Article:
Wu, Z., Dong, Y.-N., Jin, J. et al. (2 more authors) (2022) Multimedia traffic classification for imbalanced environment. IEEE Transactions on Network Science and Engineering, 9 (3). pp. 1838-1852.

https://doi.org/10.1109/TNSE.2022.3153925

Accepted for publication on 19 February 2022.

Multimedia Traffic Classification for Imbalanced Environment

Zheng Wu, Yu-ning Dong, *Member, IEEE,* Jiong Jin, *Member, IEEE*
Hua-Liang Wei, and Gaogang Xie, *Senior Member, IEEE*

***Abstract*—With ever-increasing volume and variety of multimedia traffic on the Internet, machine learning-empowered techniques nowadays tend to become indispensable for future intelligent network management. To realize automatic traffic management with Quality of Service (QoS) guarantees, there is a pressing need for accurate traffic classification. However, the inherent characteristics of networks cause imbalanced class distribution in traffic classification, which could degrade the performance of classification, especially on the minority classes. To address the issue of class imbalance in both stationary and nonstationary environments, this paper proposes a novel scheme called CHS (chain hierarchical structure) which is able to characterize class distribution from a new perspective. By building an error model, we can compute the error propagation generated by CHS and analyze the factors that affect it. More importantly, two key methods involving classifier ranking and combination with the hierarchical structure are devised to mitigate the error propagation produced by the classifier. The effectiveness of the developed framework is validated through experiments over two real-world traffic datasets in both stationary and nonstationary environments. The experimental results demonstrate that our proposed methods outperform the state-of-the-art approaches in terms of classification accuracy and running time. The proposed methods are particularly effective in the nonstationary imbalanced environment.**

***Index Terms*—Multimedia traffic classification, class imbalance, Quality of Service (QoS), chain and hierarchical structure.**

I. Introduction

In recent decades, with the advance of multimedia technology, the Internet video service has achieved fast development. As reported by Cisco, the volume of Internet video traffic will account for 82% of all consumer Internet traffic by 2022 [1]. Moreover, the emerging applications and diverse protocols make the Internet environment more complicated, leading to a series of challenges on many aspects such as achieving efficient management of networks and guaranteeing Quality of Service (QoS) of various Internet applications [2]. To tackle these problems, Internet Service Providers (ISP) and operators need more intelligent tools for the future network management. In recent years, breakthroughs have been made in artificial intelligence and machine learning such as deep networks and pattern recognition [3]. It is believed that these techniques could make network management more automatic and intelligent facing the above challenges [4].

To realize the vision of machine learning-enabled network management, an essential problem that lies in multimedia traffic classification needs to be solved first: from the QoS point of view, different video services generally require different network resources and QoS support mechanisms, so the model system should be well designed to accommodate all these differences [2]. Once the type of multimedia traffic is identified, its QoS could be better guaranteed [5]. For example, Suman *et al.* [6] proposed a real-time optimal fair packet scheduler for IP routers to maximize the QoS of multimedia traffic including Voice over Internet Protocol (VoIP), Internet Protocol Television (IPTV) and so forth. While in this scheduler system, one important premise is to implement accurate classification on these multimedia traffic. Ahmed *et al.* [7] presented an intelligent QoS management framework for multimedia based Software Defined Networks (SDNs), called LearnQoS, to provide the resource allocations to different multimedia services for optimizing end-to-end QoS. Similarly, the accurate traffic classification enable the precise resource allocations to each class of traffic. In turn, it also contributes to different QoS-aware attribute values for different multimedia services. These observations have motivated us to employ advanced machine learning methods with QoS-aware flow statistical features to classify Internet video traffic.

A. Motivation and challenges

Considering one circumstance, if the network gives different QoS classes the same resource, e.g., bandwidth, priority, delay, etc., the network is more likely to be inefficient. For example, lower-required classes (e.g., standard definition video) would occupy a considerable proportion of network resources (more than it is necessary), and it may lead to scarce resources (not enough) for higher-required classes traffic (e.g., high or ultra-definition video).

Precise traffic classification is the significant premise to conduct the differentiated service for different classes. By

This work was supported in part by the National Natural Science Foundation of China (No. 61271233) and Postgraduate Research & Practice Innovation Program of Jiangsu Province (No. KYCX18_0894). (Corresponding author: Yu-ning Dong.)

Z. Wu and Y. Dong are with the College of Telecommunications & Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (wuzheng2310174030@163.com; 19900011@njupt.edu.cn).

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

H. Wei is with the Department of Automatic Control and Systems Engineering, the University of Sheffield, Sheffield S1 3JD, U.K (w.hualiang@sheffield.ac.uk).

G. Xie is with the Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China, also with the School of Computer Science, and Technology, University of Chinese Academy of Sciences, Beijing 100049, China (xie@ict.ac.cn).

classifying the network data flow, the QoS-aware class of the data flow can be identified, and then the QoS requirements can be obtained.

However, in most network environments, some categories of service are more prevalent than others, resulting in non-uniform class distributions [8]. In addition, such an imbalanced status differs with the change of both the time and locations. Taking 2020, a special year for Covid-19, as an example, it was reported that video, gaming, and social sharing comprised over 80% of all Internet traffic and the traffic of streaming video and conference video surged most significantly under lockdown [9]. This intrinsic characteristic of Internet traffic causes imbalanced class distribution in traffic classification [8], and most standard classifiers are not capable of properly handling imbalanced datasets, and typically, the classification of the applications is performed on the basis of the traffic volume only. Moreover, the skewed distribution makes many machine learning algorithms less effective, especially in predicting the behavior and requirements of minority classes [10]. In addition, with time going on, the skewed distribution more likely makes shift. For example, a significant difference tends to be imposed on the traffic class distribution by the emerging multimedia technologies (e.g. virtual reality), important events (e.g. Covid-19 outbreak), the people's preferences and so on [11]. If the fixed model is just adopted in the changing environment, the accuracy is bound to decrease. Therefore, the imbalance classification system for Internet traffic needs to adapt to the dynamic environments.

To provide the precise traffic management, ISPs need to consider finer-grained classification of video traffic from the video resolutions. International standards (e.g., H.264 or MPEG-4) have sets of recommended video transmission rates according to different resolutions for better transmitted video quality [12]. Thus, it is necessary for service providers to classify these video traffic with different resolutions in advance to provide different resource allocations (e.g., bandwidth or priority). Besides, classifying video traffic with different resolutions is conducive to predicting the quality of experience for more focusing on the end-user experience [13]. However, this finer-grained classification is more difficult to implement due to fuzzier category boundaries [14].

B. Key contributions

This paper focuses on the classification of the imbalanced multimedia traffic. More specifically, the main contributions of the paper include:

- A novel chain structure to address the problem of imbalanced data is proposed. The theoretical error model is built to analyze the error causes to improve this structure best. Besides, we calculate the error upper bound of this structure when categories tend to be infinite. This proposed ensemble learning method is used to handle the problem of imbalanced traffic data classification whether the imbalanced degree of the environment is stationary or not.
- Importantly, to overcome the shortcomings of the associated error of this structure, a chain-hierarchical structure (CHS) is developed, involving two key techniques, i.e., heuristic classifiers ranking and the combination with hierarchical structure. Besides, to face the dynamic traffic environment, the classifiers ranking of the proposed method can adapt to the traffic imbalance changes. We also provide the quantitative measures to demonstrate the advantages of the novel method.
- A traffic classification framework for data imbalance is built covering data collection, data preprocessing, feature extraction (FE), feature selection (FS) and so forth. In addition, stationary and non-stationary mode of the proposed methods are also devised to adapt to the corresponding both environments.
- Two real-world datasets are employed for comprehensive comparisons containing both wired and mobile network traffic as well as encrypted and unencrypted traffic. Extensive experiments are carried out to validate the performances of the proposed schemes. The empirical study shows that our framework can improve the accuracy of multimedia traffic classification in both stationary and nonstationary environments.

The remainder of the paper is structured as follows. Section II discusses the related work. Section III states the problems to be solved. The framework of traffic classification is presented in Section IV. The detailed designs of classifiers based on CHS are given in Section V. Section VI presents the diagram of Internet traffic classification, experimental setup and results. Finally, conclusions are drawn and the future work is outlined in Section VII.

II. Related Work

A. Machine learning based traffic classification

Substantial efforts have been made to tackle the multimedia traffic classification problem based on machine learning techniques. Most works apply machine learning techniques to explore traffic features and improve the classification performance. For example, Goncalves *et al.* [15] used a regression-based model to divide the traffic into Peer-to-Peer (P2P) traffic and non-P2P traffic. Tang *et al.* [16] employed the fractal features to identify video traffic and good results were obtained by different machine learning based classifiers. Liu *et. al* [17] explored the in-flow behavioral features and devised a feature selection method jointly considering the traffic characteristics and the degree of feature drift. Shen *et al.* [18] leveraged the application attribute bigram to enrich the diversity of applications fingerprints and further used the second-order markov chain to complete the encrypted traffic classification.

With the proliferation of deep learning methods, researchers have recently investigated these methods for traffic classification and reported high accuracy [19]. Wang *et al.* [20] proposed a deep learning scheme, which is designed by combining RNN and CNN in a parallel way, to identify the mobile video applications. Graph neural networks (GNN) [21] is firstly used in traffic classification and obtains a high performance in identifying applications. Due to the space limitation, we cannot list all the related works here. More comprehensive reviews for multimedia traffic classification can be found in [22] and [23].

