Network Traffic Classification Model Based on Multi-Task Learning

Shuaibao Nie, Lifen Jiang*, Huazhi Sun, Chunmei Ma, Yan Liang, Yongheng Zhou, Ying Zuo
College of Computer and Information Engineering, Tianjin Normal University, Tianjin, 300387, China
nsb199562@126.com
*jianglifen@tjnu.edu.cn

Abstract. A multi-task learning network traffic classification model (MTL-NTC) is experimentally studied. The identification task of the MTL-NTC model contains three tasks: identifying the type of traffic, predicting the bandwidth, and the duration of the flow. Identifying the type of traffic is the main task. It includes the classification of five types of traffic. The bandwidth prediction and the duration of the flow are auxiliary tasks, which are used to improve the performance of the main task. In the public network traffic data set QUIC, the MTL-NTC model and the common model are used for comparative experiments. The experimental results show that the recognition accuracy of this model is 97.33%, which is better than the common models.

1. Introduction
Network traffic classification is an important task in modern communication networks [1]. Accurate traffic classification is the key to network resource management, such as monitoring, firewall, filtering, anomaly and intrusion detection [2]-[3]. Traffic classification has aroused widespread interest in industrial activities related to network management [4]-[5].

In recent years, researchers have applied deep learning technology to the field of network traffic classification to make full use of the inherent advantages of deep learning models, such as automatic feature learning, so that they have higher efficiency and accuracy than classic machine learning models. Literature [6] uses convolutional neural networks and stacked autoencoders (SAE) together for the ISCX dataset to classify traffic types and applications. Literature [7] uses Regenerated Kernel Hilbert Space (RKHS) to convert the time series features of each stream into a two-dimensional image, and then uses the resulting image as the input of the CNN model. Literature [8] proposed a semi-supervised learning method based on Deep Convolution Generative Adversarial Network (DCGAN). The basic idea is to use the samples generated by the DCGAN generator and the labeled data to improve the performance of the classifier trained on a small number of labeled samples, and to alleviate the difficulties associated with the collection and labeling of large data sets. Literature [9] uses convolutional neural networks, long short-term memory (LSTM) models, and various combinations to classify multiple services (such as YouTube and Office365).

Multi-task learning can not only improve the generalization of the model, but also use other related tasks to improve the performance of the main task. The purpose of the auxiliary task is to help find a stronger and more robust feature representation, which ultimately benefits the main task. Multi-task
learning has been widely used in other fields such as computer vision and natural language processing.
For computer vision, Literature [10] uses multi-task learning to segment images, and uses the uncertainty of the same variance as noise for training, thereby balancing different types of tasks. For natural language processing, literature [11] combines many related tasks, such as semantic annotation and name entity recognition, to learn better sentence representation. But for the problem of traffic classification, people rarely pay attention to multi-task learning methods. The MTL-NTC model introduces dilated convolution to extract the characteristics of traffic, which can effectively increase the receptive field, better learn network traffic characteristics, capture multi-scale context information without introducing additional parameters; multiple tasks share a lot Model parameters to reduce the risk of overfitting and improve the accuracy of the classifier.

2. Traffic Classification Model

2.1. Model building
In order to better classify network traffic, this paper proposes a network traffic classification model based on multi-task learning. The model includes three subtasks: identifying the type of traffic (Task1), predicting the bandwidth (Task2), and the duration of the flow (Task3). Among them, identifying the traffic type is the main task, which includes the classification of five traffic types. Predicting the bandwidth and the duration of the flow are auxiliary tasks. The overall structure of the model is shown in Fig.1.

Fig.1 Overall structure of traffic classification model based on multi-task learning

Fig.2 One-dimensional dilated convolution structure

Fig.3 Structure of the max-pooling layer

The traffic classification model based on multi-task learning includes an input layer, 4 dilated convolutional layers, 2 maximum pooling layers, 2 fully connected layers, and multi-task output layers.

Firstly, three time series features are extracted from the traffic data, namely the length of the previous $k$ data packet, the arrival interval time and the direction. It is input into a one-dimensional convolutional layer with a convolution kernel size of (3,32) for feature extraction. Dilated convolution can effectively increase the receptive field, capture multi-scale context information without introducing additional parameters. The structure of one-dimensional dilated convolution is shown in Fig.2.
The pooling layer can be used to reduce the dimensionality of features and retain valid information. The pooling layer of the MTL-NTC model adopts the maximum pooling method, and its hierarchical structure is shown in Fig.3. In model training, a maximum pooling layer is set every two one-dimensional convolutional layers, and the pooling core used is 2. Then two fully connected layers are used continuously to synthesize the previously extracted features. They improve the nonlinear expression ability of the model, thereby improving the learning ability of the model. The Relu function is used as its activation function in the fully connected layer.

