Time-Aware Machine Learning-based Traffic QoS Classification

Weichang Zheng[1,a) Ziyu Guo[1,b) Yongbing Zhang[1,c)

Abstract: With the rapid development and popularization of the Internet and communication technologies, the amount of network traffics has grown explosively. Network resources should be allocated to the applications depending on their requirements for quality of service (QoS). However, fast-growing new applications and protocols bring us difficulties and challenges to classify various traffics correctly. Machine learning-based techniques are expected to be a more time-saving and precise method for traffic classification depending on the quality of services of various applications. In this paper, we focus on the traffic QoS classification based on the deep learning technique with traditional traffic features along with a newly defined feature in this paper, that is, the time period of network traffic. Experimental results show that by considering the time period feature, the classification accuracy can be improved much better than before.

Keywords: traffic classification, quality of service (QoS) and machine learning

1. Introduction

With the popularization of 5G communication, cloud computing, Internet of Things(IoT) and other technologies, the Internet users, devices and new application requirements show a continuous growth trend [1]. The constant growing applications are inexorably pushing the limitations of current network management. Traffic classification has been regarded as a crucial part to help improving the service quality of traffic planning and management, such as reducing packet loss, improving transmission latency and so on [2]. However, traffic classification becomes a hard work which requires usage of highly complex identification technologies, to fit in with the ever-changing nature of internet traffic applications. In particular,the COVID-19 pandemic has adversely restricted the movement of people and pushed people to communicate through the Internet. Various new applications (e.g., P2P, VoIP, real-time streaming, web browsing) have significantly magnified the difficulties of traffic classification.

Due to the rapid increase in the types and variety of applications, application-based traffic classification is facing enormous challenges. Real-time traffic monitoring and analysis, routing, resource allocation and other services based on application types will not be guaranteed with high quality. However, the proposal of QoS, as a method to quantify the requirements of applications with different metrics (such as latency and bandwidth), solves this problem. Therefore, to provide better services for the continuously increasing number and types of applications (especially new applications), QoS classification for applications is of vital importance.

Traffic classification based on QoS metrics has been put forward to overcome the above mentioned problems. Different from application or traffic size classification [3, 4], QoS classification simplifies the complexity of classification by comprising different traffic applications into classes based on each QoS requirements, which provides convenience for traffic management. Moreover, with the extensive application of artificial intelligence (AI), network traffic classification using machine learning techniques provides more accurate results and higher performance [5].

In this paper, we focus on the network traffic QoS classification based on deep neural network (DNN), which is one of the deep learning methods. From the traffic data log, we find the differences in traffic flow distribution among different time periods (e.g., morning, afternoon, etc.). With the awareness of time period of network traffic, we analyze the contribution ratio of each feature including time period to the DNN model through the shapley value and other methods. In addition, we compare the training and prediction accuracy of DNN models under different feature sets.

The remainder of this paper is as follows. In Section 2, we introduce the background of traffic classification methods, machine learning techniques and QoS. In Section 3, we introduce the previous works and summarize the contributions of this research. In Section 4, we introduce our dataset used in traffic QoS classification. In Section 5, we describe our simulation experiments and carry out evaluation results. In Section 6, we draw the conclusion and future works of this research.

2. Background

2.1 Traffic classification methods

To provide more efficient and comprehensive network traffic analysis, current researches mainly focus the bit level, packet level, flow level and so on [6]. Over the past years, in addition

[1] University of Tsukuba, Tsukuba, Ibaraki 305-8577, Japan
[a) s2030110@s.tsukuba.ac.jp
[b) s2020491@s.tsukuba.ac.jp
[c) ybzhang@sk.tsukuba.ac.jp

Table 1: Port number and protocol of main applications by IANA

Port number	Protocol	Application
20	TCP	FTP data
21	TCP	FTP control
22	TCP	SSH
23	TCP	Telnet
25	TCP	SMTP
53	UDP/TCP	DNS
67,68	UDP	DHCP
69	UDP	FTP TFTP
80	TCP	HTTP
110	TCP	POP3
161	UDP	SNMP
443	TCP	SSL

to the recent machine learning-based traffic classification method, traditional methods can be approximately divided into port-based classification and payload-based classification. In this section, we briefly introduce those traditional methods along with their advantages and disadvantages.