Most works classify the multimedia traffic either according to their applications, such as the video traffic from Youtube[1], iQiyi[2], etc., or according to the specific protocols used. In contrast, this study is primarily concerned with how to guarantee QoS. Therefore, video traffic is classified according to their QoS requirements.

B. Techniques dealing with the data imbalance problem

Existing machine learning techniques for imbalanced data classification problems can be generally divided into two groups: data re-sampling and cost-sensitive learning. Techniques from the first group, also known as data-level methods, usually need to rebalance the class distribution manually in the training set, so that a good classifier can be obtained equally for each class. This group includes random undersampling (RUS), random oversampling (ROS), and synthetic minority oversampling technique (SMOTE) [24]. The main advantage of these methods is that they are independent of the underlying classifier, but there are also some drawbacks, namely making original knowledge system changed to offset the class populations, further possibly making the original knowledge system lose some important information or add some redundant information. More recently, deep learning is becoming another desired tool for handling imbalanced traffic datasets. Conditional generative adversarial network (CGAN) [25], [26] is employed to yield instances of minority classes, making a balanced class distribution. This method outperforms those based on synthetic oversampling technique but takes more training time, especially, in comparison with the approaches based on ensemble learning.

Instead of manipulating data during model training, techniques from the second group generally assign higher misclassification cost to the minority class than the majority one, and then seek the minimum total cost of both classes. This type of algorithms is mainly represented by ensemble learning [27]. Brien *et al.* [28] presented a quantile classifier based on class density for imbalanced problem, which classifies samples based on whether the conditional probability of the minority class exceeds a specified threshold. Experimental results validated its competitive performance in terms of G-means. Wei and Sun *et al.* [29] also tackled the imbalanced problem in traffic classification using the proposed hybrid approach called BalancedBoost, which combines the data sampling and boosting algorithms. Compared with existing data sampling techniques over UNIBS datasets, this method presents better performance in terms of F-measure. Ng *et al.* [30] applied a technique of imbalance-reversed bagging to overcome the negative impact on imbalanced classification, improving the true positive rate while maintaining a low false positive rate through the experimental validation. Aceto *et al.* [31] proposed a multimodal deep learning framework for encrypted traffic classification. When devising the loss function, it weights instances through different categories to adjust the discrimination capacity towards some specific classes. But most of these methods usually take much time on updating the model with multiple iterations.

TABLE I
COMPARISON BETWEEN DATA IMBALANCED METHODS.

Category	Pro	Con
Data resampling	be independent of the underlying classifier	make original knowledge system changed
Cost-sensitive	directly improve the sensitivity of classifier towards minority classes	take much time on updating model

Santiago *et al.* [32] exploited the imbalanced population presented in the Internet traffic datasets to design a novel ensemble method called T-DTC by using the chain structure to achieve faster classification. Though it is proved that T-DTC has the promising performances for imbalance problems by extensive experiments, our experimental results reveal that multiple classes classified in the T-DTC method may cause serious error propagation. Inspired by T-DTC, we propose an extended model. The following comprehensive experiments validate the superiority of our method CHS over T-DTC.

Besides, there has been much effort on the research of traffic imbalance. However, these methods mostly work at the static imbalanced status and are devised for the offline condition. Once the traffic imbalance make shifts, the methods probably could not keep up with these changes. In this paper, we intend to devise a method for imbalanced multimedia traffic in both stationary and nonstationary environments and to our best knowledge overcome the shortcomings of each type of methods. We also utilize a specific set of tools for imbalanced data classification problem in the experiment to fairly assess the experimental results. For performance evaluation, the G-means [33] and the proposed minority F-measure (MF) are employed. To measure the ratio of imbalance more accurately, a measure based on the imbalance degree (ID) [34] is utilized. Concerning the validation, we apply the distribution optimally balanced cross-validation (DOB-SCV) [35] to split the training and test sets.

III. PROBLEM STATEMENT

A. Imbalanced problem

For clarity of the discussions that follow, the major mathematical symbols and abbreviations are listed in Table II.

To analyse the issue of class imbalance, we formally describe the problem as follows.

Let $\{(x_1, c_1), \cdots, (x_l, c_l), \cdots, (x_n, c_n)\}$ be the supervised training dataset, where x_l is the l-th instance, and c_l is the label of l-th instance, $l \in [1, n]$. Let $\{f_1, \cdots, f_i, \cdots, f_N\}$ be the empirical class distribution [34] denoted by η, for N classes to estimate the real class distribution, where f_i is represented by the following equation:

$$f_i = \frac{1}{n} \sum_{l=1}^{n} \varepsilon(c_l = C_i), \quad i \in [1, N], \tag{1}$$

where $\varepsilon(t) = \left\{ \begin{array}{ll} 1 & \text{, if } t \text{ is true} \\ 0 & \text{, otherwise} \end{array} \right.$ and C_i denotes the ith class ($C_i \in C$) in class set.

As stated earlier, an improvement of accuracy for minority class(es) is crucial for classification in imbalanced dataset.

[1] https://www.youtube.com/

[2] https://www.iqiyi.com/

TABLE II
NOTATIONS AND ABBREVIATIONS.

Symbol	Definition
c_l	The class of the lth instance .
d_Δ	The distance or similarity function.
n	The number of instances.
x_l	The lth instance.
C	The label class set.
C_i	The ith classifier on chain structure.
D_i	The inter-class distance of category i.
M	The minority class set.
N	The number of categories.
R_i	The cohesion ratio of category i.
S_i	The intra-class distance for category i.
T_i	The instance size of category i.
CNN	Convolutional neural network.
DT	Decision tree.
FGC	Fine-grained classification.
IG	Information gain.
ID	Imbalanced degree.
MF	Minority F1-score.
OA	Overall accuracy.
PE	Propagation error.
QoS	Quality of Service.
QRR	QoS and resource requirements.
RE	Residual error.
RNN	Recurrent neural network.
RUS&ROS	Random undersampling and Random oversampling.

In this context, minority class set contains the classes whose empirical probability is lower than the equiprobability:

$$M = \left\{ C_i \in C | f_i < \frac{1}{N}, f_i \in \eta \right\}. \quad (2)$$

The number of minority classes m can be further represented as follows:

$$m = \sum_{i=1}^{N} \varepsilon \left(f_i < \frac{1}{N} \right). \quad (3)$$

In order to measure the imbalance degree precisely in a multi-class dataset, we use the imbalanced degree (ID) [34] to assess the extent of imbalanced distribution, since it has higher resolution and measures the difficulty degree of imbalanced data classification more sensitively. The imbalanced degree is calculated as follows:

$$ID(\eta) = \frac{d_\Delta(\eta, e)}{d_\Delta(l_m, e)} + (m-1), \quad (4)$$

where e is the special case of equiprobability; l_m is the distribution showing exactly m minority classes with the longest distance to e; d_Δ is the chosen distance or similarity function Δ to instantiate *ID*. The discriminant function Δ is selected as the Hellinger distance in this paper for its better performance.

The Internet traffic imbalanced status varies as time goes by. It is worth noticing that the changes of imbalance degree is a specific type of concept drift [36], since it causes a changes in the joint probability $p(X,C) = p(C) \cdot p(X|C)$ through the term of $p(C)$ where C is the label set given the X instance set. The dynamic changes could change the inter-distributions between classes and further impact the easiness of classification for each category.

B. Classification from the QoS and Resource Requirements perspective

The paper focuses on Internet video traffic classification from the perspective of QRR (QoS and Resource Requirements). To illustrate the problem clearly, we formulate the problem of fine-grained classification for video traffic as follows.

A network video flow is a set of packets $(pkt_1, pkt_2, \cdots, pkt_n)$ within a certain time period transmitted along the path from the specific source to the specific destination in a network, where each packet is represented by a five-tuple (Time stamp, Destination IP, Source IP, Protocol, Packet length). Given a video flow, we intend to figure out which category this flow belongs to, based on some QoS-aware flow-level statistical features. Formally, the flow represented by the p-meta QoS-aware feature set $(q_1 \sim q_p)$,

$$x_l = (pkt_1, pkt_2, \cdots, pkt_n) \xrightarrow{FE} (q_1, q_2, \cdots, q_p), \quad (5)$$

where q_i is the ith QoS-aware feature and *FE* represents the process of feature extraction, which will be mapped into a suitable category c_l' by our proposed fine-grained classification framework:

$$c_l{}' \leftarrow g(x_l) = g((q_1, q_2, \cdots, q_p)), c_l{}' \in C, \quad (6)$$

where $g(x_l)$ is a mapping function. Unlike most other fine-grained classification methods attempting to identify specific applications [37] [38], the outcome of our fine-grained classification method will be some QoS classes, and each class may contain a number of applications with similar QoS requirements. The fine-grained classification results could then be used to facilitate efficient network resource management and to provide QoS support of network video services.

In most works, multimedia traffic is classified into different applications or protocols. In contrast, this paper is concerned with how to guarantee the QoS. Therefore, multimedia traffic is classified according to their QoS requirements. In the following, we define some basic level categories including streaming video (SV), HTTP download video (HDV), conversational video (CV), P2P video (P2P_video), and Internet live video (ILV). From the view of QoS, these methods have different performances due to their requirements and technical features. For example, the CV and ILV traffic are both intolerant to latency because it demands a stable and fluent experience, while in terms of symmetry for bidirectional flows, CV is more likely to be symmetric for bidirectional flow since it needs to guarantee service experience of both sides of a remote conversation; the performance of live video traffic is completely different. Concerning P2P video, each peer involved in a content delivery contributes with its own resources to the streaming session; its QoS performance hence highly depends on the number of peers, leading to unstable QoS. The traffic of HDV possesses high throughput but no

TABLE III
MULTIMEDIA TRAFFIC DATASET DESCRIPTION.