This paper uses a traffic classification model based on multi-task learning. The model includes three subtasks: identifying the type of traffic (Task1), predicting bandwidth (Task2), and the duration of the flow (Task3). Traffic classification is the main task. Prediction of bandwidth and flow duration are auxiliary tasks. The main task is the classification task, and we divide the traffic in the QUIC data set into five categories: Google Doc, Google Drive, Google Music, Youtube and Google Search. At the same time, the two auxiliary tasks are also defined as classification tasks, the bandwidth and the duration of the flow are divided into five and four categories, and these categories are defined in an intuitive way, each of which covers a part of the data set. Their category definitions are shown in Tab.1.

| Tab.1 Auxiliary task classification |
|------------------------------------|
| Classification | Bandwidth (B) | Duration (D) |
| 1 | B<10kbps | D<10s |
| 2 | 10kbps<B<50kbps | 10s<D<30s |
| 3 | 50kbps<B<100kbps | 30s<D<60s |
| 4 | 100kbps<B<1mbps | 30s<D<60s |
| 5 | B>1mbps |

In this article, bandwidth and duration are easy to obtain in large batches and do not require manual marking, so they are used as auxiliary tasks for traffic classification tasks. Since the QUIC data set has five types of traffic, the traffic classification task is divided into five categories. Similarly, the bandwidth and the duration of the flow are divided into five and four categories. Because there are three tasks, the output of the three fully connected layers is classified by the softmax function.

2.2. Loss function

For a multi-task learning model, when optimizing network parameters, different loss functions need to be specified for different tasks. In this paper, there are three prediction tasks, namely bandwidth, duration, and traffic category classification. We use B, D and T to represent them respectively. In the experiment, the three tasks all use the cross-entropy loss function. In the MTL-NTC model, the traffic classification task is the main task, and we assign a weight to its loss function to adjust its importance in the total loss function. We need to find an appropriate weight to get the best classification effect. The total loss function in the MTL-NTC model is shown in formula (1).

$$\arg\min _{\hat{W}_s,\hat{W}_b,\hat{W}_t} \sum _{i=1}^{N} \ell \left(y_i^b, f(x_i; \hat{W}_s^b)\right) + \ell \left(y_i^d, f(x_i; \hat{W}_s^d)\right) + \lambda \ell \left(y_i^t, f(x_i; \hat{W}_s^t)\right)$$

(1)

Among them, N represents the number of training data samples, and \(x_i\) represents the input of the i-th data sample. \(y_i^b\), \(y_i^d\) and \(y_i^t\) represent the bandwidth, duration, and expected output of the traffic classification task. \(\hat{W}_s^b\), \(\hat{W}_s^d\) and \(\hat{W}_s^t\) Respectively represent the weight of bandwidth task training, the weight of duration task training and the weight of traffic classification task training. \(f(x_i; \hat{W}_s^b)\), \(f(x_i; \hat{W}_s^d)\), and \(f(x_i; \hat{W}_s^t)\) Represents the actual output of bandwidth, duration and traffic classification task respectively. \(\ell\) Represents the cross entropy loss function. \(\lambda\) is the weight representing the importance of the traffic category prediction task.
3. Experiment and analysis

3.1. Data set establishment and parameter setting
This article uses the QUIC data set of the University of California, Davis. The data set contains five types of traffic, Google Doc, Google Drive, Google Music, Youtube and Google Search. The number of streams included in each traffic is shown in Fig.4. This paper extracts the time series characteristics of each traffic in the data set (the length of the data packet, the time between arrivals and the direction) to obtain a new data set. The training set samples in this data set account for 80% of the total samples, and the test data set samples account for 20% of the total samples.

This article trains the model on the QUIC dataset and uses Python and Keras to implement a multi-task learning method. In the deep learning model, the selection of different parameters has an important influence on the experimental results. This article manually adjusts the parameters through experiments and combined with practical experience. When training the model, the Adam optimizer is used for training, and the parameters batch_size=16, epochs=30 are set. Next, we set the number of data packets \( k \) and the weight \( \lambda \) of the traffic classification task through experiments to optimize the performance of the model.

3.2. Number of data packets \( k \)
We need to evaluate the impact of the number of packets \( k \) on the task of traffic classification. In theory, increasing the number of data packets can enhance the model's ability to extract data features, thereby improving the performance of the model. In the process of using the model in this paper to classify traffic, the number of data packets needs to be verified experimentally. The verification results are shown in Fig.5. As the number of data packets increases, the accuracy of the model shows a trend of first rising and then falling. Through experimentation and combining experience, when the number of used data packets is 60, the performance of the model is optimal.
3.3. Weight of traffic classification task

Appropriate weight $\lambda$ can help the model improve the accuracy of traffic classification tasks, so it is necessary to find a suitable value as a hyperparameter. The comparative experimental results of different weights are shown in Fig.6.

![Graph showing comparison of experimental results of different weight models](image)

Fig.6 Comparison of experimental results of different weight models

It can be seen from Table 5 that the accuracy of the model changes with the increase of the traffic classification task weight $\lambda$. According to the experimental results, this paper sets the size of the traffic classification task weight $\lambda$ to 1.