Port-based classification identifies the application based on inspecting the packet header and matching it with specific port number registered on the IANA [7]. The classification process of this method is uncomplicated and time-saving, and it behaves well with those well-known applications which have specific port number shown in Table 1. However, with the emergence of some applications using unregistered port numbers or picking random ports such as P2P applications, the port-based classification is no longer reliable. A few applications (usually non-legitimate applications) hide themselves transferred by well-known ports to avoid being filtered and bypassing restrictions of firewall or access control. Furthermore, modern security features like IP layer encryption (e.g., IPsec), and some encrypt-ions introduced by the application will make it impossible to know the actual port numbers. Therefore, the port-based method is a completely inapplicable choice for traffic classification.

Payload-based classification, also known as Deep Packet Inspection (DPI), is proposed as a more reliable method based on the inspection of packets compared with Port-based method [8]. This method has been widely used in several commercial and open-source tools, such as Linux Kernel Firewall [9] and network intrusion detection systems (IDS) [10]. The reliability of payload-based method has been inspected widely when the payload is not encrypted. However, this method requires excessive computing consumption and raises the consideration in privacy. Consequently, payload-based classification is not suitable for dealing with large volume network traffic.

With the rapid increasing of network traffic, traditional network traffic classification methods can not satisfy current requirements. To make classification performance more precise based on statistical characteristics, machine learning techniques are applied to traffic classifiers.

2.2 Machine learning techniques

Various machine learning techniques have been developed to deal with analyzing huge scale data. Machine learning techniques can be basically divided into four categories: unsupervised, supervised, semi-supervised and reinforcement learning [11]. The ability to analyze huge amount of data in a relatively short time and better classification accuracy make machine learning techniques extensively used.

Unsupervised learning is a kind of learning algorithm which do not require any prior knowledge or labelled data [12]. Unsupervised learning algorithms, which are commonly used in clustering, dimension reduction and association rule, mainly including K-means, Generative Adversarial Network (GAN), Self-Organizing Map (SOM), Local Outlier Factor (LOF) and Isolation Forest (IF). Advantages of unsupervised learning are less workload requirements including preparing and preprocessing the training dataset. However, with a huge amount of unlabeled training data requiring, unsupervised learning often converges more slowly to acceptable performance and requires extensive computational power and storage.

In contrast to unsupervised learning, supervised learning is a learning process to map the inputs to particular outputs achieved by using labelled data. While it requires much more labor to preprocess the dataset, its better performance and invulnerability to artifacts and anomalies make it applicable to be used in network traffic classification.

Commonly used supervised learning algorithms include Support Vector Machines (SVM), Deep Neural Network (DNN), Decision Tree (DT), Long Short-Term Memory Recurrent Neural Network (LSTM RNN), K-Nearest Neighbors (KNN), Naive Bayes (NB) and Random Forest (RF) [13]. In this paper, we use DNN for traffic classification which is suitable for various types and distributions of datasets [14]. The architecture of DNN is flexible to be adapted to classification problems of different scales. Furthermore, DNN can yield good performance within an acceptable time period.

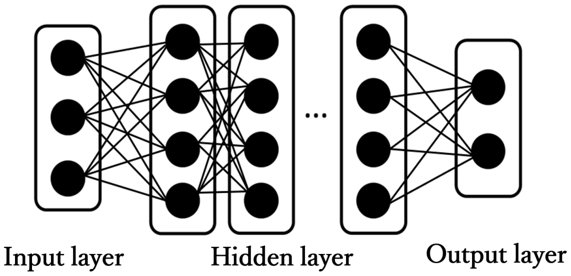


Fig. 1: Architecture of a DNN

As shown in Figure 1, DNN consists of three types of layers: input layer, hidden layer(s) and output layer. The input layer receives the input signal and transfer it to next layer through neurons without any processing. Then the signal is processed in the hidden layers, which perform nonlinear transformations and computations to abstract the input signal features at multiple levels. After signal processing in the hidden layers, the output layer outputs the results within a reasonable range by activation functions on neurons. The sigmoid function is usually regarded as the activation function of the output layer to make the value of output results between 0 and 1.