No.	Category	Application	Data size(GByte)	#Flow(k)
1.1	SD	iQIYI, Youku[2], Youtube	9.75	5.4
1.2	HD	iQIYI, Youku, Youtube	15.05	5.4
1.3	UD	iQIYI, Youku, Youtube	34.66	3.6
2	HDV	Firefox[4], Chrome[5]	67.56	1.8
3	CV	Wechat[6],QQ[7]	19.12	3.6
4	P2P_video	Xunlei[8]	57.85	1.8
5	ILV	Sopcast[9], CNTV[10]	61.91	3.6

requirement for latency. Compared to ILV and CV, SV has more lenient QoS requirements because they are not latency-sensitive and largely not jitter sensitive (because of application buffering).

To more elaborately control and manage network resources, we further classify SV into finer-grained categories in terms of different resolutions: standard definition (≤480p) (SD), high definition (720p) (HD), ultra-clear definition (≥1080p) (UD) video streaming, which have different QoS and resource requirements (e.g., bandwidth).

IV. FRAMEWORK OVERVIEW

We begin by providing an overview of the proposed traffic classification framework including data collection, feature extraction and processing, feature selection and CHS modules, as shown schematically in Fig. 1.

a) Data collection: The network flow traces are collected on the client's side by Wireshark[11] during different periods (morning, noon, evening) from Nov. 2018 to Jul. 2019 through a dedicated 100Mb/s campus network of Nanjing University of Posts and Telecommunications (NUPT). We have collected seven data categories, with each piece of raw data being represented by the 5-tuple (i.e., Timestamp, Destination IP, Source IP, Protocol, Packet length). When collecting certain application traffic, to guarantee the application generate pure flows, all other applications were shut down, and IP subnet and protocol filtering is carried out to remove dirty traffic. The specific description of the dataset is listed in Table III.

b) Feature extraction and preprocessing: A total of 40 statistical features are extracted using the continuous 100 packets (determined by multiple tests using grid search from 200 packets to 10 packets) from the collected flow traces. From the perspective of connected behavior, these flow features could be divided into three categories, comprising the uplink, downlink and datalink features. The details of the 40 features are listed in Table IV. In the framework, we utilize the z-score [23] to normalize data and discrete the features using Chi2-based discretization (the discretized interval is set to 40 through exhaustive search.) [39] to help improve the performances of classifiers [40].

TABLE V
QOS-AWARE FEATURES.

QoS	Feature	IG	PCC
Jitter	Mean of downstream interval	1.776	0.808
	Entropy of downstream packet size	1.498	-0.538
	Variance of downstream packet size	1.442	-0.517
	Mean of uplink packet size	1.436	-0.575
	Percentage of downstream to uplink bytes count	1.311	0.583
Throughput	Downstream packets rate	1.696	0.815
	Average downstream bytes rate	1.567	0.815
Delay	Mean of datalink interval	1.699	0.835
	Mean of uplink interval	1.600	0.783
	Variance of uplink interval	1.441	0.879

c) FS (Feature selection): In the extracted features, the top 10 features are selected using information gain (IG) [42]. To verify their correlations with QoS indexes, we also calculated the Pearson correlation coefficient (PCC) [43] between the features and QoS indexes (i.e., throughput, delay, jitter), and the significance test (at significance level 0.05) is conducted to ensure the validity of results. The top 10 features and their scores are listed in Table V. The results suggest that these features have a relatively strong correlation with QoS indexes. Thus the features we use in this paper are QoS-aware. After that, the proposed method FS&IP [44] is implemented to simultaneously select the optimal feature subset and purify the datasets.

d) CHS: A proposed classification structure CHS, containing a series of binary classifiers (decision tree is used in this paper), is used in this study. In Fig. 1, the structure of CHS is presented, including two key concepts of classifier ranking and hierarchical structure. Besides, facing dynamic imbalanced traffic environments, this model can make adaptive response for stable performances.

V. CLASSIFICATION

A. Classification structure and error model

This section presents the implementation details of the proposed chain structure, which is shown in Fig. 2, where $C\_i(i = 1, \cdots, N)$ denotes the i-th classifier in a chain structure. In this structure, all the classifiers apply binary classification. Each time when the data passes through a classifier, the corresponding category is classified and removed from the dataset, and the rest of the data flows into the next stage for further classification. This approach has several unique advantages:

1. This structure enables more convenient deployment, requiring less classifiers. Specifically, the chain structure needs $(N-1)$ classifiers for N-variant classification.
2. Unlike most multi-class classifiers, it removes the data classified by pre-classifier in advance without training the duplicate data, it is thus able to achieve faster identification as the process goes on.

[1] http://www.iqiyi.com/

[2] https://www.youku.com/

[3] https://www.youtube.com/

[4] http://www.firefox.com.cn/

[5] https://www.google.cn/intl/zh-CN/chrome/

[6] https://weixin.qq.com/

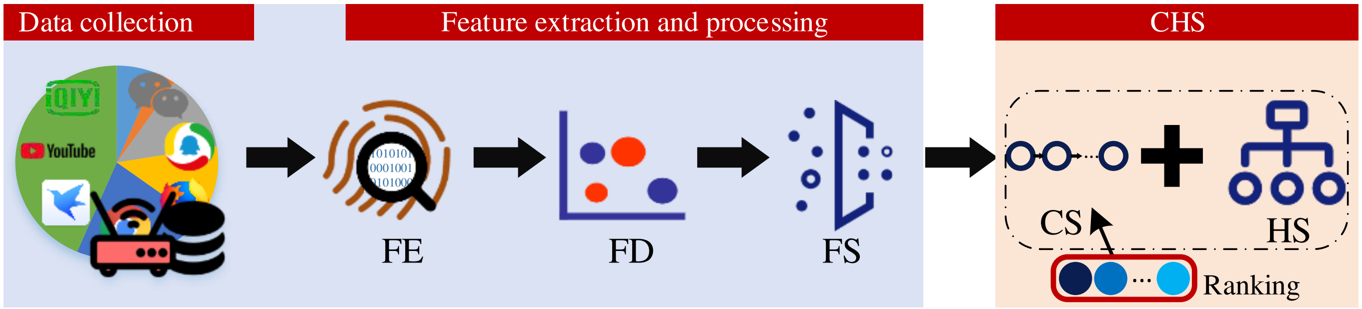
[7] https://im.qq.com/

[8] https://www.xunlei.com/

[9] http://www.sopcast.com/

[10] http://www.cctv.com/

[11] https://www.wireshark.org/



FE: Feature Extraction; FD: Feature Discretization; FS: Feature Selection; CS: Chain Structure; HS: Hierarchical Structure.

Fig. 1. Internet video classification framework

TABLE IV
STATISTICAL FEATURES USED IN THIS PAPER.

Direction	Statistical features
Downlink	Mean, variance, Maximum, minimum, weight and information entropy of downstream packet size; Number of downstream subflows;* Number of downstream valid IP address; Mean, variance of TCP window size over number of downstream packets; Mean, variance, maximum, minimum and information entropy of interval between downstream packets;
Uplink	Mean, variance, maximum, minimum of upstream packet size; Mean, variance, maximum, minimum of interval between upstream packets; Number of upstream subflows; Number of upstream valid IP address; Mean, variance of TCP window size over number of upstream packets;
Datalink	Average downstream packet rate and byte rate; Mean, variance and maximum packet size for the whole datalink; Mean, variance and maximum interval between packets for whole datalink; Ratio of downstream interval to upstream interval; Ratio of downstream TCP windows size to upstream TCP windows size; Average window size over number of packets for whole datalink; Percentage of downstream bytes to upstream bytes; Percentage of downstream packets to upstream packets;

* The subflow refers to the serial packets with the same source IP or destination IP addresses [41].

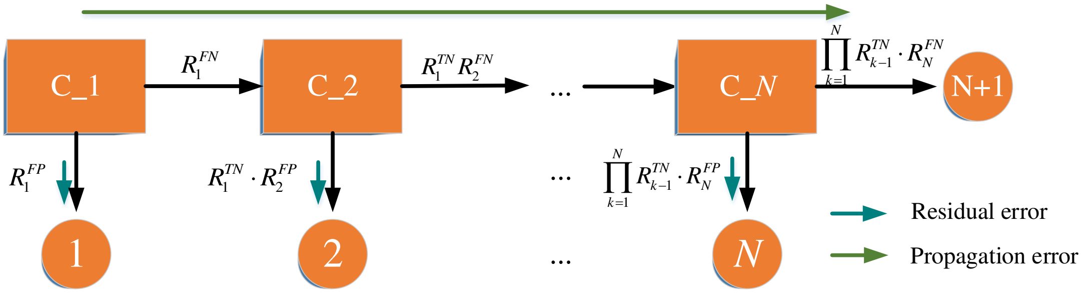

C_1
C_2
C_N
N+1
R_1^{FN}
$R_1^{TN} R_2^{FN}$
$\prod_{k=1}^{N} R_{k-1}^{TN} \cdot R_N^{FN}$
R_1^{FP}
$R_1^{TN} \cdot R_2^{FP}$
$\prod_{k=1}^{N} R_{k-1}^{TN} \cdot R_N^{FP}$
1
2
N
...
Residual error
Propagation error

Fig. 2. The chain structure and classification error propagation

3. It is noticeable that the pre-classifier can change the class distribution of the post-classifier naturally and automatically, rather than manually, which results in a distinct advantage when dealing with imbalanced datasets.

