\mu(\varepsilon)$ and the observation scale ε satisfy

$$\mu(\varepsilon) \propto \varepsilon^{\alpha}. \tag{6}$$

That is,

$$\alpha \triangleq \frac{\ln \mu(\varepsilon)}{\ln \varepsilon}, \tag{7}$$

where α is called the *holder index* or *singularity index*, which has been widely used in prediction of gas emission in mines, classification of hydrological and water resources, anti-interference treatment of artificial scenes [38].

According to (1), flows satisfy the definition of $\{F(t)|_{(t=i)}\}$ proposed by Mandelbrot. In (7), ε is a continuous variable. It needs to be sampled to apply to discrete flow sequence F [39], and thus the structure granules are established as

$$\boldsymbol{\alpha} \triangleq \left\{\frac{1}{m}\ln\mu_m\,|_{m=1,2,\ldots,Z}\right\} \tag{8}$$

$$\text{s.t.}\quad \mu_m \triangleq \sum_{k=1}^{\frac{Z}{m}}\left|\sum_{i=1}^{m}\bar{\aleph}\left(m\left(k-1\right)+l\right)\right|^2, \tag{9}$$

where $\aleph(\cdot)$ refers to the spatial granule $\aleph_v(x)$ or temporal granule $\aleph_t(y)$; $\bar{\aleph}(\cdot)$ is the average of members in the flow granule; $Z=\{X,Y\}$ is the number of flow granules; and m refers to the observation scale. The minimum scale is $m=1$, which means that each flow granule is treated as a separate granule; the maximum scale is $m=Z$, which means that all flow granules merge into one granule, corresponding to the SF of "average packet size". Therefore, SF is a special case of structure granules when the observation scale reaches the maximum. More concretely, structure granules captures the varying process of a flow when the observation scale m is changed from 1 to N_r.

C. Granular Relation Matrix (GRM)

In Section III-A, two types of flow granules are defined: spatial and temporal granules. Substituting the two types of granules into (6)–(8), two types of structure granules can be obtained: spatial structure granule $\boldsymbol{\alpha}_v$ and temporal structure granule $\boldsymbol{\alpha}_t$. The former describes the changing traits of packets size, while the latter describes the bursting traits of packets at different scales. The two vectors are cross multiplied to obtain the GRM, which describes the changing traits of bursty data at different spatial and temporal scales:

$$\boldsymbol{C}|_{X*Y} \triangleq \boldsymbol{\alpha}_v\cdot{\boldsymbol{\alpha}_t}^{\mathrm{T}}, \tag{10}$$

where $\boldsymbol{\alpha}_v$ is deduced from spatial granules $\aleph_v(x)|_{x=1,2,\cdots,X}$. Here, the minimum observation scale is $m=1$, while the maximum scale is $m=X$. Therefore, $\boldsymbol{\alpha}_v$ has X observations. Similarly, $\boldsymbol{\alpha}_t$ is deduced from temporal granules $\aleph_t(y)|_{x=1,2,\cdots,Y}$, and thus $\boldsymbol{\alpha}_t$ has Y observations when the time scale is changed from 1 to Y. T is the transpose of matrix. Therefore, the order of granular relation matrix $\boldsymbol{C}$ is $X*Y$.

Proposition 1. *Granular relation matrix $\boldsymbol{C}$ uniquely identifies the type of the network flow.*

Proof: Suppose there are two flows: F_a and F_b. For flow F_a, the spatial and temporal structure granules are $\boldsymbol{\alpha}_{va}$ and $\boldsymbol{\alpha}_{ta}$, respectively. For flow F_b, the spatial and temporal structure granules are $\boldsymbol{\alpha}_{vb}$ and $\boldsymbol{\alpha}_{tb}$. Then, the observation scale of time for the temporal structure granules is fixed as $\boldsymbol{\alpha}_t|_{m=y_0}$, and only study the changing traits of packets size $\boldsymbol{\alpha}_v$. Here, we aim to compute $\boldsymbol{\alpha}_{vz}$ of the aggregated flow $Z=F_a+F_b$. According to the theory proposed by Mandelbrot, ε in (6) is a continuous variable. Thus, (8) can be obtained by sampling the observation scale ε as

$$\boldsymbol{\alpha}=\{\alpha|_{\ln\varepsilon=m}\}\triangleq\left\{\lim_{\ln\varepsilon\to m}\frac{\ln\mu(\varepsilon)}{\ln\varepsilon}\right\}, \tag{11}$$

where α is a continuous variable, and $\boldsymbol{\alpha}$ is a vector. The members of $\boldsymbol{\alpha}$ are sampled from α. From (6), $\mu_a(\varepsilon)\propto\varepsilon^{\alpha_{va}}$, $\mu_b(\varepsilon)\propto\varepsilon^{\alpha_{vb}}$, and then

$$\alpha_{vz}=\lim_{\ln\varepsilon\to m}\frac{\ln\left(\mu_a\left(\varepsilon\right)+\mu_b\left(\varepsilon\right)\right)}{\ln\varepsilon}. \tag{12}$$

Hence the boundaries of α_{vz} can be deduced as

$$\inf\left(\alpha_{vz}\right)=\lim_{\ln\varepsilon\to m}\frac{\ln\sqrt{2\mu_a\left(\varepsilon\right)\mu_b\left(\varepsilon\right)}}{\ln\varepsilon}=\frac{1}{2}\left(\alpha_{va}+\alpha_{vb}\right), \tag{13}$$

$$\sup\left(\alpha_{vz}\right)=\lim_{\ln\varepsilon\to m}\frac{2\max\left(\mu_a\left(\varepsilon\right),\mu_b\left(\varepsilon\right)\right)}{\ln\varepsilon}=\max\left(\alpha_{va},\alpha_{vb}\right). \tag{14}$$

In particular, when $\alpha_{va}=\alpha_{vb}=\alpha_v$, $\inf(\alpha_{vz})=\sup(\alpha_{vz})=\alpha_v$, which indicates that, if flow F_a belongs to the same class as flow F_b, then the aggregated flow $Z=F_a+F_b$ will fall in the same class. If flows F_a and F_b belong to different classes, the spatial structure granule α_{vz} of the aggregated flow Z would be neither α_{va} nor α_{vb}. Therefore, we prove that the vector $\boldsymbol{\alpha}_v$, samples of α_v with different scale m as in (11), is unique under fixed temporal scale $\alpha_t|_{m=y_0}$. As a result, the orthogonal matrix of all members $\boldsymbol{C}=\boldsymbol{\alpha}_v\cdot{\boldsymbol{\alpha}_t}^{\mathrm{T}}$ can uniquely identify the type of network flow.

In addition, we provide a detailed description on the physical meaning of GRM here. GRM is based on the holder index $\boldsymbol{\alpha}$. According to the theory proposed by Mandelbrot, $\boldsymbol{\alpha}$ represents the burst index of objects, and $\ln\mu_m$ refers to the burst amount when the observation scale is m. Based on (11), we calculate $\boldsymbol{\alpha}$ for the spatial granules (i.e., packet sizes) and temporal granules (i.e., packet intervals), respectively, and thus obtain vectors $\boldsymbol{\alpha}_v$ and $\boldsymbol{\alpha}_t$:

$$\boldsymbol{\alpha_v}=\{\alpha_v|_{\ln\varepsilon_1=m}\}\triangleq\left\{\lim_{\ln\varepsilon_1\to m}\frac{\ln\mu_{vm}(\varepsilon_1)}{\ln\varepsilon_1}\right\},$$
$$\boldsymbol{\alpha_t}=\{\alpha_t|_{\ln\varepsilon_2=n}\}\triangleq\left\{\lim_{\ln\varepsilon_2\to m}\frac{\ln\mu_{tn}(\varepsilon_2)}{\ln\varepsilon_2}\right\},$$

where ε_1 and ε_2 are the observation scales, $ln\mu_{vm}$ refers to the burst amount of packet sizes when the observation scale is m ($\ln\varepsilon_1=m$), and $\boldsymbol{\alpha_v}$ represents the burst index of packet sizes. $ln\mu_{tn}$ refers to the burst amount of intervals when the observation scale is n, and $\boldsymbol{\alpha_t}$ is the burst index of packet intervals. $\boldsymbol{\alpha_v}$ and $\boldsymbol{\alpha_t}$ are cross multiplied to obtain GRM:

$$\mathbf{C}|_{X*Y}\triangleq\boldsymbol{\alpha}_v\cdot{\boldsymbol{\alpha}_t}^{\mathrm{T}}=\{\alpha_v\cdot\alpha_t|_{\ln\varepsilon_1=m,ln\varepsilon_2=n}\}$$
$$=\left\{\lim_{\ln\varepsilon_1\to m}\lim_{\ln\varepsilon_2\to n}\frac{\ln\mu_{vm}(\varepsilon_1)\ln\mu_{tn}(\varepsilon_2)}{\ln\varepsilon_1\ln\varepsilon_2}\right\}.$$