Table 2: Characteristics of the traffic flow in the UPC dataset [21]

Name	Date(YMD)	Day	Start Time	Duration	Packets	Bytes	Avg.Util
UPC-I	08-12-11	Thu	10:00	15min	95M	53G	482Mbps
UPC-II	08-12-11	Thu	12:00	15min	114M	63G	573Mbps
UPC-III	08-12-12	Fri	16:00	15min	102M	55G	500Mbps
UPC-IV	08-12-12	Fri	18:30	15min	90M	48G	436Mbps
UPC-V	08-12-21	Sun	16:00	1h	167M	123G	279Mbps
UPC-VI	08-12-22	Mon	12:30	1h	345M	256G	582Mbps
UPC-VII	09-03-10	Tue	03:00	1h	114M	78G	177Mbps

2.3 Quality of Service (QoS)

QoS refers to the measurement of the overall performance of different services for applications. To quantitatively measure QoS, several metrics of network services are generally considered, such as bandwidth, latency, jitter, packet loss and so on [15, 16]. The most critical QoS parameters used for data transmission are the bandwidth and latency requirements since they can intuitively describe the requirements of different network applications. In this paper, we focus on the QoS traffic classification considering bandwidth and latency requirements of each application. According to different levels of sensitivity to bandwidth and latency, we divide all the applications into four categories.

3. Related works and key contributions

In recent years, many researches are devoted to network traffic classification. In [21], Carela-Espanol et al. proposed a machine learning-based traffic classification model using the C4.5 supervised ML method, analyzing traffic flows collected in different time periods, from a NetFlow enabled server placed in the network edge between Internet and intranet in a university. The average classification accuracy based on machine learning is 90.59% which is approximately 10 times than traditional port-based method. This shows the tremendous improvement in classification accuracy brought by machine learning techniques. However, the time period, as a key factor in their dataset, was not considered as a feature for traffic analyzing. In [18] and [19], Balanici and Pachnicke proposed a new ML-based traffic prediction model using Long short-term memory neural networks (LSTM NNs), analyzing and predicting traffic floes of real-time server in a Data Center (DC). This research demonstrates an incredible 99% accuracy for highly variable and bursty traffic flows, which is much higher than other related works. Researchers implemented not only the LSTM model to vanilla recurrent neural networks (RNNs), but also a prediction sliding window and performing forecasting to solve the challenging issue of multi-step forecasting. Although the LSTM model is efficient and capable for real-time traffic prediction, it is relatively limited that the characteristics and QoS requirements can not be obtained from the model. In [20], Xiaotao Guo et al. proposed a QoS aware resource allocation method based on deep reinforcement learning. This research focused on various QoS requirements supported by DCs and trying to optimize the DC network topology by using the deep reinforcement learning and a technique called Actor-Critic Reconfiguration for OP Square (ACRO). However, the time period should be considered in the DC network topology optimization because of its significant impact on traffic distribution.

In this paper, besides the traditional features used for describing traffic flow of applications, we additionally consider the time each traffic flow occurs as a key feature in traffic classification based on DNN. The main contributions of this paper are as follows. Our research features on traffic classification considering the correlation between time period and applications and extends it to the QoS groups. We select dataset from 5 different time periods and introduce the time period as a feature for classification in our DNN model. Furthermore, we classify the traffic applications into 4 QoS categories in terms of latency and bandwidth requirements. Focusing on QoS classification is not so limited to well-known network applications as the previous application classification, and is more suitable for the explosive growth of network applications in the future.

4. Dataset used in traffic QoS classification

The dataset we used in this paper is provided freely at the Gigabit access link of the Universitat Politecnica de Catalunya (UPC) [21]. The dataset consists of 7 traffic log blocks from distinct time periods and contains 50,000 applications execution records (IS THE CORRECT?).