However, this structure still bears serious error propagation, which motivates us to build the error model of the classification structure in order to precisely analyze this propagation error.

To simplify the description, we define the true positive ratio R_i^{TP} as the ratio of the number of the true positive instances to the number of all instances for classifier C_i, that is,

$$R_i^{TP} = \frac{TP_i}{TP_i + TN_i + FP_i + FN_i}, \tag{7}$$

where TP_i, TN_i, FP_i and FN_i represent the true positive, true negative, false positive and false negative instances of classifier C_i, respectively. We can define additional ratios R_i^{TN}, R_i^{FN} and R_i^{FP} of true negative, false negative and false positive instances, respectively, in an analogous manner. Given that each classifier does a binary One-vs-Rest classification in the chain structure, the identified category is specified as positive class, and the remaining categories are marked as the negative class. Then, the error predicting model can be built as follows.

Each single classifier makes a contribution to the overall classification error, and each individual classifier has two error sources: the residual error (RE) and the propagation error

(PE), where RE remains in identified categories while PE will further affect the posterior classifiers as shown in Fig. 2. For the K-th ($K \in [1, N]$) classifier, the errors can be described as:

$$\begin{aligned} RE &= R_K^{FP} \cdot \prod_{k=1}^{K} R_{k-1}^{TN}, \\ PE &= R_K^{FN} \cdot \prod_{k=1}^{K} R_{k-1}^{TN}, \end{aligned} \tag{8}$$

where ${R_0}^{TN}$ is set to one. Given an $N+1$-meta classification problem, the errors produced by all classifiers are graphically shown in Fig. 2. The overall predicted error is obtained by adding errors of all classifiers:

$$\begin{aligned} Error = n \cdot \Bigg(\sum_{i=1}^{N} \Bigg[& R_i^{FP} \prod_{k=1}^{i} R_{k-1}^{TN} \\ & + R_i^{FN} \prod_{k=1}^{i} R_{k-1}^{TN} \Bigg] \Bigg). \end{aligned} \tag{9}$$

As $R_i^{FP} + R_i^{FN} = ER_i$, where ER_i is the error rate of one classifier, the overall error rate of the structure (ER) can be simplified to

$$ER = \sum_{i=1}^{N} \left[ER_i \cdot \prod_{k=1}^{i} R_{k-1}^{TN} \right]. \tag{10}$$

This results for an ($N+1$)-class classification problem through this procedure results in an N-term polynomial. From the theoretical model, we could observe the following points:

1. From Eq. (10), when N increases, ER becomes larger, indicating that the chain structure becomes less accurate with the increase of identified classes.
2. It is worth noting that, given the relative position of classifiers in the chain structure, the classifiers in the front would have more effect on the performance of structure than the later ones. This can be ascertained by checking individual terms of Eq. (10). For instance, the generative error by C_1 is ER_1, while the generative error of C_2 is $ER_2 \cdot R_1^{TN}$, with R_1^{TN} ranging from $[0, 1]$. Assuming C_1 and C_2 have equal performance, namely $ER_1 = ER_2$, C_1 would produce a higher error than C_2. Thus it is significant to rank these base classifiers properly to best mitigate the error propagation.
3. In general, the i-th term in Eq. (10) would be multiplied by $\prod_{k=1}^{i} R_{k-1}^{TN}$ on account of data transfer to the post-classifiers in each stage of the classifier. Hence, reduction of R_i^{TN} for each classifier is beneficial for the overall accuracy. This could be achieved by increasing the number of positive instances and decreasing the number of negative instances in each classifier which leads to a reduction of the amount of data sent backwards.

B. The boundary of error in chain structure

With N increasing, the errors produced by the chain structure also rise. Thus we need to explore further whether this produced error has a boundary, and if a boundary exists, what the boundary approximate.

Now, consider a special case of the error generative model Eq. (10). Assuming that all the classifiers have the same R^{TP} and R^{TN} performance, i.e., $ER_i = \alpha$ and $R_i^{TN} = \beta$, with $i = 1, 2 \cdots$, the error generative model Eq. (10) can be rewritten as:

$$ER = \sum_{i=1}^{N} [\alpha \cdot \prod_{k=1}^{i} \beta] = \sum_{i=1}^{N} [\alpha \cdot \beta^i] = \alpha\beta \cdot \frac{1 - \beta^N}{1 - \beta} \tag{11}$$

Note that for real video traffic systems, $R_i^{TN} = \beta < 1$, this is because if $R_i^{TN} = \beta = 1 (i = 1, 2, \cdots)$, then all the values of true negative and false positive instances for all classifiers are zero, this is an impossible case in practice applications. With $\beta < 1$, when the number of classes N is large, Eq. (11) recuses to:

$$ER = \frac{\alpha \cdot \beta}{1 - \beta} \tag{12}$$

This suggests that the overall error rate of the structure converges to a limited value given by Eq.(12).

C. Combination with a hierarchical structure

As explained in the previous subsection, classification with too many categories results in serious error propagation; this issue could be alleviated by reducing the length of chain structure. To reduce its length and achieve the desired level of fine-grained classification, a priori knowledge (e.g., the density of distribution or the distances) of distribution could be exploited to cluster these finer-grained categories into basic categories. Thus we define an inter-class distance matrix as below:

$$d_{ij} = d_{\Delta}(B_i, B_j), i, j \in C, \tag{13}$$

where B_i denotes the centroid vector of class i and d_{Δ} represents a measure of similarity, e.g., distance.

In this paper, the inter-class Euclidean distances are calculated according to their attributes as in Fig. 3. It is observed that the distances between SD, HD, and UD are smaller than other basic categories.

Basic classes are put to the chain structure first, then they are refined, and consequently classified by hierarchical structure into finer-grained classes, if they could be further divided into subclasses. This process also mimics the thinking mode of human beings: thinking of one thing first from the overall perspective, and then focusing on finer local parts. For example, we put SD, UD and HD as a common parent class SV in advance, because they belong to the same basic category and have similar feature patterns. In this manner, the length of structure is reduced.

D. Classifier ranking

According to the above analysis, how to rank these classifiers is a critical issue for mitigating the error propagation. The most accurate method to rank classifiers would be to test all the permutation modes (with N-class classification, there

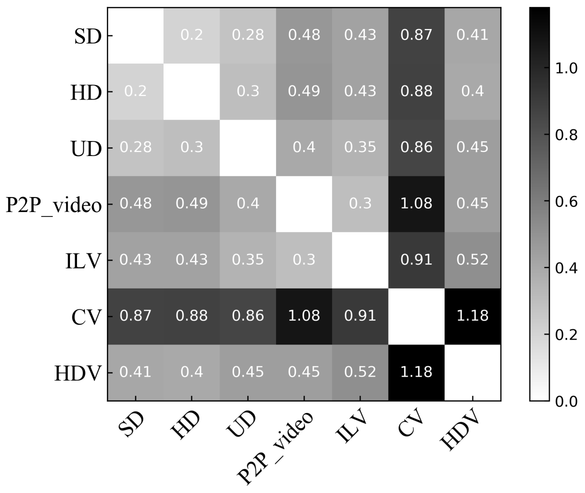


Fig. 3. Inter-class distances between seven video categories.

are a total of $N!(=1 \times 2 \times \cdots \times N)$ modes and select the best mode to organize the chain structure. But if the number of classes is large, it becomes impossible to exhaust and test all the possible permutation modes. Therefore, it is necessary to devise a method to balance the efficiency and accuracy. As each binary classifier corresponds to a category, classifiers ranking amounts to categories ranking. In the following discussions, we will provide a unified description of the two concepts, and propose three methods to rank the classifiers.

1) Accuracy-based method (AbCHS): One method to rank classifiers is based on their accuracy performances. As each classifier is binary in the chain structure, we can get the average accuracy of each classifier with the one-vs-rest training mode and n-fold cross validation. Then the classifiers are ranked by the average accuracy, i.e., the classifiers with higher accuracy are placed at the front position in the chain structure.

2) Instance ratio-based method (IRbCHS): In addition to accuracy, we need to reduce the amount of data flowing to post-classifiers for lowering their TNR. Inspired by these observations, we can rank the classifiers according to the instance ratio of each class. In other words, the majority classes are arranged in the front part of structure and the minority classes are put at the back of structure according to class distribution η. Besides, this approach could make the class distribution of the post-classifiers more balanced than pre-classifiers. It should be noted that if different classes have similar or even equal number of instances, accuracy can be used as an additional measure.

3) Cohesion-based method (CbCHS): To highlight the distinction between inter- and intra-class difference, we define cohesion ratio as the ratio of intra-class distance to inter-class distance. The intra-class distance S_i for class C_i is defined as follows. Given an N-class problem,

$$S_i = \frac{1}{T_i} \cdot \sum_{j=1}^{T_i} d_{\Delta}(x_j, B_i), \quad x_j \in C_i, \tag{14}$$

where T_i is the size of category C_i, x_j is the jth flow object, B_i represents the centroid of category C_i, which is calculated as $\frac{1}{T_i} \cdot \sum_{j=1}^{T_i} x_j$, and $d_{\Delta}(\cdot)$ denotes the distance or similarity function.