For ease of understanding, the burst amount of packet sizes $ln\mu_{vm}$ is supposed to be a green surface as shown in Fig. 2. Consequently, its observation scale $\ln\varepsilon_1$ is also a surface. The burst amount of intervals $ln\mu_{tn}$ is supposed to be a line segment and hence its observation scale $\ln\varepsilon_2$ is also a line segment. $ln\mu_{vm}$ multiplied $ln\mu_{tn}$ turns out to be a volume and its observation scale $\ln\varepsilon_1\ln\varepsilon_2$ is a small cube. That is, $ln\mu_{vm}ln\mu_{tn}$ refers to the burst amount of volume. Just as $\boldsymbol{\alpha_v}$ represents the burst index of packet sizes and $\boldsymbol{\alpha_t}$ represents the burst index of packet intervals, the physical meaning of GRM is the burst index of the traffic volume.

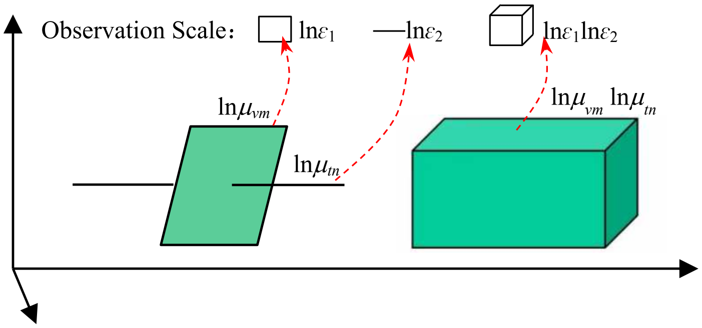


Fig. 2. The physical meaning of GRM.

D. Differences Between GRMs

For a certain type of flows, they always follow a specific communication protocol and transmission pattern, so that they have similar variations reflecting the inherent traits. Due to this reason, SFs (e.g., mean packet size, maximum and minimum packets) are used to identify different flows. However, as described in Section III-B, these SFs are static, which cannot reflect the varying features of traffic. In contrast, GRM not only contains the SFs, but also describes the varying features reflecting the deeper nature. GRM depicts the traits more comprehensively. Consequently, it can be used to achieve accurate identification of network flows for fine classes.

Matrix, which physically refers to a certain transformation, describes the movement track. For example, $y = \boldsymbol{A}x$, where matrix $\boldsymbol{A}$ represents the movement track from state x to y in space Q. If we stand in space R to observe this movement, we have $y' = \boldsymbol{B}x'$, where x' and y' respectively correspond to the state of x and y in the new space R, and matrix $\boldsymbol{B}$ represents the movement track from state x' to y'. So we have $\boldsymbol{H}x' = x$, $\boldsymbol{H}y' = y$. Then $\boldsymbol{H}y' = y = \boldsymbol{AH}x' = \boldsymbol{H}(\boldsymbol{H}^{-1}\boldsymbol{AH})x'$. That is, in space R, the movement track from state x' to y' can be described by $\boldsymbol{B} = \boldsymbol{H}^{-1}\boldsymbol{AH}$. It can be seen that the similar matrices of $\boldsymbol{A}$ and $\boldsymbol{B} = \boldsymbol{H}^{-1}\boldsymbol{AH}$ essentially describe the same movement, which are observed in different space. GRM describes the trajectory of bursty data at different observation scales. The similarity of two GRMs is measured as

$$D\left(\boldsymbol{C}_a, \boldsymbol{C}_b\right) \triangleq \frac{\boldsymbol{C}_a{\boldsymbol{C}_b}^{\mathrm{T}} + \boldsymbol{C}_b{\boldsymbol{C}_a}^{\mathrm{T}}}{\boldsymbol{C}_a{\boldsymbol{C}_a}^{\mathrm{T}} + \boldsymbol{C}_b{\boldsymbol{C}_b}^{\mathrm{T}}}, \tag{15}$$

where $\boldsymbol{C}_a$ and $\boldsymbol{C}_b$ refer to the GRMs of flows F_a and F_b, respectively. Suppose the order of matrix $\boldsymbol{C}_a$ is $X_a * Y_a$, and that of $\boldsymbol{C}_b$ is $X_b * Y_b$. When comparing $\boldsymbol{C}_a$ and $\boldsymbol{C}_b$ by (15), the comparison should be made at the same observation scale, so the dimensions are selected to be $\min(X_a, X_b)$ and $\min(Y_a, Y_b)$. For similar matrices $\boldsymbol{A}$ and $\boldsymbol{H}^{-1}\boldsymbol{AH}$, tr$(\boldsymbol{H}^{-1}\boldsymbol{AH})$ = tr$(\boldsymbol{HH}^{-1}\boldsymbol{A})$ = tr$(\boldsymbol{A})$, where tr$(\cdot)$ refers to the trace of matrix. Similar matrices have the same trace. In addition, GRM $\boldsymbol{C}$ is the cross product of spatial structure granule $\boldsymbol{\alpha}_v$ and temporal structure granule $\boldsymbol{\alpha}_t$. Therefore, tr$(\boldsymbol{\alpha}_t\boldsymbol{\alpha}_v^{\mathrm{T}})$= $\boldsymbol{\alpha}_t\boldsymbol{\alpha}_v^{\mathrm{T}}$. Then the similarity measurement matrix in (15) is converted into a scalar, called the difference degree:

$$Dif\left(\boldsymbol{C}_a, \boldsymbol{C}_b\right) \triangleq 1 - \frac{\mathrm{tr}\left(\boldsymbol{C}_a{\boldsymbol{C}_b}^{\mathrm{T}} + \boldsymbol{C}_b{\boldsymbol{C}_a}^{\mathrm{T}}\right)}{\mathrm{tr}\left(\boldsymbol{C}_a{\boldsymbol{C}_a}^{\mathrm{T}} + \boldsymbol{C}_b{\boldsymbol{C}_b}^{\mathrm{T}}\right)}. \tag{16}$$

According to (16), $Dif(\boldsymbol{C}_a, \boldsymbol{C}_b)$=$Dif(\boldsymbol{C}_b, \boldsymbol{C}_a)$, and $Dif(\cdot)$ is between 0 and 1. $Dif(\cdot)$ is used to measure the difference degree between matrices. The smaller the value of $Dif(\cdot)$, the smaller the difference, and the higher the similarity. In the extreme case, $Dif(\boldsymbol{C}_a, \boldsymbol{C}_a)$=0, which means there is no difference between the two matrices.