The dataset is not a raw one but processed by UPC. Table 2 illustrates the characteristics of the traffic flows in the dataset. The UPC dataset has 10 features and 11 groups of network applications, which are shown in Table 3 and 4, respectively.

Table 3: Description of features in UPC dataset

Feature	Description	Size (Unit: bits)
Sport	Source port of the flow	16
Dport	Destination port of the flow	16
protocol	IP protocol value	8
ToS	Type of Service from the first packet	8
flags	Cumulative OR of TCP flags	6
duration	Duration of the flow in nanosec precision	N/A
packets	Total number of packets in the flow	N/A
bytes	Flow length in bytes	N/A
pkt-size	Average packet size of the flow	N/A

Different network applications are divided into 11 groups. Applications in each group have the same characteristics with each other. Otherwise, the proportions of application groups vary from different time periods in a day and the average flow size of each application group is also different. This further shows the necessity of time period as a feature for classification.

In this paper, we consider further the time a flow occurs as a feature for classification. From the characteristics shown in Table 2, we can find that UPC-II and UPC-VI are collected at around 12:00 and are classified as Noon, UPC-III and UPC-V

Table 4: Application groups used in UPC datasets

Group	Applications
P2P	Peer-to-peer file sharing applications (BitTorrent, Edonkey)
HTTP	HTTP related applications (http, httpcachemiss, etc.)
VoIP	Voice communication applications (Skype, Teamspeak, etc.)
Network	Network applications (BGP, DHCP, SSH, Telnet, etc.)
Streaming	Media streaming applications (PPLive, Itunes, QuickTime, etc.)
DNS	Domain Name System traffic
Others	CVS, Hddtemp, IPP, unknown data, etc.
Chat	Real-time chat applications (Aim, IRC, MSN Messenger, etc.)
Email	Email related traffic (IMAP, POP3, SMTP)
FTP	File transfer applications (FTP, Tftp, etc.)
Games	Online games (Battlefield, Counter-strike, WoW, etc.)

are collected from 16:00 to 16:15, whose time period are defined as Afternoon. In addition, the time periods of UPC-I, UPC-IV and UPC-VII are defined as Morning, Evening and Late Night respectively according to their start time. Therefore, as shown in Table 5, the 7 datasets are divided into 5 categories, each of which presents a specific time period.

Table 5: Time periods defined in UPC datasets

Dataset	Start Time	Time period	Time class
UPC-I	10:00	Morning	1
UPC-II, UPC-VI	12:00, 12:30	Noon	2
UPC-III, UPC-V	16:00	Afternoon	3
UPC-IV	18:30	Evening	4
UPC-VII	03:30	Late night	5

Due to the rapid increasing of application categories, it is hard to classify new applications into those well-known 11 application groups shown in Table 4. In addition, different kinds of applications may have the same QoS requirements. Therefore, in this paper, we focus on the traffic QoS classification instead of application classification based on 2 key metrics: latency and bandwidth.

According to the QoS requirements of different applications defined in [22], we divide the applications into 2 categories: real time application and non-real time application based on the effect of latency on Human Voice Perception. As shown in Table 6, we define the label of non-real time applications as L1, which indicates weakly sensitive to latency. Otherwise, we define the sensitivity of real time applications to latency as L2, which means strongly sensitive to latency.