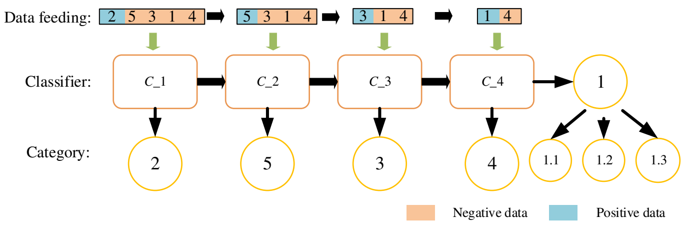


Fig. 4. An illustration of the training process. (The structure of CbCHS using in dataset WD2 is given as an example.)

The inter-class distance D_i is calculated as

$$D_i = \frac{1}{N-1} \cdot \sum_{j=1}^{N} d_{\Delta}(B_i, B_j), i, j \in N; j \neq i. \tag{15}$$

Accordingly, the cohesion ratio R_i for category C_i can be obtained as

$$R_i = \frac{S_i}{D_i}, \quad i \in N. \tag{16}$$

A smaller value of R_i indicates bigger inter-class separation and smaller intra-class cohesion. In this paper, d_{Δ} is defined as the Euclidean distance but other distance and similarity functions can also be used [34]. We plan to compare the results by utilizing different distance and similarity functions in our future work.

When comparing the three classifier ranking methods, we note that AbCHS needs to perform binary classification for each classifiers ranking, hence the effectiveness of this method depends on the performance of the classifiers. On the other hand, IRbCHS just needs to calculate the instance number of each category, but may not reflect the goodness of classifiers accurately. Finally, CbCHS does not need model training like AbCHS and therefore it is effective, because the cohesion criterion reflects the degree to which one category could be differentiated from others. The cohesion based method could also be used to evaluate the identification difficulty of each category.

E. Model training

The training procedure is specially designed and different from traditional training mode. The training process demands an order according to the rank of classifiers. Besides, due to the unique structure of CHS, before training the next classifier, the category that has been trained needs to be removed. An advantage of such a training mode is that the training process will proceed in an increasingly faster way. Thus CHS-based methods can take little time on training model to fast adapt the dynamic environments. For the hierarchical combination, the classifiers are fed with the traditional training method, which does not need instances to be removed. The training process is illustrated in Fig. 4. We summarize the overall process of CHS in Algorithm 1.

For dynamic imbalance environments, once an imbalance shift occurs, CHS-based methods launch the procedure of classifiers reranking and retrainning, as shown in Fig. 5. In

Algorithm 1 The overall procedure of organizing CHS

```
Preparation: Organize the hierarchical structure according to
    natural relationship or priori knowledge in advance;
Learning Phase:
Rank the classifiers of base classifiers
for $i$ = $1 \to N$ do
    Select features for classifier $i$
    Train the classifier $i$
    if classifier $i$ further makes hierarchical classification then
        Train the classifiers of hierarchical structures
    Remove instances of class $i$
Classification Phase:
Classify Internet flows using trained model
```

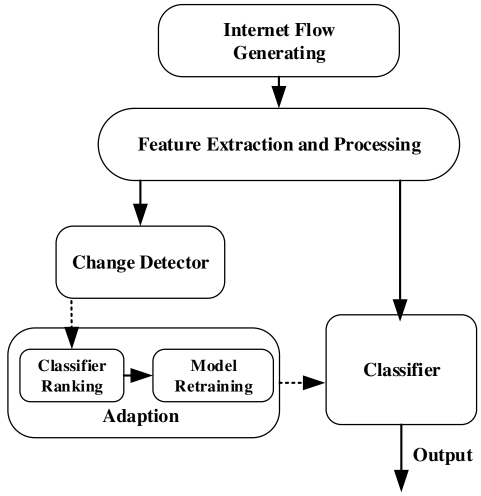


Fig. 5. The flow chart of the proposed CHS for dynamic environment (The dash lines refer to the adaption phase if the imbalance shift occurs).

general, the proposed CHS-based methods can change the order of based classifiers timely when detecting imbalanced degree changes and take little time in training model with few instances to update models. The experimental results will present the actual performances of the proposed methods in the following.

VI. Experiments

Except the part of discretization was coded with C language for fast implementation, all experimental evaluations were conducted in Python 2.7 on a computer running Windows 10 with Intel® Core™ i5-8500 3-GHZ CPU, 16GB of RAM and an NVIDIA RTX 2060 6-GB graphics card.

A. Evaluation metrics

TABLE VI
The key parameters of the compared methods

Classifier	Parameters
RFQ [28]	max_depth=None,* min_samples_split=2, n_estimators=10
BB [29]	n_estimators=10, learning_rate=1.0
CGAN [25]	batch_size=100, learning_rate=0.002, n_layer=5
MC [24]	random_state=42, max_depth = None
T-DTC [32]	base classifier=DT, rank strategy=accuracy, max_depth=None, criterion=gini

* The symbol of None* max depth denotes that nodes are expanded until all leaves are pure or until all leaves contain less than the minimum samples required to split an internal node;

To diminish overfitting, the distribution optimally balanced cross-validation (DOB-SCV) scheme is employed; such a scheme is resilient to variance shift through random selection and its strength was afterwards confirmed in [35]. Also, as some datasets have small instances, we perform a five-fold DOB-SCV with ten runs instead of the traditional ten-fold CV. Furthermore, to evaluate the performance of the proposed method comprehensively, both time- and accuracy-based evaluation metrics are employed. For time evaluation, we use training time and test time to measure the time consumption over the training and test periods, respectively. For evaluating the entire accuracy, we use overall accuracy (OA) which is defined as the ratio of correctly classified samples to all samples. For the imbalanced problem, to avoid misleading observation by the majority classes, we evaluate two other metrics, namely, G-mean (GM) [33] and minority F1-score (MF). To give more attention to the classification precision of minority classes, we propose the following measure:

$$MF=\frac{1}{m}\cdot\sum_{i=1}^{m}F1(C_i),C_i\in M,i\in[1,N], \tag{17}$$

where M represents the minority class set, and $F1(C_i)$ is a function used for measuring the F1-score of the category C_i according to classification results. MF can better reflect classifiers discriminative ability of minority classes.

B. The baseline methods

As described in Section I, ensemble algorithms and data resampling methods are able to deal with imbalanced dataset with good performance. Therefore, we make in-depth comparisons of our method with the state-of-the-art algorithms including the ensemble learning, i.e., Random Forest Quantile (RFQ) [28], BalancedBoost (BB) [29] and the resampling techniques, i.e., MSMOTE plus C4.5 (MC) [24] and CGAN [25]. Besides, the original method T-DTC [32] is also compared, where the classifier ranking adopts OA scores as the reference [32]. The base classifiers are deployed by the liarary scikit-learn[12] and the algorithm of CGAN is carried out by Pytorch[13]. The key parameters of the compared methods are presented in Table VI. Specially, through multiple tests and also for fair comparisons, CHS based methods apply the same base classifier decision tree (DT) and parameter set with the method T-DTC.

C. Dataset description

The collected dataset by ourselves through wired networks, thus called WD for short. To validate the robustness and generalization performance of the proposed method, we also make comparisons on other benchmark datasets. There are many public traffic datasets, but most of them were collected long time ago. To guarantee the validity of the experiment,

[12] https://scikit-learn.org/stable/index.html

[13] https://pytorch.org/

the recent dataset, hereafter referred to as MD for short, on mobile app traffic [45] are used.

MD is an application-level dataset with 118,020 instances in 12 categories, collected from mobile devices, i.e., QQ, WeChat, Facebook, Weibo, Youku, TencentVideo, MgTV, Browser, JdShop, VipShop, QQMail, YahooMail. All of these traffic is collected from the mobile phones installed with Android System and further extracted into 14 flow features. This dataset could be used to verify the performances of different methods in mobile traffic.

D. Classification in stationary imbalanced environments

Different network environments may have different degrees of traffic imbalances. To better understand the performance of these methods, we set up two specific traffic scenarios WD2 and WD3 with different IDs (imbalanced degrees) corresponding to the typical network environments of office and home networks respectively. WD1 is a balanced traffic dataset for comparison. Table VII lists the detailed information including instance number n, class distribution, ID and the number of minority classes m. The results of the three datasets are shown in Fig. 6, where the results of CHS are the best performances of three methods using different ranking strategies.

WD1 is a balanced dataset, thus there is no MF score in Fig. 6(a). As Fig. 6(a) shown, the proposed method have the best performance. RFQ and MC have similarly better OA and GM performances than the others. T-DTC behaves worst. It is because the multiple classes lead to serious error propagation.

From Fig. 6(b), CHS performs best in terms of OA score, and its higher MF and GM scores reveal a better-balanced classification performance over different classes because of its special structure and heuristic classifier ranking. Besides, methods BB, RFQ and MC have the roughly same scores on both OA and GM, suggesting that these three methods work well in this moderate imbalanced environment.

WD3 is a highly imbalanced dataset. The experimental results are presented in Fig. 6(c). The proposed CHS gets the highest scores on both OA and GM. Besides, resampling methods are more likely to behave better than ensemble learning at this status, indicating that the impact of calculating the costs for minority and majority classes is limited to some degree.

Mobile traffic is also verified. Among these applications, WeChat, Facebook, Weibo and QQ are social apps, so we combine them into a group called WFWQ. Since both QQMail and YahooMail are e-mail apps, we group them into a combination called QY. Similarly, TencentVideo, Youku and MgTV are all the multimedia apps, so they form a common set called TYM. Finally, JdShop and VipShop are both e-commerce platforms, which are grouped into the class called JV. Therefore, WFWQ, QY, TYM, JV and Browers are put as the basic classes in CHS. Following that, these combined groups are further classified into corresponding subclasses.