E. Semi-supervised Classification and Threshold Setting

Suppose there are L classes $\{M_l\}_{l=1}^{L}$, and several flows $\{\cdots, F_j, \cdots, F_k, \cdots\}$ in each class. The centers of classes are $\{P_l\}_{l=1}^{L}$. As described in Section III-D, $Dif(\cdot)$ is uniformly distributed between 0 and 1. Therefore, the center P_l is determined by

$$P_l \triangleq \min_{F_k \in \mathrm{M}_l} \left\{ \max_{j \neq k, F_j \in \mathrm{M}_l} Dif\left(\boldsymbol{C}_{F_j}, \boldsymbol{C}_{F_k}\right) \right\}. \tag{17}$$

According to (17), the difference degree between P_l and other flows $\{\cdots, F_j, \cdots, F_k, \cdots\}$ is the smallest. In order to judge whether a flow F_k belongs to the lth class M_l, it just needs to calculate the difference degree between flow F_k and the class center, i.e., $Dif(\boldsymbol{C}_{F_k}, \boldsymbol{C}_{P_l})$. If the difference degree is less than or equal to the threshold, then F_k belongs to class M_l; otherwise F_k does not belong to class M_l. That is,

$$\begin{cases} F_k \in M_l, \text{ if } \{Dif\left(\boldsymbol{C}_{F_k}, \boldsymbol{C}_{P_l}\right) \leq T\}, \\ F_k \notin M_l, \text{ if } \{Dif\left(\boldsymbol{C}_{F_k}, \boldsymbol{C}_{P_l}\right) > T\}. \end{cases} \tag{18}$$

The proposed traffic classification model is a semi-supervised learning. The system is first trained based on manually labeled samples. Then unlabeled samples are gradually added into the learning system and are classified by (18). When the number of samples accumulates to a certain amount, the system parameters (e.g., threshold T) will be adjusted. In (18), threshold T significantly affects the performance of the system. The maximum between-class variance (Otsu) method is adopted to establish an adjustment mechanism for the global optimal threshold as

$$\mathrm{T}^* = \arg\max \sum_{\mathrm{i} \neq \mathrm{j}} \left(Dif^2\left(t; M_i \leftrightarrow M_j\right) \right), \tag{19}$$

where $Dif(t; M_i \leftrightarrow M_j)$ is the difference degree between M_i and M_j when the threshold is set to t. According to Otsu, the maximum variance between classes implies the smallest false rate: $\min(frr + far)$, where frr is the false rejection rate and far is the false acceptance rate.

As in (17) and (18), the basic principle of classification is based on k-means. In order to prevent parameter solidification, here an improvement is made on the threshold adjustment using the idea of genetic algorithm. According to biological evolution theory, genes need to be crossed and mutated. Therefore, the thresholds are randomly adjusted (i.e., the mutation operation) to obtain new centers, and choose the better one between the old and the new one. The procedure of threshold adjustment is presented in Algorithm 1, where $t(e)$ is randomly adjusted to $t(e) \pm \Delta$. In the iterations, if the difference is obviously increased, the threshold and center are updated. Otherwise, Δ is continuously indented by 1/2 (i.e., dichotomy). Thus, the iterative calculation of the threshold 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

Algorithm 1: Setting the Threshold

Input: $F_k|_{(k=1,2,\cdots,N_s)}$;
Output: $T = t(e+1)$;
for *unlabeled flows* F_k **do**
  { Calculate $Dif(\boldsymbol{C}_{F_k}, \boldsymbol{C}_{P_l})$;
  Find $\min = \min\limits_{L} Dif\left(\boldsymbol{C}_{F_k}, \boldsymbol{C}_{P_l}\right)$ and compare with $t(e)$;
  if $\min \leq t\left(e\right)$ **then**
    Put F_k into class M_l;
  end
  else
    Create new class M_{L+1};
    Put F_k into class M_L;
  end
end
Update centers $\{P_l\}_{l=1}^{L}$;
do
  $\{\sigma(e+1), \sigma'(e+1)\} = \frac{1}{\{i,k\}} \sum Dif^2\left(\{\boldsymbol{C}_{P_i}, \boldsymbol{C}_{F_k}\}, \boldsymbol{C}_{P_l}\right)$;
  if $|\sigma(e+1) - \sigma(e)| < 0$ **then**
    $t(e+1) = t(e) \pm \Delta$;
  end
  Update centers $\{P_l\}_{l=1}^{L}$ and $\sigma'(e+1)$;
while $|\sigma'(e+1) - \sigma'(e)| > \varepsilon$;
return $T = t(e+1)$;

Algorithm 2: Classification of Traffic Flows

1 Input: flow F_k;
2 Output: Re;
3 Obtain flow sequence (P_i, T_i) by (1);
4 Partition flow sequence into subflows;
5 **for** *subflow* **do**
6   Calculate: // (see Section III-A)
    Spatial granules $\aleph_v(x) = \bigcup P_i \in U$;
    Temporal granules $\aleph_t(y) = \bigcup T_i \in U$;
7 **end**
8 Obtain spatial structure granules $\boldsymbol{\alpha}_v$ and temporal structure granules $\boldsymbol{\alpha}_t$; // (see Section III-B)
9 Establish GRM: $\boldsymbol{C}_{F_k} = \boldsymbol{\alpha}_v \cdot {\boldsymbol{\alpha}_t}^{\mathrm{T}}$; // (see Section III-C)
10 **for** *each class* $c_l|_{l\leq L}$ **do**
11   Compare $\boldsymbol{C}_{F_k}$ with typical GRM: $\boldsymbol{C}_{p_l}$;
12   Difference between GRMs is $Dif(\boldsymbol{C}_{F_k}, \boldsymbol{C}_{p_l})$; //(see Section III-D)
13 **end**
14 **if** $Dif(\boldsymbol{C}_{F_k}, \boldsymbol{C}_{p_l}) \leq T_l$ **then**
15   $Re = 1$; // F and P_l are of the same class
16 **else**
17   $Re = 0$; // F and P_l are of the different class
18 **end**
19 **return** Re;

iteration. In the worst case, the proposed algorithm $(t(e) \pm \Delta)$ degenerates back to the original k-means algorithm $(t(e))$.

The complete classification process of GCCM illustrated in Algorithm 2 is summarized as follows: i) GCCM classify flow F_k according to its bitstream as in Line 3. Our method does not need the payload, and thus it is able to deal with encrypted traffic flows as well as unencrypted traffic. ii) In order to reduce the computation, flows are divided into subflows [10] as in Line 4. More details about the settings of resolution N_r for subflow can be obtained in Section IV-B. iii) GRM, which is based on granules $\aleph_v(x)$ and $\aleph_t(y)$ as in Line 6, is effectively cope with missing, incomplete, or noisy data.

TABLE II
DATASETS

Dataset	Year	Linktype	Volume	Flows
WIDE	2020	backbone	33GB	80K
UCI	2020	edge	29GB	46k
NJUPT	2018	edge	42GB	106k
ISP	2018	backbone	36GB	77K
UNB	2016	edge	28GB	65K
UNIBS	2009	edge	27GB	79k

IV. Configuration and Parameter Settings

A. Datasets and Traffic Classes

In this paper, four public datasets (i.e., UNB, UNIBS, WIDE, and UCI) and two private datasets (i.e., NJUPT and ISP) are used to evaluate the classification performance as shown in Table II. The WIDE traces (http://mawi.wide.ad.jp/mawi/) began on Jun, 2020 and were taken from a US-Japan-Pacific backbone line (a 150-Mb.s Ethernet link) that carries commodity traffic for WIDE organizations. The UCI (http://archive.ics.uci.edu/ml/index.php) maintains 557 datasets, including the YouTube Collection Dataset, the Spam Base Dataset, etc., from which various types of traffic is obtained. The NJUPT traces are captured by Wireshark in the campus network of Nanjing University of Posts and Telecommunications. The ISP traces are collected at a leading Internet service provider of China (the name of the city is omitted as required by commercial confidentiality). This traces contains important surveillance and conferencing videos, such as Ezviz and Gotomeeting. The UNB trace (http://www.unb.ca/cic/research/datasets/vpn.html) has many network applications. Researchers are allowed to read the full payload trace. The UNIBS traces (http://netweb.ing.unibs.it/ntw/tools/traces/index.php) are collected from the edge routers of the campus network of the University of Brescia, which include the applications such as Edonkey, Skype, and BitTorrent.