3.4. Experimental results and analysis

In order to verify the effectiveness and performance of the model in this paper, a multilayer perceptron (MLP) and a convolutional neural network (CNN) are built, using the following structures and hyperparameters, as shown in Tab.2 and Tab.3.

| Tab.2 Architecture and hyperparameters of MLP |
|-----------------------------------------------|
| Architecture | Hyperparameters                                      |
| Layer 1      | Dense, output:512, relu activation, dropout(rate=0.2) |
| Layer 2      | Dense, output:512, relu activation, dropout(rate=0.2) |
| Layer 3      | Dense, output:5, softmax activation                  |
| Layer 3      | Dense, output:5, softmax activation                  |

| Tab.3 Architecture and hyperparameters of CNN |
|-----------------------------------------------|
| Architecture | Hyperparameters                                      |
| Layer 1      | 2D convolutional layer 32 (5*5) filters, relu activation |
| Layer 2      | 2D convolutional layer 32 (5*5) filters, relu activation |
| Layer 3      | Dense, output:5, softmax activation                  |

In order to evaluate the effect of the model, we compare the MTL-NTC model proposed in this paper with the baseline model (MLP, CNN, Single-task learning [12]) on the QUIC data set. The statistical results of the traffic classification task are shown in Tab.4.

| Tab.4 Identification accuracies of 4 types of traffic classification |
|---------------------------------------------------------------|
| Models            | Accuracy/\%                  |
| MLP               | 77.55                         |
| CNN               | 83.53                         |
| Single-task learning | 96.67                   |
| MTL-NTC           | 97.33                         |

It can be seen from Fig.7 that the recognition accuracy of the traffic classification model MTL-NTC proposed in this paper based on multi-task learning reaches 97.33%, which is higher than 77.55% of the MLP model and 83.53% of the CNN model, and slightly higher than 96.67% of the
Single-task learning model. The above experimental results show that the MTL-NTC model proposed in this paper has certain advantages in solving the task of network traffic classification.

4. Conclusion
Based on the results and discussions presented above, the conclusions are obtained as below:
(1) The use of dilated convolution in the MTL-NTC model to extract the characteristics of the flow can effectively increase the receptive field and better learn the characteristics of the network flow.
(2) Applying the knowledge of multi-task learning to the field of network traffic classification, a large number of model parameters can be shared between multiple tasks, which reduces the risk of overfitting and improves the accuracy of the classifier.
(3) It is concluded that the MTL-NTC model can effectively classify network traffic, and the accuracy obtained is higher than that of commonly used models.

Acknowledgments
This work was financially supported by the National Natural Science Foundation of China (61702370); Tianjin Natural Science Foundation (18JCYBJC85900, 18JCQNJC70200); Tianjin Municipal Education Commission Scientific Research Project (JW1702); Tianjin Normal University 131 three-level candidates (043/135305QS20).

References
[1] Bagui S,Fang X,Kalaimannan E,et al. Comparison of machine-learning algorithms for classification of vpn network traffic flow using time-related features[J].Journal of Cyber Security Teclnology,2017,1(2):108-126.
[2] T. T. Nguyen, G. Armitage.A survey of techniques for Internet traffic classification using machine learning[J].IEEE Communications Surveys & Tutorials, 2008,10(4): 56-76.
[3] F. Zhang,W. C, Liu,et al.Inferring Users Online Activities Through Traffic Analysis[C]//ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2011.
[4] Dainotti A,Pescape A,Claffy K C,et al.Issues and future directions in traffic classification[J].IEEE network,2012,26(1):35-40.
[5] Finsterbusch M,Richter C,Rocha E,et al.A survey of payload-based traffic classification approaches[J]. IEEE Communications Surveys & Tutorials,2014,16(2):1135–1156.
[6] Lotfollahi,Mohammad,Mahdi Jafari Siavosha-ni,et al.Deep packet: A novel approach for encrypted traffic classification using deep learning[J].Soft Computing,2020,24: 1999-2012.
[7] Chen,Zhitang,Ke He,et al.Seq2img: A sequence-to-image based approach towards ip traffic classification using convolutional neural networks [C]//IEEE International Conference on Big Data (Big Data),2017.
[8] Auwal Sani Iliyasu,Huifang Deng.Semi-Supervised Encrypted Traffic Classification With Deep Convolutional Generative Adversarial Networks [J].IEEE Access,2020,5:118-126.
[9] Lopez-MartnManuel,Belen Carro,et al.Netwo- rk traffic classifier with convolutional and recurrent neural networks for Internet of Things[J]. IEEE Access,2017,5:18042-18050.
[10] R. Cipolla,Y. al,A. Kendall.Multi-task learning using uncertainty to weigh losses for scene geometry and semantics[J].arXiv:1705.07115,2017.
[11] V. Perera,T. Chung,T. Kollar,et al.Multi-task learning for parsing the alexa meaning representation language[C]//AAAI,2018.
[12] S. Rezaei,X. Liu.Multitask learning for network traffic classification[J].arXiv:1906.05248,2019.