Table 6: Latency and bandwidth requirements of different applications

App.	Latency req.	Latency class	Bandwidth req.	Bandwidth class
P2P	Non-real time	L1	>10	B1
HTTP	Non-real time	L1	5 ~ 25	B1
VoIP	Real time	L2	<0.5	B2
Network	Non-real time	L1	<0.5	B2
Streaming	Real time	L2	5 ~ 8 for HD 25 for 4K	B1
DNS	Non-real time	L1	<0.5	B2
Others	Non-real time	L1	<0.5	B2
Chat	Real time	L2	1	B2
Email	Non-real time	L1	1	B2
FTP	Non-real time	L1	>10	B1
Games	Real time	L2	>4	B1

We can obtain from [23] that a 4Mbps connection is sufficient for bandwidth requirements of basic applications, but inadequate for highly demanding applications. Therefore, we set 4Mbps as the boundary of sensitivity to bandwidth requirements. Similarly, Table 6 indicates the 2 types of bandwidth requirements of applications, in which B1 represents high bandwidth requirement (>4Mbps), B2 represents low bandwidth requirement (<4Mbps). We assume the average bandwidth requirements illustrated in FCC [24] as the reference of bandwidth requirements of applications.

Therefore, we can divide all the applications into 4 QoS groups based on different latency and bandwidth classes defined in Table 6: Group 1: (L1,B1) represents applications which are weakly sensitive to latency and require high bandwidth; Group 2: (L1,B2) represents applications which are weakly sensitive to latency and require low bandwidth; Group 3: (L2,B1) represents applications which are strongly sensitive to latency and require high bandwidth; Group 4: (L2,B2) represents applications which are strongly sensitive to latency and require low bandwidth. The QoS requirements and classes of different applications are shown in Table 7.

Table 7: QoS requirements of different applications

Application	QoS requirement	QoS class
P2P	(L2,B1)	3
HTTP	(L2,B1)	3
VoIP	(L1,B2)	2
Network	(L2,B2)	4
Streaming	(L1,B1)	1
DNS	(L2,B2)	4
Others	(L2,B2)	4
Chat	(L1,B2)	2
Email	(L2,B2)	4
FTP	(L2,B1)	3
Games	(L1,B2)	2

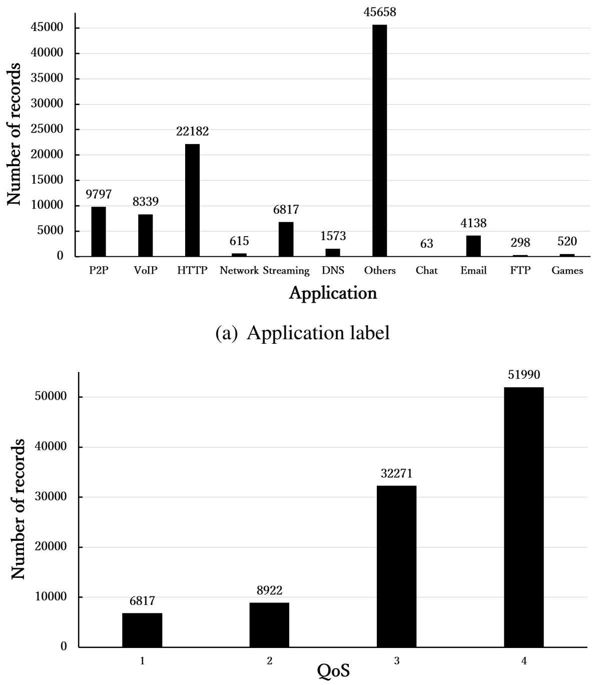
5. Simulation experiments and results

In this section, to evaluate classification accuracy of datasets with different feature sets and labels, we use the same scale training and testing sets to conduct simulation experiments under the DNN model with the same parameter settings.

Table 8: Parameters for DNN model

Item	Parameters
Input layer	Num. of neuron units: same as number of features Activation function: ReLu Drop out rate: 0.1
Hidden layer	Num. of layers: 10 Num. of neyron units: 200 Activation function: ReLu Drop out rate: 0.1
Output layer	Num. of neuron units: same as number of labels Activation function: Softmax
Loss function	Cross entropy
Batch size	1000
Epoch	100
Learning rate	0.001

Due to the large scale of the original dataset, we take the data



(a) Application label

(b) QoS label

Fig. 2: Distribution of datasets with application label and QoS label

from different time periods and application groups according the distribution of the original dataset to form training and testing sets. Specifically, we choose 100,000 records from the original dataset, including 90,000 records as training set and 10,000 records as test set. Therefore, the distribution of datasets labeled with applications and QoS is shown in Figure 2.