In the field of traffic classification, one of the first important issues is how to define the classes [40]. Most of the classes in the prior works such as [41], [42] are application-based, and consequently the traffic is labeled as YouTube, Facebook, Skype, QQ, Tik Tok, WeChat, etc. However, after carefully observing the datasets, we have figured out the following: i) One application might generate different types of bitstreams. For instance, WeChat generates video and audio flows. Clearly, although they are from the same application, WeChat video and audio need to be classified into different classes from the perspective of network differentiated services. ii) Some applications, such as QQ and WeChat, which were developed with a similar mechanism, often generate similar types of video bitstreams. In summary, different applications may generate similar types of bitstreams, while the same application may generate different types of bitstreams. Therefore, in this paper, we define the classes from the perspective of NRQ (Network Resource and QoS Requirement). The mapping between the NRQ classes and typical applications is presented in Table III.

TABLE III
CLASSES OF NETWORK TRAFFIC

Coarse classes	NRQ classes	Label	Typical Apps
Video	Video conferencing	1	Gotomeeting
	Telemedicine	2	FsMeeting
	Instant messaging videos	3	QQ, WeChat
	Group chat videos	4	Skype
	E-commerce	5	Direct connect
	Unidirectional videos	6	PPlive
	Bidirectional videos	7	TVant
	multidirectional videos	8	BitTorrent
	BT video on demand	9	Jjvod
	SD	10	
	HD	11	Youku, Tudou
	UD	12	
	Video broadcast	13	UUSee
	Video surveillance	14	Ezviz
Audio	Audio conversation	15	QQ, WeChat
	P2P audio	16	Peergine,
	Online music	17	TTplayer,
	Audio broadcasting	18	GoldenRadio
WB	Web browsing	19	Baidu, Blogger
TC	Text communication	20	Baidu, Blogger
Email	Email	21	Gmail, Hotmail
File transfer	FTP	22	Baidu Netdisk
	P2P	23	Baidu Netdisk

TABLE IV
SETTING OF THRESHOLD

Thr_v	Spatial granules : $\aleph_v(x)$
1500 1300 1100	$\{\cdots$ 1194,1194,1194,1117,60,60,45,1141,82,1133,1141,$\cdots\}$
1000 500 30	$\{\cdots$ {1194,1194,1194,1117},{60,60,45},{1141},{82},$\cdots\}$
10 5 1	$\{\cdots$ {1194,1194,1194},{1117},{60,60},{45},{1141},$\cdots\}$

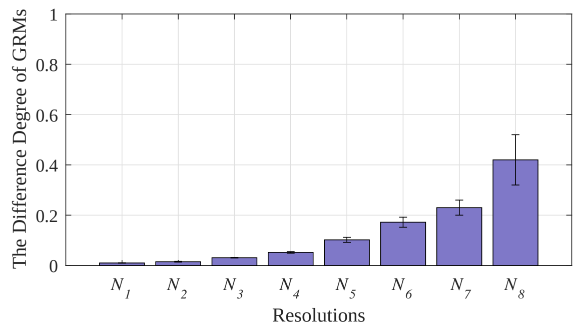


Fig. 3. Setting of resolution.

B. Parameter Settings

The most important parameters in this paper are Thr_v, Thr_t and resolution N_r.

i) Thresholds Thr_v and Thr_t, which control the size of spatial granules and temporal granules respectively and thus determine the capability of noise tolerance. These thresholds are easy to set in practice. Taking an email flow as an example, the spatial granules under different Thr_v are demonstrated as shown in Table IV. No matter what the value of Thr_v is set to, $\aleph_v(x)$ has only three results. If Thr_v is greater than 1000, the size of granule will be too large: all packets are fused into one granule, and thus the differences between granules for classification cannot be obtained. If Thr_v is lower than 10, the size of granule will be too small. In the extreme case, each packet is a granule and thus granular computing lost its function (note that granular computing aims at analyzing objects with granules rather than individual elements). Therefore, the suitable Thr_v for email flows locates between 10 and 1000. For other types of traffic, a suitable Thr_v is located between 10 and a $(300 < a < 1000)$. Therefore, threshold Thr_v is finally set to 100 in this paper. Threshold Thr_t is also based on the same simple manual observation, and finally set to 0.001 in this paper. What needs to be especially emphasized is that the size of granule will not increase or decrease in linear manner with thresholds Thr_v and Thr_t, but in a jumping manner. As shown in Table IV, when Thr_v is set to 10 to 1000, the spatial granules are basically the same. Therefore, the performance of classification will not get much better when the settings of thresholds Thr_v and Thr_t are further improved.

ii) Resolution N_r. The length of flows is of great difference. Short flows, such as email, may have only a few hundreds of Byte. Many text flows are below 1MB. Long flows (e.g., videos) are usually as large as several MB. Longer flows (e.g., streaming media) may last more than one hour. In practice, long flows are divided into subflows to reduce the computation as shown in Algorithm 2 Line 4. The resolution of subflow is set to $N_r = 5000$. These packets are enough to obtain the comprehensive traits of flows. N_r can certainly be further reduced. However, with a smaller number of packets, the difference degree of GRMs for the same type of flows will become larger, leading to unstable classification. Here, video flows are used to study the impact of resolution N_r on GRMs. The flows are segmented into subflows with different resolutions $N_i = \{20000, 10000, 8000, 5000, 2000, 1000, 500, 100\}$. The difference degrees of GRMs under $N_r = N_i$ are calculated by $Dif(\boldsymbol{C}_j, \boldsymbol{C}_k)|_{(N_r=N_i)}$. As shown in Fig. 3, when $N_r = N_1 = 20000$, the difference degree of GRMs for all subflows $Dif(\boldsymbol{C}_j, \boldsymbol{C}_k) \sim 0.011 \pm 0.002$, which is highly stable. With the decrease of N_r, the difference degree of GRMs becomes relatively more unstable. Especially, when $N_r = N_8 = 100$, $Dif(\boldsymbol{C}_j, \boldsymbol{C}_k) \sim 0.413 \pm 0.107$. The difference degree of GRMs for subflows becomes huge, which will cause great instability in classification. We repeatedly tested and verified the above situation with other classes of long flows, and the results are basically similar. Therefore, the resolution for long flows is set to $N_r = N_4 = 5000$, which not only ensures the stability of classification, but also only requires a small amount of computation and less storage space. For short flows (e.g., the email flow), the GRM features are the same whether the resolution is 2000, 3000, or other. In order to take into account long flows (upward compatibility), the resolution is finally set to 5000 for all flows.

C. Metrics of traffic classification

Precision, recall and F1-score are commonly used to measure the accuracy of traffic classification model [6]. Here we also use them to evaluate the classification performance.

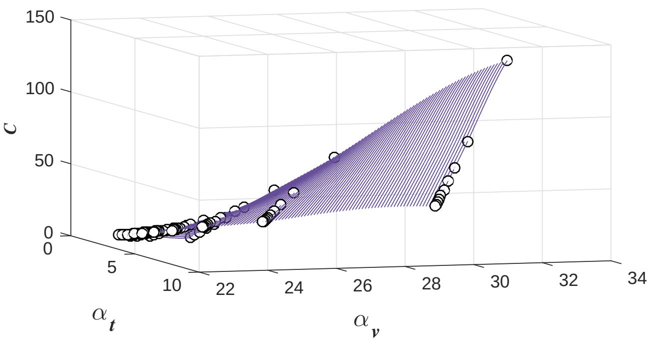


Fig. 4. The 3D surface of $\boldsymbol{C}$.

- *Precision*: the number of flows correctly classified as a class divided by the total number of flows classified as that class.
- *Recall*: the number of flows classified as a class divided by the total of flows actually belonging to that class.
- *F1-score*: be defined as a harmonic mean of precision and recall as follows.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{20}$$

V. Performance Evaluation

A. Evaluating the GRM of a Single Flow

In this experiment, a single video flow generated by Youku is used to demonstrate how to calculate the GRM. By packet capture software (i.e., Wireshark), we obtain the size and arrival time of each packet of flow F_k:{(470, 2.649745), (462, 2.650173), (1494, 2.650256), $\cdots$, (68, 359.282943), (1494, 359.434729), (1494, 359.493700)}, and thus get T_i, P_i as in (1). Then, the following three steps are executed.