Using the Random undersampling technique on MD, we get three manual datasets with different imbalanced degrees. Table VIII lists the specific distribution. Performances comparison of different methods applied to three datasets are shown in Fig. 7.

TABLE VII
THREE MANUAL DATASETS WITH DIFFERENT DISTRIBUTIONS ON WD.

Dataset	n	Class distribution	ID	m
WD1	540	0.14/0.14/0.14/0.14/0.14/0.14/0.14	0	0
WD2	2520	0.21/0.21/0.14/0.14/0.14 /0.07/0.07	4.20	2
WD3	843	0.02/0.64/0.09/0.09/0.07/0.05/0.04	5.50	6

TABLE VIII
THREE MANUAL DATASET WITH DIFFERENT DISTRIBUTIONS ON MD.

Dataset	n	Class distribution	ID	m
MD1	16800	0.083/0.083/0.083/0.083/0.083/0.083/ 0.083/0.083/0.083/0.083/0.083/0.083	0	0
MD2	118020	0.145/0.115/0.012/0.215/0.049/0.013/ 0.119/0.216/0.034/0.039/0.012/0.030	6.490	7
MD3	80560	0.210/0.020/0.010/0.320/0.010/0.010/ 0.060/0.320/0.020/0.020/0.020/0.010	8.602	9

TABLE IX
COMPARISON OF RUNNING TIME OF DIFFERENT METHODS

Dataset / Method	WD			MD		
	Train (MS)	Test (MS)	Memory (MB)	Train (MS)	Test (MS)	Memory (MB)
CHS	**0.0038**	0.0054	1.8	**0.0192**	0.0040	**1.8**
RFQ	0.0311	0.0051	1.8	0.0287	0.0026	38.0
BB	1.1198	0.0076	2.5	2.2749	0.0138	6.2
MC	0.0226	**0.0012**	1.7	0.0866	0.0006	2.1
T-DTC	0.0202	0.0068	**0.8**	0.1554	0.0048	4.1
CGAN	6.8902	**0.0012**	978.7	3.0378	**0.0003**	1114.5

As shown in Fig. 7, similar results are obtained. Also, there are some noticeable points. With imbalance degree increasing, these schemes dealing with imbalanced problems gradually present their higher advantages over the learner not focusing on imbalanced problem, indicating it is necessary to research on the traffic imbalanced problem to help improve the classification performance. Moreover, our proposed methods can maintain the stable superiority over the other methods. The main reason is that our proposed methods have the advantages of resampling methods and cost-sensitive methods. On one hand, the unique classification structure enable the front classifiers change the distributions of the back classifiers. On the other hand, smart classifier ranking actually assigns flexible weights to categories to mitigate the negative impact of imbalanced problem.

Additionally, we record the running time of each sample that includes training and test time as well as memory usage during training models; the results are presented in Table IX from which several observations could be made. Regarding model training, CGAN takes the longest time of all approaches tested, while CHS based methods need the least time. It is because CGAN needs epochs of training to optimize the results, while our proposed methods adopt the fast training way (not needing to learn the trained data by the front classifiers regarding the back classifiers). Compared with T-DTC, CHS-based methods have better performances in terms of both training and test time. It is because they have shorter structures than that of T-DTC. Overall, CHS based methods perform better in training time, meaning that they take less time to fit the model. In terms of testing time, CGAN has the best performance.

The methods with less memory consumption is more con-

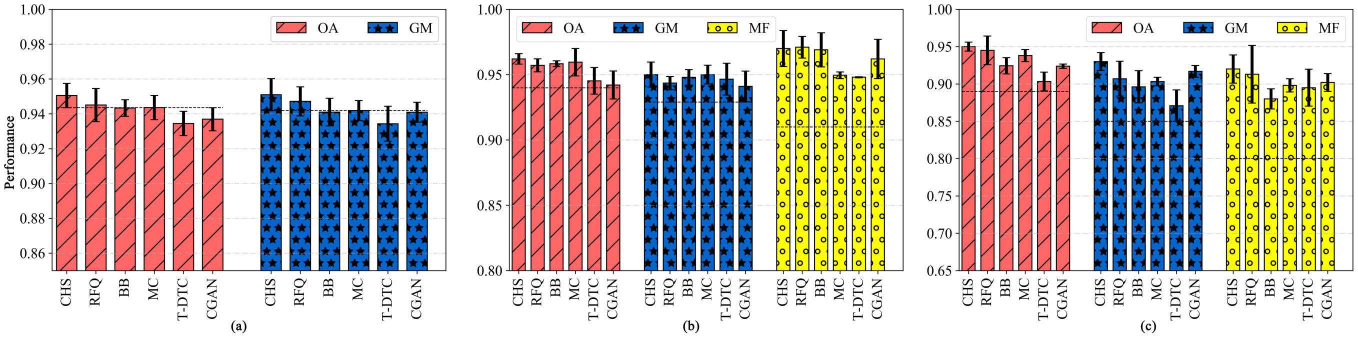


Fig. 6. Performances comparison on WD datasets (where the dotted lines represent the scores of decision tree without any imbalanced technique).

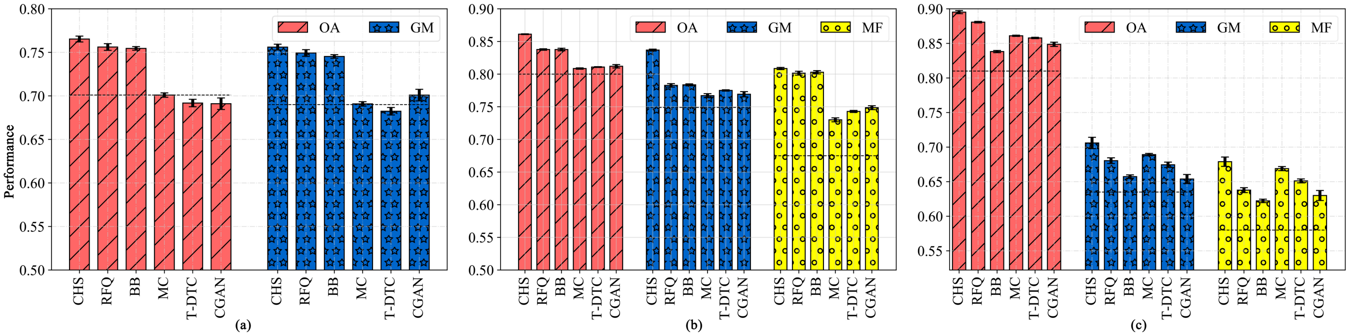


Fig. 7. Performances comparison on MD datasets (where the dotted lines represent the scores of decision tree without any imbalanced technique).

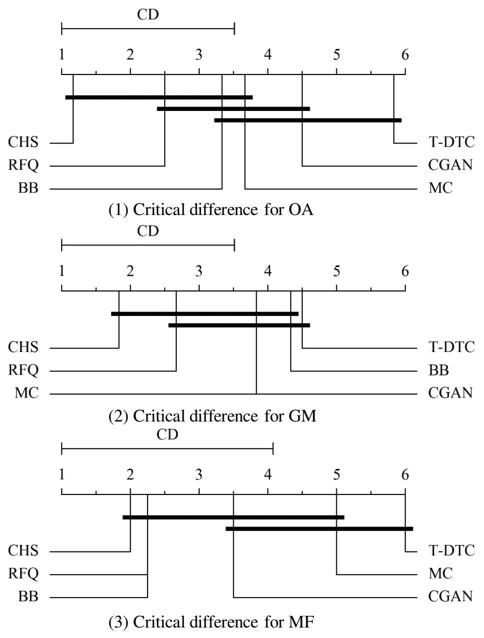
venient to be deployed in the edge devices of networks by ISPs and network operators [46]. For one classifier, much computational memory will be occupied when training model; hence the memory usage in training model is recorded. The results in Table IX prove that the proposed methods have the comparative or better performances than the other methods regarding memory utilization.

From the above comparison, several observations can be made. At first, the proposed methods CHS behave in a more stable manner and achieves a promising performance on OA, MF and GM among the 6 state-of-the-art methods, suggesting that the proposed method could manage the imbalanced problem well. Next, T-DTC performs better in the high imbalanced datasets with few categories than the opposite circumstance, since a longer chain structure will bring negative effects and lead to worse results. On the contrary, our proposed methods obtain a significant enhancement (by about 3.3% and 24% respectively on WD and MD datasets) in performances comparing with T-DTC, by employing proper arrangements for classifiers and right techniques for shortening the structure. Then, cost-sensitive and data sampling approaches also get relatively good results when dealing with imbalanced dataset, but CHS-based methods behave in a more stable manner when facing highly imbalanced condition. Finally, CHS is more likely to training models with less time than the others. It is the reason why it is suitable for online implementation. While cost-sensitive methods and resampling methods usually need much time to either update the model using multiple iterations or sample balanced instance set out of imbalance dataset. Thus most of these methods are limited in the offline mode [36].

Also, the statistical tests on these experimental results are conducted. Friedman test [47] is performed to compare the results of the proposed method and other methods on different datasets to determine whether there are statistically significant differences between the results. The superiority of the methods is ranked accordingly. Particularly, the best method is ranked at the first place, the second best one is ranked at the second place, and so on. Then, the average rank for each method is calculated. If the null-hypothesis is rejected, we proceed to Nemenyi posthoc [48] test in a pairwise manner. The performances of two methods are significantly different if their corresponding average ranks differ from the critical difference threshold $CD = q_\alpha\sqrt{\frac{k(k+1)}{6G}}$ where k is the number of methods, G is the amount of datasets, and the critical value q_α is based on the studentized range statistic divided by $\sqrt{2}$. The comparison of different methods with Nemenyi posthoc test is illustrated in Fig. 8.