As mentioned before, we divide our final dataset with 100,000 records into training and testing set in ratio 9:1. With different training and testing sets, we use DNN model to evaluate the loss of training process and the classification accuracy. The neuron units in input and output layer differ in different experiments carried out due to the difference in number of features and labels. The corresponding parameters of DNN model are defined as shown in Table 8.

To illustrate the necessity of time period and QoS requirements being considered into traffic classification, we compare our proposed time-aware QoS classification with application classification and QoS classification without time period awareness. The performance results of these three classifications including training loss and classification accuracy are illustrated in Table 9.

Table 9: Training loss and classification accuracy via different classification methods

Classification	Training loss	Accuracy
Time-aware QoS-based	0.6405	0.7643
Time-aware Application-based	0.6733	0.7627
Application-based	0.9724	0.6674

As known from Table 9, our proposed time-aware QoS classification outperforms other previous methods in both training loss and classification accuracy. Specifically, taking into account time period as a feature for network traffic classification will have an significant improvement in performance. Moreover, treating QoS requirements as classification labels rather than applications can improve the convergence efficiency of DNN model.

However, using QoS labels instead of application labels does not perform well as expected. This may be due to the lack of detailed description of applications in the original data. For example, a network traffic with the application label of HTTP can be further divided into HTTP control traffic or HTTP data traffic, which has different QoS requirements. Another example as P2P traffic, which mainly contains P2P peer discovery traffic , P2P peer communication traffic and P2P data traffic. The bandwidth requirements of P2P peer discovery and communication traffic are far less than P2P data traffic, resulting in different QoS requirements. Therefore, the imprecise description of traffic records in the original dataset causes QoS classification performance yields not as expected.

To further investigate the impact of time period on classification performance, we evaluate training loss and classification accuracy of DNN model with different feature sets, which correspond to the feature sets after removing one feature respectively. The classification performance comparisons of complete 10 feature-based with other 9 feature-based are shown in Table 10.

Table 10: Training loss and classification accuracy via different feature sets

Feature set	Training loss	Accuracy
10 features	0.6405	0.7643
9 features without time period	0.7850	0.6712
9 features without Sport	0.6853	0.7485
9 features without Dport	0.6694	0.7519
9 features without protocol	0.6459	0.7622
9 features without ToS	0.6303	0.7675
9 features without flags	0.6667	0.7663
9 features without duration	0.7214	0.7429
9 features without packets	0.6637	0.7581
9 features without bytes	0.6534	0.7570
9 features without pkt-size	0.7395	0.7068

No matter which feature is removed except ToS and flags, training accuracy and loss performance are worse than complete 10 feature-based classification. Figure 3 indicates the accuracy and loss gap between different feature-based classifications shown in Table 10 and complete 10 feature-based classification. The accuracy and loss gap are computed as formula (1) and (2). x is the feature set with complete 10 features and $x\setminus i$ is the feature set after feature i removed. Apparently, if the time period feature is removed, the classification accuracy will be reduced to the greatest extent. Similarly, the training loss will has the greatest extent of increase. From the above results, we can obtain that time period is the most critical of the 10 features, which further proves the necessity of time period as a feature for network traffic classification.

$$Gap_{Accu_{x\setminus i}} = (Accu_{x\setminus i} - Accu_x) \times 100\% \qquad (1)$$

$$Gap_{Loss_{x\setminus i}} = (Loss_{x\setminus i} - Loss_x) \times 100\% \qquad (2)$$

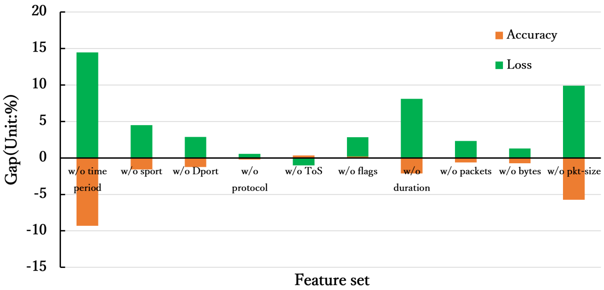


Fig. 3: Performance gap with 10 feature-based classification via different feature sets