(*Step i*): Scanning T_i, P_i to obtain the flow granules. According to (2)–(3) and (4)–(5), the neighborhood members are aggregated to form the spatial and temporal granules:

$$\begin{aligned}\aleph_v(x) =& \{\{470, 462\}, \{1494\}, \cdots, \{68\}, \{1494, 1494\}\},\\ \aleph_t(y) =& \{\{0.000428, 0.00083, 0.00045\}, \cdots, \{0.151786\},\\ & \{0.05897\}\}.\end{aligned}$$

(*Step ii*): Observing the above flow granules at various scales to form structure granules. For different observation scales $m = 1, 2, \cdots, \lceil \log N_r \rceil$, the structure granules $\boldsymbol{\alpha}_v$ and $\boldsymbol{\alpha}_t$ are generated by (8)–(9):

$$\begin{aligned}\boldsymbol{\alpha}_v =& \{32.345, 27.299, 25.560, 24.677, 24.159,\\ & 23.814, 23.567, 23.379, 23.225\},\\ \boldsymbol{\alpha}_t =& \{9.326, 7.229, 5.198, 4.704, 3.382,\\ & 2.152, 1.016, 0.926, 0.824\}.\end{aligned}$$

(*Step iii*): Generating GRM. According to (10), GRM is finally calculated to be $\boldsymbol{C} = \boldsymbol{\alpha}_v \boldsymbol{\alpha}_t{}^{\mathrm{T}}$. The corresponding 3D surface of $\boldsymbol{C}$ is shown in Fig. 4.

As shown in the first step, the spatial and temporal granules have different dimensions, so the dimensions of their corresponding structure granules are also different, which consequently causes the GRMs to have different orders. That is, for $\boldsymbol{C}_{X*Y}$, the values of X and Y are different. As discussed in Section III-D, the two GRMs should be compared at the same observation scale. Thus, the dimensions are selected to be $m = 1, 2, \cdots, \lceil \log N_r \rceil$ for all flow granules.

B. Dealing with the noise

Based on granular computing, flow granules makes GCCM less sensitive to missing, incomplete, or noisy data. Here, we take the spatial granules of an email flow as an example to demonstrate how flow granules eliminates the noise. The raw data $\{P_i\}$ captured by Wireshark is

$$\{60, 76, 60, 239, 84, 76, 90, 67, 83, 67, 460, 1456, \cdots\}, \tag{21}$$

Suppose P_6 is lost and P_7 is varied by noise. That is, (21) is changed into:

$$\{60, 76, 60, 239, 84, \cancel{76,} 170, 67, 83, 67, 460, 1456, \cdots\}, \tag{22}$$

Based on the technique of granular computing in (2), the spatial granules of the raw data and noisy data are

$$\{\{60, 76, 60\}, \{239\}, \{84, 76, 90, 67, 83, 67\}, \cdots\}, \tag{23}$$
$$\{\{60, 76, 60\}, \{239\}, \{84, 170, 67, 83, 67\}, \cdots\}, \tag{24}$$

After the averaging processing according to (9), we have

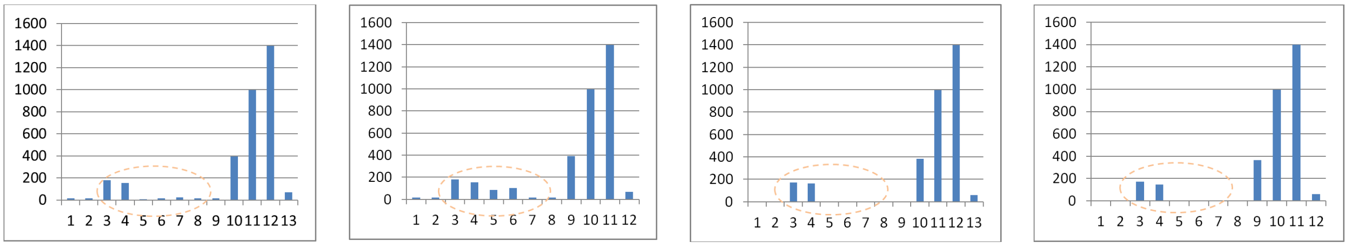
$$\{\{65, 65, 65\}, \{239\}, \{78, 78, 78, 78, 78, 78\}, \cdots\}, \tag{25}$$
$$\{\{65, 65, 65\}, \{239\}, \{94, 94, 94, 94, 94\}, \cdots\}, \tag{26}$$

In order to display the deviations in burst features between the raw data and the noisy data, we draw the burst amount between packet sizes (i.e., the absolute value of difference between two adjacent packet sizes) as shown in Figs. 5(a) and 5(b). Compared with the burst shape of (21) in Fig. 5(a) (the raw data), the burst shape of (22) in Fig. 5(b) (containing noisy data) is changed a lot, which will lead to some differences in burst index $\boldsymbol{\alpha}$. Actually, the proposed technique of granules will also lead to some deviations to the raw data in burst index as shown in Fig. 5(c). However, without granules, the noise will generate even greater deviations in burst index as shown in Fig. 5(b). By using flow granules, the burst shape of (26) in Fig. 5(d) remains consistent with that of (25) in Fig. 5(c), and thus their burst index $\boldsymbol{\alpha}$ are basically the same. That is, the proposed flow granules can deal with such incomplete and noisy data.

C. Performance of GCCM

Generally, classification models can be divided into two major categories: i) Probability based models (e.g., random forest). These models predict the probability that samples belong to a class. The output is the probability. ii) Target based models (e.g., k-means). These models directly figure out whether a sample belongs to a class or not. There are many metrics to evaluate the performance of a classification model. Note that ROC and AUC are based on probability models. Therefore, the commonly used metrics of precision, recall, and F1-score are exploited to demonstrate the performance.

First, 3000 flows are randomly selected from NJUPT, including video, audio, web browsing (WB), text communication



(a) The burst of the raw data. (b) The burst of the noisy data. (c) Handle the raw data with granules. (d) Eliminate the noise with granules.

Fig. 5. Dealing with noise by flow granules. Figs. 5(a) and 5(b) represent the burst shape of the raw data and the noisy data, respectively. The noise generates obvious deviations in burst shape. Figs. 5(c) and 5(d) show the burst shape of the raw data and the noisy data by using the technique of granular computing. These two burst shapes are basically the same. Flow granules can deal with noise.

(TC), FTP, and email, with 500 flows for each class. The 2-fold cross-validation is carried out on these flows. The final result is obtained by averaging the results of 20 runs, which is presented in Table VI. The proposed method works well for each type of traffic. The highest F1 is 97.25%, and the average F1 reaches 95.95%. Even the worst F1 is still above 94%.

Table V presents the confusion matrix of the classification results, where we have aggregated the results across all 20 runs. The small differences observed between Tables VI and V are due to the average of ratios not necessarily being equal to the ratio of sums. The ratio of video flows being identified as video is 95.43%, and the ratios of video flows being misidentified as audio, WB, TC, FTP, and email are 0.48%, 1.25%, 0.29%, 2.29%, and 0.26%, respectively. The ratio of audio flows being identified as audio is 97.14%, and the ratios of audio flows being misidentified as video, WB, TC, FTP, and email are 0.26%, 1.94%, 0.11%, 0.34%, and 0.21%, respectively. From Table V, we can also compute the frr $(= 1 - pre.)$ of video, audio, WB, TC, FTP, and email flows as 4.57%, 2.86%, 5.29%, 4.75%, 2.93%, 4.68%, respectively, and the far $(= 1 - rec.)$ for the six types of flows are 3.55%, 3.09%, 6.05%, 4.47%, 3.25%, and 4.65%, respectively. These results are consistent with the Otsu scheme given in (19), which can avoid the local worst case.

Genuinely, these sums and averages in Tables VI and V cannot reflect the differences between each run, so the 95% confidence intervals are plotted as error bars in Fig. 6 to demonstrate the differences between each run. Here, we provide the overall accuracy of GCCM in classifying video, audio, WB, TC, FTP, and email flows from different dadasets. For example, the overall F1, precision and recall for UNIBS are 95.13%, 93.53% and 96.62%, respectively; the corresponding deviations are 1.71%, 2.29% and 2.12%. It can be seen that: i) GCCM shows a stable classification performance with slight deviation. ii) When different datasets are tested as shown in Fig. 6, the classification results do not exhibit much difference. Therefore, Section V-D will not discuss the classification performance under different datasets.