The analysis reveals that the proposed method performs best in terms of each criterion. RFQ behaves at the second-best place. In addition, it is observed that T-DTC has a critical difference from the proposed method in terms of each criterion. As for other methods, their ranks are not stable regarding these criteria.

In the stationary scenario, the representatives of the resampling method and ensemble learning tend to have similar scores to the proposed CHS with respect to some criterion in the low and middle imbalanced environments. Let's have a look at the performances of different methods in the nonstationary traffic environment.



(1) Critical difference for OA

(2) Critical difference for GM

(3) Critical difference for MF

Fig. 8. Comparison of all six methods with Nemenyi posthoc test Groups of methods that are not significantly different (at $\alpha = 0.1$) are connected with bold lines.

E. *Classification in dynamic imbalanced environment*

The dynamic imbalanced environment is simulated by the three different traffic scenarios with low, medium and high imbalanced degrees in turn, in the wired networks. Each traffic scenario is composed of 3000 transmitted flows and these flows are sent to classifiers subjected to uniformed distribution for guaranteeing its stable imbalance degree. In this experiment, we focus on the performances of compared methods in the dynamic environment, ignoring the impact of change detector. Thus, we enable these methods detect the shifts of imbalance degree in time and make response to this change. The performances of different methods are presented in Fig. 9.

It is observed that they have similar performances at the initial process. However, as the imbalanced degrees of the traffic environments shift, the proposed methods behave stably at the high GM scores, displaying their superior abilities compared with the other methods. That is because of the adapted classifier order and updated model with the dynamic environment changing. The other methods just adopt the fixed model, difficult fitting the changing environments.

In general, the main advantage of the proposed CHS-based methods lie in their hybrid characteristic of the resampling methods and cost-sensitive methods. First, the pre-classifiers could change the class distribution for post-classifiers naturally, avoiding classifier overfitting for manual resampling instances. Besides, CHS based methods actually assign the weights to different categories through classifier ranking to improve the performances in imbalance environments, where the proposed methods are able to save much time taken in training models like cost-sensitive methods.

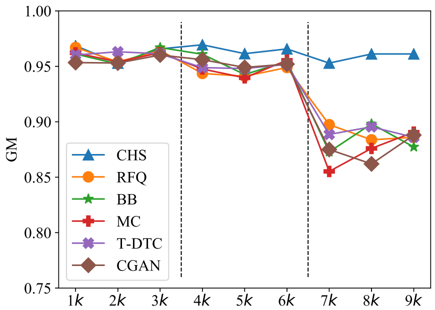


Fig. 9. The performances in the dynamic imbalanced environment.

TABLE X
INTERNAL COMPARISON OF THREE PROPOSED METHODS

	AbCHS	CbCHS	IRbCHS	Ave.*
WD1	0.948 (5,2,3,1,4)**	**0.954** (2,5,3,4,1)	0.948 (3,2,1,4,5)	0.932
WD2	0.959 (5,2,3,4,1)	**0.964** (2,5,3,1,4)	0.950 (1,3,5,4,2)	0.921
WD3	0.951 (5,2,3,1,4)	**0.950** (2,5,3,1,4)	0.940 (1,3,2,4,5)	0.903
Time	0.325s	0.242s	**0.023s**	

* Ave. means the average OAs of all the permutations of base classifiers except the recommended permutations by the proposed methods;

** Classifiers order (classifier is represented by the number in Table III).

F. *Internal comparisons of the proposed methods*

As stated above, the best method is to pick the best classifiers ranking from all the permutations in terms of classifiers ranking, but this way has extremely high complexity. While the proposed heuristic ranking strategies can give the proper suggestions despite probably not the best ranking scheme all the time. To validate these three methods, we list the scores of three methods and the corresponding order of classifiers in Table X, where the average time of ranking classifiers by the proposed methods is also recorded.

From Table X, we can observe that the proposed methods are apparently better than the other permutations of classifiers, validating the effectiveness of our methods. AbCHS has a similar rank to CbCHS generally, but there are changes in individual positions. That is because they measure the easiness of classification from different views. IRbCHS seems to be the fastest one in terms of ranking time among CHS-based methods since it just needs to sort the number of each class without other computations in general. In terms of OA and stability, CbCHS may be the better choice as it could measure the goodness from the cohesion of categories for classifiers.

G. *Open-world evaluation*

In this subsection, we evaluate the proposed method with a more realistic scene. The following problem aims at distinguishing non-multimedia and multimedia traffic; thus it is a binary classification problem. If a traffic flow could be identified as multimedia traffic, this flow could further be classified by the proposed model.

In this experiment, we randomly select an average of 1000 flows for each non-multimedia traffic (including email, FTP

TABLE XI
EVALUATION IN THE OPEN-WORLD ENVIRONMENT

Method	GM	Memory (MB)	Time (S)
CHS	0.98	**0.2377**	**0.129**
RFQ	**0.99**	0.4141	0.131
BB	0.97	0.3633	0.131
MC	0.98	1.3279	**0.129**
T-DTC	0.98	**0.2377**	**0.129**
CGAN	**0.99**	979.58	44.48

and some control signals) and obtain a total of 3000 non-multimedia flows. Besides, the multimedia flows are collected with the equal number of the non-multimedia traffic.

To realize the open-world classification, the CHS-based methods just need to make the extension by adding one base classifier in the beginning of this structure. While the other methods need to add the same classifier applying in the multimedia traffic to identify the non-multimedia traffic. Five-fold cross-validation is conducted in this experiment. The average results of these methods are presented in Table XI in terms of the GM scores and the memory usage taken in training models.

From an accuracy viewpoint, it is noticed that all these methods have similar high performances in terms of GM since multimedia traffic has a significant difference from the non-multimedia counterpart regarding the flow features. In terms of memory occupation, the proposed methods and T-DTC take the least cost. In other words, when ISPs or network operators intend to use these methods to realize the traffic classifications, our methods have the minimum memory requirements for the deployed devices. As for the easiness of deployment, the proposed methods need to add one base classifier into the head of the structure. It is undoubtedly more convenient and lightweight for deployment than other methods. As for time consuming, T-DTC, MC, and the proposed methods take less time than the others on the training model.

VII. CONCLUSION

With the growing diversity and volume of video traffic, the classification of multimedia traffic which contains several highly imbalanced classes becomes more and more important. We have described and evaluated an intelligent framework that achieves higher accuracy and better classification stability on multimedia traffic with imbalanced dataset. To mitigate the impact of class imbalance to classifiers, we propose to use a chain structure and analyze its error model theoretically. Using the results of this analysis, two techniques are developed including classifier ranking and combination with hierarchical structure. Furthermore, to obtain better ranking performance, three strategies, namely, accuracy, instance ratio and cohesion-based methods, were introduced.

The performances of the proposed methods are compared with state-of-the-art approaches in both stationary and non-stationary environments. Our experimental results clearly indicate the superiority of the proposed methods over the compared state-of-the-art methods in dealing with both stationary and nonstationary imbalanced datasets. Besides, the proposed methods are proved light-weight for their less memory usage and training time consumption.

Still, our work can be extended and improved by the following ways:

1. using different distance or similarity measures for CbCHS to probably get a better result;
2. explore other machine learning-based classifiers as base classifiers to get a better result;
3. expanding our method toward recognition of different or new applications or QoS classes.

REFERENCES

[1] Cisio, "Cisio White Paper," Website, 2019, https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html.

[2] 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 Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1988–2014, 2018.

[3] A. DAlconzo, I. Drago, A. Morichetta, M. Mellia, and P. Casas, "A survey on big data for network traffic monitoring and analysis," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 800–813, 2019.

[4] T. Shapira and Y. Shavitt, "Flowpic: A generic representation for encrypted traffic classification and applications identification," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1218–1232, 2021.

[5] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, "A big data-enabled hierarchical framework for traffic classification," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2608–2619, 2020.

[6] S. Paul and M. K. Pandit, "A QoS-enhanced intelligent stochastic real-time packet scheduler for multimedia IP traffic," *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 12 725–12 748, 2018.

[7] A. Al-Jawad, P. Shah, O. Gemikonakli, and R. Trestian, "LearnQoS: A Learning Approach for Optimizing QoS Over Multimedia-Based SDNs," in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2018, pp. 1–6.

[8] S. E. Gómez, L. Hernández-Callejo, B. C. Martínez, and A. J. Sánchez-Esguevillas, "Exploratory study on class imbalance and solutions for network traffic classification," *Neurocomputing*, vol. 343, pp. 100–119, 2019.

[9] Sandvine, "Sandvine White Paper," Website, 2021, https://www.sandvine.com/resources.

[10] X. Ma and W. Shi, "Aesmote: Adversarial reinforcement learning with smote for anomaly detection," *IEEE Transactions on Network Science and Engineering*, 2020.

[11] A. Feldmann, O. Gasser, F. Lichtblau, E. Pujol, I. Poese, C. Dietzel, D. Wagner, M. Wichtlhuber, J. Tapiador, N. Vallina-Rodriguez *et al.*, "Implications of the COVID-19 Pandemic on the Internet Traffic," in *Broadband Coverage in Germany; 15th ITG-Symposium*. VDE, 2021, pp. 1–5.