Furthermore, to accurately quantify the importance of time period feature to the classification by DNN model, we introduce another performance metric called Shapley value [25,26] to measure the importance of each feature to DNN model. Shapley value is a game-theoretic formulation of feature importance which is defined through a cooperative game between features and distribute influence among input elements [27]. Concretely, during the training process of DNN model, we compute Shapley value of each feature with following formula:

$$\Phi_i(x) = \sum_{S \in N \setminus i} \frac{|S|!(|N| - |S| - 1)!}{|N|!}(f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)) \quad (3)$$

In formula (3), x is the feature set, i is the ith feature and $\Phi_i(x)$ refers to the importance of ith feature in feature set x. N is the set of permutations and combinations of all features and S is the set excluding feature i in set N. $|S|!(|N| - |S| - 1)!$ defines the number of permutations and combinations of features before and after feature i and $|N|!$ defines number of permutations and combinations of all features. In addition, $f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)$ refers to the gain obtained by adding feature i.

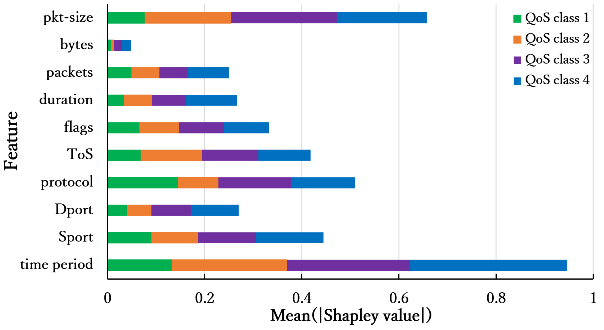


Fig. 4: Shapley value: average impact on model output magnitude via different features

Figure 4 shows the shapley values of different features corresponding to 4 QoS categories. Inevitably, time period has the maximum Shapley value among 10 features no matter in which QoS class. In other words, time period is the feature that has the greatest impact on classification output of DNN model, which also proves the importance of time period as a feature of network traffic classification.

6. Conclusions and future works

In this paper, we focus on network traffic QoS classification based on one of the supervised learning methods called DNN with awareness of time period. We introduced the Shapley value to examine the importance of the features used in the DNN model. The result shows that the time period is the most critical feature. We compare our proposed time-aware QoS classification with previous works in classification accuracy and training loss based on a DNN model developed in this paper. The results show that the time feature plays a most important role in traffic classification and the accuracy and loss when adding the time feature is much better than any other feature. Overall, our findings indicate that considering QoS requirements for traffic classification can improve convergence efficiency of model and time period is a feature worthy of consideration for network traffic classification.

Future works will consider the correlation between features combined with Shpaley value of features for further study on feature dimension reduction. Moreover, more QoS requirement parameters such as jitter and packet loss should be considered in traffic QoS classification.

Acknowledgments This work has been partially supported by JST SPRING, Grant Number JPMJSP2124 and by JSPS Grant-in-Aid for Scientific Research (C) 21K04544.

References

[1] Cisco U.: Cisco annual internet report (2018–2023) white paper. 2020[J]. Acessado em, 2021, 10(01).

[2] Tahaei H, Afifi F, Asemi A, et al. : The rise of traffic classification in IoT networks: A survey. *Journal of Network and Computer Applications*, 2020, 154: 102538.

[3] Chhabra A, Kiran M.: Classifying elephant and mice flows in high-speed scientific networks. *Proc. INDIS.*, 2017: 1-8..

[4] Erman J, Arlitt M, Mahanti A.: Traffic classification using clustering algorithms. *Proceedings of the 2006 SIGCOMM workshop on Mining network data.*, 2006: 281-286.

[5] AlZoman R M, Alenazi M J F.: A comparative study of traffic classification techniques for smart city networks. *Sensors*, 2021, 21(14): 4677.