D. Comparisons

We further test several state-of-the-art schemes, including FSM [10], FSIP [9], SFNN [8], and DPI [4]. Application generates traffic under specific communication protocol and transmission pattern, etc., so traffic flows always have different shapes. In [10], the fractal characteristics were used to describe the shape of the flow and thus to facilitate classification. Wu *et al.* [9] proposed the method FSIP to classify network flows, where instance purification aims to remove redundant SFs and thus obtaining an effective feature set to achieve accurate classification. In [8], Kornycky *et al.* made use of the well-known vector quantization algorithm SFNN to investigate traffic classification for encrypted WLAN data. Yun *et al.* [4] exploited the DPI, i.e., the semantic information in protocol message formats, to identify real-world network traces.

The classification results are presented in Table VI. Some methods show wonderful performance for certain classes, e.g., the F1 of SFNN for Audio is 99.35%. It can be seen that DPI, based on the payload to achieve classification, is relatively more accurate than other methods. Most of the recall values of DPI are higher than other methods. The mean F1-scores are 95.9%, 95.64%, 95.29%, 94.34%, and 97.6% for GCCM,

TABLE V
CONFUSION MATRIX (%)

Class	Video	Audio	WB	TC	FTP	Email	Pre.
Video	**9543**	48	125	29	229	26	95.43
Audio	26	**9714**	194	11	34	21	97.14
WB	102	196	**9471**	91	26	114	94.71
TC	11	18	145	**9525**	24	277	95.25
FTP	194	22	12	38	**9707**	27	97.07
Email	18	26	134	277	13	**9532**	95.32
Rec.	96.45	96.91	93.95	95.53	96.75	95.35	

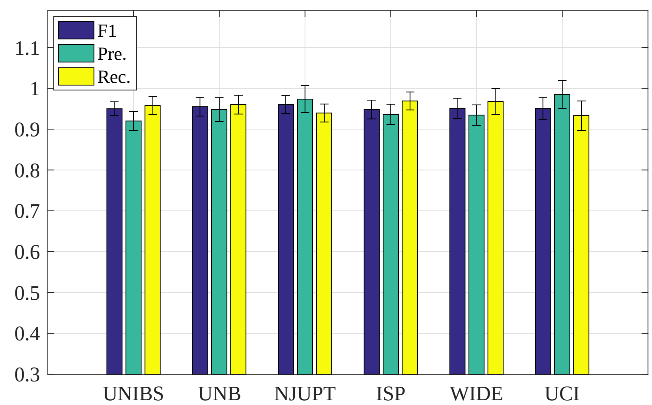


Fig. 6. Classification performance of GCCM.

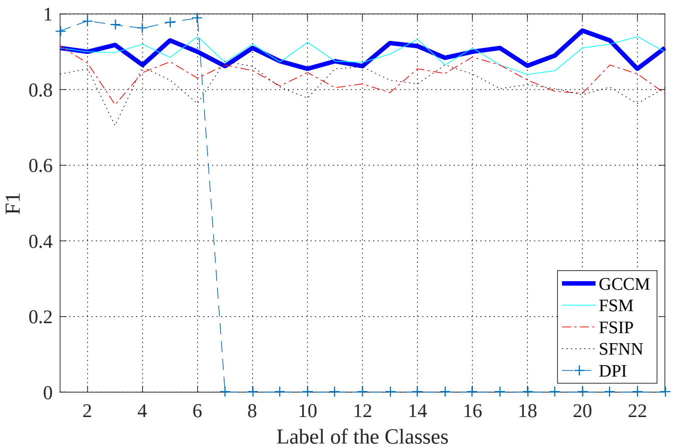


Fig. 7. Classification performance for fine classes.

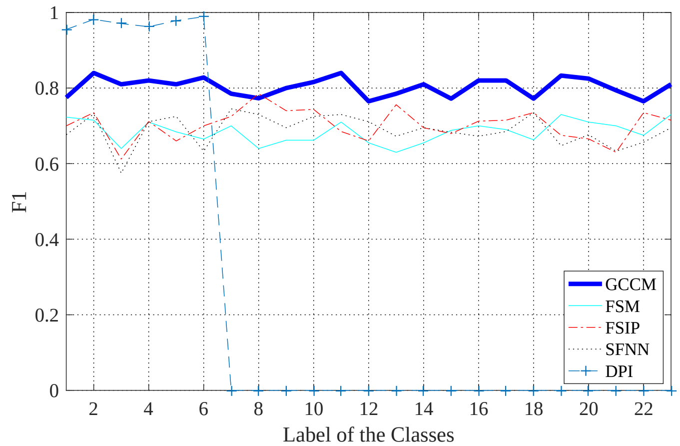


Fig. 8. Classification performance under congestion and noise.

FSM, FSIP, SFNN, and DPI respectively. In general, all these methods achieve good performance. This is mainly because there are only six coarse classes in this experiment. In next subsection, the classification performance of the five schemes for 23 fine classes will be further tested.

E. Fine Classification

In order to verify whether these schemes can adapt to fine classification, more classes, e.g., video streaming and online music (more details on the fine classes can be obtained in Table III), are randomly selected from the datasets, with 500 flows for each class. In Fig. 7, the x-axis represents the label classe and the y-axis is the F1-score. Note that F1 is the harmonic mean of precision and recall. A high F1 score indicates high precision and recall. Therefore, the precision and recall results are no longer presented in the remainder of Section V.

As shown in Fig. 7, the DPI scheme exploits the semantic information of the payload to identify traffic, and thus is relatively more accurate than other methods when classifying unencrypted traffic. However, it cannot work for encrypted flows from classes 7 to 23. The F1-scores of SFNN for the 23 classes are around 0.8, and that of FSIP is slightly higher. FSIP removes redundant SFs and thus obtains an effective feature set to achieve accurate classification. However, FSIP implements feature purification under given classes. Consequently, those extracted SFs are only effective for a specific set of classes. If the classes change, the classification system needs to be completely retrained. In contrast, the average F1 of FSM is as high as 0.9, which is comparable to that of GCCM. The fractal characteristics are different from the commonly used traditional SFs (e.g., the mean, variance, and kurtosis of packets) in that they capture the non-linear characteristics of traffic, which do not change much as the classes of flows is increased, and thus they work well in fine classification. However, FSM can only work well for flows under smooth network conditions. In dynamic network environments, especially when noise occurs, the fractal characteristics are changed, resulting in a decline in classification performance.

F. Adaptability to Variations

This subsection continues to use the flows as in Section V-E. In order to simulate network noise and congestion, we make some random adjustments of packet loss and delay for the original traces. In practice, there are two main technical reasons for packet loss rate exceeding 5%: hardware failures and network attacks. The research of this paper aims at neither hardware failure nor network attack detection. Therefore, The packet loss rate is set within 5% to simulate the variable network environment with normal congestion. Note that traffic will be interfered and varied during transmission, i.e., network noise. To simulate noisy data, we further modify and add some extra packets. In each flow, the intensity of packet modification is also controlled within 5%. Then these flows are used to test whether the above classification methods have resilience to noise and tolerance to congestion.

As shown in Fig. 8, DPI, based on the payload to achieve classification, is not affected by network dynamics (e.g., congestion and traffic noise). Therefore, DPI is more accurate than other approaches when classifying unencrypted traffic. However, it cannot work for the encrypted traffic. FSIP, SFNN and FSM present obvious decrease in F1-score. These SFs and fractal characteristics, which are obtained in a friendly network environment, does not work well in an adversarial network environment. Take video flow as an example. In a good network environment, the fractal characteristics $\tau(q)$ (q=1,2,3,4,5) are 1.395, 4.715, 5.265, 7.152, and 9.609, respectively. While in the bad network environment, they are changed to 1.203, 4.158, 5.594, 7.863 and 9.472, respectively. Actually, the fractal characteristics $\tau(q)$ for the flows are always varying under different network environments, which results in unstable classification results. The F1-scores of the proposed scheme are around 0.8, consistently higher than the scores of other baseline methods. GCCM analyzes deep into the trajectory of change for different flows, and the neighborhood granules can effectively deal with noisy and missing data. Therefore, GCCM is more suitable and robust for online classification in dynamic network environments.