[12] T. Laude, Y. G. Adhisantoso, J. Voges, M. Munderloh, and J. Ostermann, "A comprehensive video codec comparison," *APSIPA Transactions on Signal and Information Processing*, vol. 8, 2019.

[13] Y. Zhu, S. C. Guntuku, W. Lin, G. Ghinea, and J. A. Redi, "Measuring individual video QoE: A survey, and proposal for future directions using social media," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 2s, pp. 1–24, 2018.

[14] W. Qiu, G. Chen, K. N. Nguyen, A. Sehgal, P. Nayak, and J. Choi, "Category-based 802.11 ax target wake time solution," *IEEE Access*, vol. 9, pp. 100 154–100 172, 2021.

[15] G. D. Gonçalves, Í. Cunha, A. B. Vieira, and J. M. Almeida, "Predicting the level of cooperation in a Peer-to-Peer live streaming application," *Multimedia Systems*, vol. 22, no. 2, pp. 161–180, 2016.

[16] P. Tang, Y. Dong, J. Jin, and S. Mao, "Fine-Grained Classification of Internet Video Traffic From QoS Perspective Using Fractal Spectrum," *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2579–2596, 2019.

[17] Z. Liu, R. Wang, N. Japkowicz, Y. Cai, D. Tang, and X. Cai, "Mobile app traffic flow feature extraction and selection for improving classification robustness," *Journal of Network and Computer Applications*, vol. 125, pp. 190–208, 2019.

[18] M. Shen, M. Wei, L. Zhu, and M. Wang, “Classification of encrypted traffic with second-order markov chains and application attribute bigrams,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1830–1843, 2017.

[19] Y. C. Chen, Y. J. Li, A. Tseng, and T. Lin, “Deep learning for malicious flow detection,” 2018.

[20] X. Wang, S. Chen, and J. Su, “App-net: A hybrid neural network for encrypted mobile traffic classification,” in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 424–429.

[21] M. Shen, J. Zhang, L. Zhu, K. Xu, and X. Du, “Accurate decentralized application identification via encrypted traffic analysis using graph neural networks,” *IEEE Transactions on Information Forensics and Security*, vol. PP, no. 99, pp. 1–1, 2021.

[22] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, “Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges,” *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 445–458, 2019.

[23] 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 Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1988–2014, 2018.

[24] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, “SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary,” *Journal of Artificial Intelligence Research*, vol. 61, pp. 863–905, 2018.

[25] P. Wang, S. Li, F. Ye, Z. Wang, and M. Zhang, “Packetcgan: Exploratory study of class imbalance for encrypted traffic classification using cgan,” in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–7.

[26] X. Xia, R. Togneri, F. Sohel, and D. Huang, “Auxiliary classifier generative adversarial network with soft labels in imbalanced acoustic event detection,” *IEEE Transactions on Multimedia*, vol. 21, no. 6, pp. 1359–1371, 2018.

[27] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, “A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2011.

[28] R. OBrien and H. Ishwaran, “A random forests quantile classifier for class imbalanced data,” *Pattern Recognition*, vol. 90, pp. 232–249, 2019.

[29] H. Wei, B. Sun, and M. Jing, “Balancedboost: A hybrid approach for real-time network traffic classification,” in *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*, 2014, pp. 1–6.

[30] W. W. Y. Ng, J. Zhang, C. S. Lai, W. Pedrycz, L. L. Lai, and X. Wang, “Cost-sensitive weighting and imbalance-reversed bagging for streaming imbalanced and concept drifting in electricity pricing classification,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1588–1597, 2019.

[31] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapè, “MIMETIC: Mobile encrypted traffic classification using multimodal deep learning,” *Computer Networks*, vol. 165, p. 106944, 2019.

[32] S. E. Gómez, B. C. Martínez, A. J. Sánchez-Esguevillas, and L. H. Callejo, “Ensemble network traffic classification: Algorithm comparison and novel ensemble scheme proposal,” *Computer Networks*, vol. 127, pp. 68–80, 2017.

[33] Y. Sun, M. S. Kamel, and Y. Wang, “Boosting for learning multiple classes with imbalanced class distribution,” in *Sixth International Conference on Data Mining (ICDM’06)*. IEEE, 2006, pp. 592–602.

[34] J. Ortigosa-Hernández, I. Inza, and J. A. Lozano, “Measuring the class-imbalance extent of multi-class problems,” *Pattern Recognition Letters*, vol. 98, pp. 32–38, 2017.

[35] J. G. Moreno-Torres, J. A. Sáez, and F. Herrera, “Study on the impact of partition-induced dataset shift on k-fold cross-validation,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1304–1312, 2012.

[36] S. Wang, L. L. Minku, and X. Yao, “A systematic study of online class imbalance learning with concept drift,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4802–4821, 2018.

[37] B. Park, J. W. Hong, and Y. J. Won, “Toward fine-grained traffic classification,” *IEEE Communications Magazine*, vol. 49, no. 7, pp. 104–111, 2011.

[38] Z. Zou, J. Ge, H. Zheng, Y. Wu, C. Han, and Z. Yao, “Encrypted traffic classification with a convolutional long short-term memory neural network,” in *2018 IEEE 20th International Conference on High Performance Computing and Communications*. IEEE, 2018, pp. 329–334.

[39] C.-T. Su and J.-H. Hsu, “An extended chi2 algorithm for discretization of real value attributes,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, pp. 437–441, 2005.

[40] J. Zhang, X. Chen, Y. Xiang, W. Zhou, and J. Wu, “Robust network traffic classification,” *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1257–1270, 2014.

[41] Y. Dong, J. Zhao, and J. Jin, “Novel feature selection and classification of Internet video traffic based on a hierarchical scheme,” *Computer Networks*, vol. 119, pp. 102–111, 2017.

[42] B. Azhagusundari and A. S. Thanamani, “Feature selection based on information gain,” *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 2, no. 2, pp. 18–21, 2013.

[43] M. M. Mukaka, “A guide to appropriate use of correlation coefficient in medical research,” *Malawi Medical Journal*, vol. 24, no. 3, pp. 69–71, 2012.

[44] Z. Wu, Y. n. Dong, H. L. Wei, and W. Tian, “Consistency measure based simultaneous feature selection and instance purification for multimedia traffic classification,” *Computer Networks*, vol. 173, p. 107190, 2020.

[45] R. Wang, Z. Liu, Y. Cai, D. Tang, J. Yang, and Z. Yang, “Benchmark data for mobile app traffic research,” in *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2018, pp. 402–411.

[46] L. Li, K. Ota, and M. Dong, “DeepNFV: A lightweight framework for intelligent edge network functions virtualization,” *IEEE Network*, vol. 33, no. 1, pp. 136–141, 2018.

[47] D. Haralayya, P. Aithal *et al.*, “Analysis of bank productivity using panel causality test,” *Journal of Huazhong University of Science and Technology*, vol. 50, no. 6, pp. 1–16, 2021.

[48] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

Zheng Wu received his Ph.D degree from Nanjing University of Posts & Telecommunications (NUPT). He is currently a Postdoc with the Computer Network Information Center (CNIC), Chinese Academy of Sciences (CAS). His research interests include multimedia communications, end-to-end QoS provisioning and network traffic identification.

Yu-ning Dong (M’07) received his Ph.D degree from Southeast University in Electrical Engineering, and M.Phil degree in Computer Science from The Queens University of Belfast (QUB). He is currently a professor with the College of Communications and Information Engineering, Nanjing University of Posts & Telecommunications(NUPT). He has authored/coauthored over 200 papers in IEEE and other technical journals and referred conference proceedings. He was a British Council postdoctoral fellow at Imperial College London, 1992-93; a visiting scientist at University of Texas, 1993-95; and a research fellow at QUB and the University of Birmingham, 1995-98. His research interests include wireless networking, multimedia communications and network traffic identification.

Jiong Jin (M11) received the B.E. degree with First Class Honours in Computer Engineering from Nanyang Technological University, Singapore, in 2006, and the Ph.D. degree in Electrical and Electronic Engineering from the University of Melbourne, Australia, in 2011. From 2011 to 2013, he was a Research Fellow in the Department of Electrical and Electronic Engineering at the University of Melbourne. He is currently an Associate Professor in the School of Science, Computing and Engineering Technologies, Swinburne University of Technology, Melbourne, Australia. His research interests include network design and optimization, edge computing and networking, robotics and automation, and cyber-physical systems and Internet of Things as well as their applications in smart manufacturing, smart transportation and smart cities.

Hua-Liang Wei received the Ph.D. degree in the Department of Automatic Control and Systems Engineering, the University of Sheffield, UK, in 2004. Dr Wei is currently a Senior Lecturer with the Department of Automatic Control and Systems Engineering (ACSE), the University of Sheffield, Sheffield, U.K. He is head of the laboratory of Dynamic Modelling, Data Mining and Decision Making. He previously held academic positions (Assistant Professor, Lecturer and Associate Professor, 1992 - 2000) at the Beijing Institute of Technology, China; he joined the department of ACSE in 2004 initially as a Senior Research Fellow immediately after the completion of the PhD study. His research interests include nonlinear system identification, machine learning, computational intelligence, data-driven modelling, and data mining, fault diagnosis, with applications in many multidisciplinary domains.

Gaogang Xie is a Professor at Computer Network Information Center (CNIC), Chinese Academy of Sciences (CAS), and School of Computer Science and Technology, University of Chinese Academy of Sciences (UCAS). His research interests include Internet architecture and systems.