[6] Liu Y, Li W, Li Y. : Network traffic classification using k-means clustering. *PSecond international multi-symposiums on computer and computational sciences (IMSCCS 2007). IEEE*, 2007: 360-365.

[7] IANA: Internet Assigned Number Authority. *https://www.iana.org/assignments/service-names-port-numbers/service-names- port-numbers.xhtml*, 2021.

[8] Moore A W, Papagiannaki K.: Toward the accurate identification of network applications. *International Workshop on Passive and Active Network Measurement.* Springer, Berlin, Heidelberg, 2005: 41-54.

[9] Levandoski J.: Application layer packet classifier for Linux. *http://l7-filter. sourceforge. net/*, 2008.

[10] Paxson V.: Bro: A system for detecting network intruders in real-time. *Computer networks* 1999, 31(23-24): 2435-2463.

[11] Liu W, Wang Z, Liu X, et al.: A survey of deep neural network architectures and their applications. *Neurocomputing* 2017, 234: 11-26.

[12] Wang D L. : Unsupervised learning: foundations of neural computation. *Ai Magazine*, 2001, 22(2): 101-101.

[13] Muhammad I, Yan Z. : SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY. *ICTACT Journal on Soft Computing*, 2015, 5(3).

[14] Zheng W, Yang M, Zhang C, et al.: Application-aware QoS routing in SDNs using machine learning techniques. *Peer-to-Peer Networking and Applications*, 2021: 1-20

[15] Wang Z, Crowcroft J. : Quality-of-service routing for supporting multimedia applications. *EEE Journal on selected areas in communications*, 1996, 14(7): 1228-1234.

[16] Conti M, Gregori E, Panzieri F.: Load distribution among replicated Web servers: A QoS-based approach. *ACM SIGMETRICS Performance Evaluation Review*, 2000, 27(4): 12-19.

[17] Carela-Español V, Barlet-Ros P, Cabellos-Aparicio A, et al.: Analysis of the impact of sampling on NetFlow traffic classification. *Computer Networks*, 2011, 55(5): 1083-1099.

[18] Balanici M, Pachnicke S.: Machine learning-based traffic prediction for optical switching resource allocation in hybrid intra-data center

networks. *Optical Fiber Communication Conference*, Optical Society of America, 2019: Th1H. 4.

[19] Balanici M, Pachnicke S.: Classification and forecasting of real-time server traffic flows employing long short-term memory for hybrid E/O data center network. *Journal of Optical Communications and Networking*, (distributed to authors).2021, 13(5): 85-93.

[20] Guo X, Yan F, Xue X, et al.: A QoS-aware network reconfiguration method in data centers based on deep reinforcement learning. *45th European Conference on Optical Communication (ECOC 2019)*, IET, 2019: 1-4.

[21] Carela-Espanol V, Barlet-Ros P, Cabellos-Aparicio A, et al.: Analysis of the impact of sampling on NetFlow traffic classification. *Computer Networks*, 2011, 55(5): 1083-1099.

[22] Chen Y, Farley T, Ye N.: QoS requirements of network applications on the Internet, *Information Knowledge Systems Management*, 2004, 4(1): 55-76.

[23] Saunders J D, McClure C R, Mandel L H.: Broadband applications: Categories, requirements, and future frameworks. *First Monday*, 2012.

[24] Federal Communications Commission (FCC).: Broadband Speed Guide. *https://www.fcc.gov/sites/default/files/broadband_speed_guide.pdf*, 2021.

[25] Sundararajan M, Najmi A.: The many Shapley values for model explanation. *International Conference on Machine Learning*, PMLR, 2020: 9269-9278.

[26] Wang R, Wang X, Inouye D I.: Shapley Explanation Networks. *arXiv preprint arXiv:2104.02297*, 2021.

[27] Kumar I E, Venkatasubramanian S, Scheidegger C, et al.: Problems with Shapley-value-based explanations as feature importance measures. *International Conference on Machine Learning*, PMLR, 2020: 5491-5500.