TABLE VI
COARSE CLASSIFICATION RESULTS

Class	GCCM			FSM [10]			FSIP [9]			SFNN [8]			DPI [4]		
	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.
Video	95.97	94.95	96.78	96.08	96.77	95.39	95.75	96.87	94.63	**97.49**	**96.93**	98.06	95.87	93.03	**98.91**
Audio	96.78	96.63	97.02	95.35	96.67	94.11	95.87	97.55	94.26	**99.35**	**99.37**	99.34	99.27	99.19	**99.35**
WB	94.21	95.12	93.09	95.88	96.28	95.48	95.03	94.34	95.79	86.35	88.60	84.21	**98.18**	**98.97**	**97.39**
TC	95.53	95.27	96.12	94.12	93.26	94.97	94.82	93.81	95.93	98.81	**99.17**	98.45	**99.12**	98.34	**99.90**
FTP	**97.25**	97.35	**96.92**	95.63	95.53	95.82	95.54	95.42	95.66	91.69	90.23	93.20	97.07	**99.52**	94.73
Email	95.64	95.74	95.39	**96.76**	96.89	**96.63**	94.71	93.58	95.85	92.35	91.95	92.76	96.23	**98.29**	94.26

G. Time and Space Complexities

In this subsection, 1000 flows are used to evaluate the classification time. As shown in Fig. 9, it takes GCCM 1.527s for 6 classes, 1.653s for 12 classes, and 1.769s for 20 classes. It can be seen that GCCM has lower computation times than FSM, FSIP, SFNN and DPI. The results illustrated in Fig. 9 agree with the theoretical analysis in Table VII. The computation of GCCM is mainly involved in:

- Data preprocessing. Flow granules are formed in this step. According to (2)–(3) and (4)–(5), flow granules can be obtained by just scanning the flow sequence, so the computational complexity is $O(N_r)$, where N_r is the resolution of the flow sequence.
- Obtaining structure granules. Calculations of $\boldsymbol{\alpha_v}$ and $\boldsymbol{\alpha_t}$ need $O(N_r(\log m))$. Here, the observation scale is set to $m = \lceil \log N_r \rceil$ as in Section V-A, so the time complexity is $O(N_r(\log(\log N_r))) \approx O(N_r)$.
- Generating GRM. The computation required to generate the two-dimensional matrix GRM is $O((\log N_r)^2)$.
- Classifying flow. The main computation of this step is to calculate the difference degree $Dif(\boldsymbol{C}_{F_k}, \boldsymbol{C}_{P_l})$ between flow F_k and center P_l. Note that $tr(\boldsymbol{\alpha_t}\boldsymbol{\alpha_v}^{\mathrm{T}}) = \boldsymbol{\alpha_t}\boldsymbol{\alpha_v}^{\mathrm{T}}$, so the calculation of (16) is greatly simplified. The dimension of structure granules equals to the observation scale $\lceil \log N_r \rceil$, and thus the time complexity is $O(L \log N_r)$, where L is the number of classes.

Therefore, the complexity of classifying flow F_k is $O(N_r + (\log N_r)^2 + L \log N_r) \approx O(LN_r)$. Here, the overall time complexity is mainly dependent on the calculation of structure granules. Classification of N_t flows will result in computation of $O(LN_rN_t)$. In order to calculate the difference degree $Dif(\boldsymbol{C}_{F_k}, \boldsymbol{C}_{P_l})$ between flow F_k and center P_l, their corresponding GRMs need to be stored. The observation scale is set at $m = \lceil \log N_r \rceil$. Accordingly, the required space for GRMs is $O((\log N_r)^2)$. There are L centers, plus N_t flows, so the overall space complexity is $O((L + N_t)(\log N_r)^2)$.

According to the previous experiments, parameters N_s, N_t, and N_r are fixed. Here, we only pay attention to variable parameters. As shown in Table VII, the time and space complexities of GCCM and FSM depend only on L, while those of the other methods depend not only on L, but also on other factors (e.g., J and N_f). As the number of classes (i.e., L) is increased from 6 to 20, J will also increase as a result. Therefore, GCCM has the lowest time and space complexity.

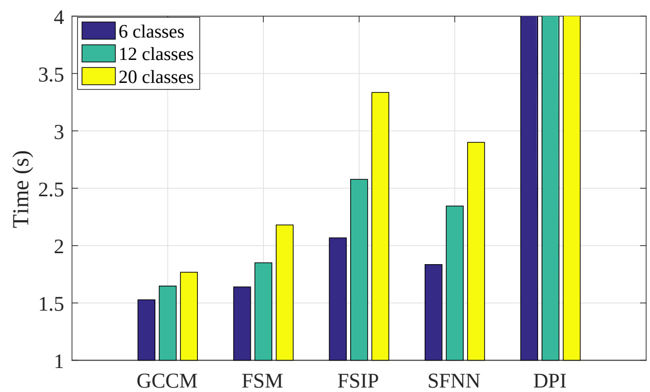


Fig. 9. Comparison of classification time.

TABLE VII
COMPARISON OF TIME AND SPACE COMPLEXITY

	Time Complexity	Space Complexity
GCCM	$O(LN_rN_t)$	$O((L+N_t)(\log(N_r))^2)$
FSM	$O(LN_r \log(N_r/s)N_t)$	$O((L+N_t)N_r \log(N_r))$
FSIP	$O(J^2LN_sN_t)$	$O(JL(N_s+N_t))$
SFNN	$O(JLN_f^2N_sN_t)$	$O(JN_f(N_s+N_t))$
DPI	$O(IKLN_tW)$	$O(KL+N_tW)$
Parameters	I: no. of iterations K: no. of keywords N_f: no. of feature values N_s: no. of sample flows S: no. of segments	J: no. of features L: no. of classes N_r: resolution of flows N_t: no. of testing flows W: no. of grams

VI. CONCLUSIONS

In this paper, we conducted an in-depth analysis of traffic classification, and found that the existing flow features are inadequate for online classification under highly varying network environments. Taking the behavior features as an example, they are based on the sequential message pattern between packets, making it challenging to work well for traffic with missing data. GRM is presented to address this issue, which included two core stages. Firstly, two types of flow granules were defined to make the model less sensitive to noise and missing data. Therefore, it can work well in poor network environments. Secondly, the spatial and temporal correlations between flow granules are explored to establish GRM, where the relationship between packets was not isolated but closely correlated. Many SFs can be treated as a special case of GRM. GRM describes the flows more comprehensively, and thus can classify the flows more accurately.

However, there are some issues that need to be further explored in the future: i) High dimensional granular relation matrix (HGRM). This paper only established a two-dimensional GRM from the perspective of time and space. We hope to explore other useful observations to build a HGRM to further improve the classification accuracy. ii) The application scope of GCCM. GCCM can be applied to a series of classification tasks, such as classification of encrypted, unencrypted, unknown, and even anomaly traffic flows as long as they have certain flow shapes. For the traffic flows that have time-varying shapes (e.g., some malware traffic), we will further explore novel flow features and design a new model to achieve good identification in our future work.

REFERENCES

[1] Z. Hu, P. Shi, J. Zhang, and F. Deng, "Control of discrete-time stochastic systems with packet loss by event-triggered approach," *IEEE Trans. Syst., Man, Cybern., Syst.,* vol. 51, no. 2, pp. 755–764, Feb. 2021.

[2] K. Xiao, S. Mao, and J. K. Tugnait, "Robust QoE-driven DASH over OFDMA networks," *IEEE Trans. Multimedia,* vol. 22, no. 2, pp. 474–486, Feb. 2020.

[3] M. U. Yaseen, A. Anjum, O. Rana, and N. Antonopoulos, "Deep learning hyper-parameter optimization for video analytics in clouds," *IEEE Trans. Syst., Man, Cybern., Syst.,* vol. 49, no. 1, pp. 253–264, Jan. 2019.

[4] 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.

[5] E. Peltonen *et al.*, "6G white paper on edge intelligence," *ArXiv e-prints,* Apr. 2020, [online] Available: https://arxiv.org/abs/2004.14850.

[6] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: a systematic survey," *IEEE Commun. Surveys Tuts.,* vol. 21, no. 2, pp. 1988–2014, Secondquarter 2019.

[7] A. M. Sadeghzadeh, S. Shiravi, and R. Jalili, "Adversarial network traffic: towards evaluating the robustness of deep-learning-based network traffic classification," *IEEE Trans. Netw. Serv. Man.,* vol. 18, no. 2, pp. 1962–1976, June 2021.

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

[9] Z. Wu, Y. Dong, H. Wei, and W. Tian, "Consistency measure based simultaneous feature selection and instance purification for multimedia traffic classification," *Comput. Netw.,* vol. 173, no. 2020, pp. 107190–107203, May 2020.

[10] P. Tang, Y. Dong, J. Jin, and S. Mao, "Fine-grained classification of Internet video traffic from QoS perspective using fractal spectrum," *IEEE Trans. Multimedia,* vol. 22, no. 10, pp. 2579–2596, Oct. 2020.

[11] H. Yao, P. Gao, J. Wang, P. Zhang, C. Jiang, and Z. Han, "Capsule network assisted IoT traffic classification mechanism for smart cities," *IEEE Internet Things J.,* vol. 6, no. 5, pp. 7515–7525, Oct. 2019.

[12] T. Zhang and S. Mao, "Energy-efficient power control in wireless networks with spatial deep neural networks," *IEEE Trans. Cogn. Commun. Netw.,* vol. 6, no. 1, pp. 111–124, Mar. 2020.

[13] R. Kumar, M. Swarnkar, G. Singal, and N. Kumar, "IoT network traffic classification using machine learning algorithms: an experimental analysis," *IEEE Internet Things J.,* vol. 9, no. 2, pp. 989–1008, Jan. 2022.

[14] P. Tang, Y. Dong, and S. Mao, "Online traffic classification using granules," in *Proc. IEEE INFOCOM WKSHPS,* Toronto, Canada, July 2020, pp. 1135–1140.

[15] A. Tongaonkar, R. Torres, M. Iliofotou, R. Keralapura, and A. Nucci, "Towards self adaptive network traffic classification," *Comput. Commun.,* vol. 56, no. 2015, pp. 35–46, Feb. 2015.

[16] Y. Wang, C. Chen, and Y. Xiang, "Unknown pattern extraction for statistical network protocol identification," in *proc. 40th IEEE LCN,* Clearwater Beach, FL, USA, Nov. 2015, pp. 506–509.

[17] J. Kampeas, A. Cohen, and O. Gurewitz, "Traffic classification based on zero-length packets," *IEEE Trans. Netw. Serv. Man.,* vol. 15, no. 3, pp. 1049–1062, Sept. 2018.

[18] L. Wang, H. Mei, and V. S. Sheng, "Multilevel identification and classification analysis of Tor on mobile and PC platforms," *IEEE Trans. Ind. Informat.,* vol. 17, no. 2, pp. 1079–1088, Feb. 2021.

[19] M. Zou, C. Wang, F. Li, and W. Song, "Network phenotyping for network traffic classification and anomaly detection," in *proc. 18th IEEE HST,* Waltham, MA, USA, May 2018, pp. 1–6.

[20] 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.

[21] H. Shi, H. Li, and Z. Dan, "An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification," *Comput. Netw.,* vol. 132, no. 2018, pp. 81–98, Feb. 2018.

[22] R. Nossenson and S. Polacheck, "On-line flows classification of video streaming applications," in *Proc. 14th IEEE NCA,* Cambridge, MA, USA, Sept. 2015, pp. 251–258.

[23] C. Thay, V. Visoottiviseth, and S. Mongkolluksamee, "P2P traffic classification for residential network," in *Proc. ICSEC'16,* Chiang Mai, Thailand, Feb. 2016, pp. 23–26.

[24] H. Shi, H. Li, D. Zhang, C. Cheng, and W. Wu, "Efficient and robust feature extraction and selection for traffic classification," *Comput. Netw.,* vol.119, no. 2017, pp. 1–16, June 2017.

[25] Z. Jin, Z. Liang, Y. Wang, and W. Meng, "Mobile network traffic pattern classification with incomplete a priori information," *Comput. Commun.,* vol. 166, no. 4, pp. 262–270, Jan. 2021.

[26] Z. Wu, Y. Dong, L. Yang, and P. Tang, "A new structure for Internet video traffic classification using machine learning," *in Proc. 4th IEEE CBD,* Lanzhou, China, Aug. 2018, pp. 322–327.

[27] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning: experimental evaluation, lessons learned, and challenges," *IEEE Trans. Netw. Service Manag.,* vol. 16, no. 2, pp. 445–458, Feb. 2019.

[28] W. Li, T. Yang, O. Rottenstreich, X. Li, G. Xie, H. Li, B. Vamanan, D. Li, and H. Lin, "Tuple space assisted packet classification with high performance on both search and update," *IEEE J. Select. Areas Commun.,* vol. 38, no. 7, pp. 1555–1569, July 2020.

[29] W. Chen, F. Lyu, F. Wu, P. Yang, G. Xue, and M. Li, "Sequential message characterization for early classification of encrypted Internet traffic," *IEEE Trans. Veh. Technol.,* vol. 70, no. 4, pp. 3746–3760, Apr. 2021.

[30] J. Garcia, T. Korhonen, R. Andersson, and F. Vastlund, "Towards video flow classification at a million encrypted flows per second," in *Proc. 32th IEEE AINA,* Cracow, Poland, May 2018, pp. 358–365.

[31] M. Shen, Y. Liu, L. Zhu, K. Xu, X. Du, and N. Guizani, "Optimizing feature selection for efficient encrypted traffic classification: a systematic approach," *IEEE Network,* vol. 34, no. 4, pp. 20–27, July 2020.

[32] W. Xu and W. Li, "Granular computing approach to two-way learning based on formal concept analysis in fuzzy datasets," *IEEE Trans. Cybern.,* vol. 46, no. 2, pp. 366–379, Feb. 2016.

[33] W. Lu, D. Shan, W. Pedrycz, L. Zhang, J. Yang, and X. Liu, "Granular fuzzy modeling for multidimensional numeric data: a layered approach based on hyperbox," *IEEE Trans. Fuzzy Syst.,* vol. 27, no. 4, pp. 775–789, Apr. 2019.

[34] W. Pedrycz, "Granular computing for data analytics: a manifesto of human-centric computing," *IEEE-CAA J. Automatic.,* vol. 5, no. 6, pp. 1025–1034, Nov. 2018.

[35] L. A. Zadeh, "Toward a generalized theory of uncertainty (GTU) - an outline," *Inform. Sciences,* vol. 172, pp. 1–40, Aug. 2005.

[36] S. K. Pal and D. B. Chakraborty, "Granular flow graph, adaptive rule generation and tracking," *IEEE Trans. Cybern.,* vol. 47, no. 12, pp. 4096–4107, Dec. 2017.

[37] B. B. Mandelbrot and J. R. Wallis, "Some long-run properties of geophysical records," *Water Resour. Res.,* vol. 5, no. 2, pp. 321–340, Apr. 1969.

[38] I. Hernandez-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.

[39] 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.

[40] Z. Wang, S. Mao, and W. Yang, "Deep learning approach to multimedia traffic classification based on QoS characteristics," *IET Netw.,* vol. 8, no. 3, pp. 145–154, May 2019.

[41] S. Tian, F. Gong, S. Mo, M. Li, W. Wu, and D. Xiao, "End-to-end encrypted network traffic classification method based on deep learning," *J. China U. Posts Telecommun.,* vol. 27, no. 3, pp. 25–34, June 2020.

[42] T. Shapira and Y. Shavitt, "FlowPic: A generic representation for encrypted traffic classification and applications identification," *IEEE Trans. Netw. Serv. Man.,* vol. 18, no. 2, pp. 1218–1232, June 2021.