Effective Media Traffic Classification Using Deep Learning

Qing Lyu
Tsinghua University
Beijing
lvq16@mails.tsinghua.edu.cn

Xingjian Lu
Beijing University of Posts and Telecommunications
Beijing
lxjarooba@bupt.edu.cn

ABSTRACT

Traffic classification (TC) is very important as it can provide useful information which can be used in the flexible management of the network. However, TC has become more and more complicated because of the emergence of various network applications and techniques. In this paper, we apply deep learning based method to the classification of four different kinds of media traffic, i.e., audio, picture, text and video. We collect traffic data from the real network environment. Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN) based traffic classification method are designed to accurately classify the target traffic into different categories. We found that MLP has very good performance in most scenarios. Moreover, specific architecture can reduce the training time of the neural network in the classification.

CCS CONCEPTS

• **Networks → Network algorithms**; *Network services*; • **Theory of computation →** *Design and analysis of algorithms*;

KEYWORDS

Traffic Classification; Deep Learning

ACM Reference Format:
Qing Lyu and Xingjian Lu. 2019. Effective Media Traffic Classification Using Deep Learning. In *Proceedings of International Conference on Compute and Data Analysis (ICCDA' 2019).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The internet has become more and more complex due to the various appearing new applications and the development of techniques such as Software-Defined Networking (SDN) [10, 27] and Network Function Virtualization (NFV) [11]. Under the situation that numerous and complicated traffic are often protected by the modern internet techniques, a crucial task is to correctly classify them into different categories, which is known as traffic classification (TC). Accurate classification of internet traffic can bring the Internet Service Provider (ISP) with more flexible arrangements of internet service such as pricing and bandwidth allocation. Operators can also learn from the classification results and promptly adjust the management of internet devices or the forwarding rules to make the internet more efficient [3, 23, 44]. The key problem is to provide precisely traffic classification.

TC needs to take care of ever-changing traffic in the internet and make sure that the classification results can meet the requirements of network users. For example, flow priorities of the incoming flows and the bandwidth allocation [19] should be suitably arranged according to the classification results. How precisely and can the problem of TC be solved is responsible for the efficiency of bandwidth management [16, 19, 22, 23, 40], On the other hand, different kinds of traffic from different customers should be properly managed to meet the customers' Quality of Experience (QoE) Therefore, it is of significant importance to improve the performance of TC.

A lot of work has been proceeded by researchers under the topic of TC. Some of them focus on the classification of internet applications such as whether it is HTTP traffic or FTP traffic [4, 25, 28, 29, 36]. The research emphasis is put on the identification of different protocols. Some priori knowledge is often required there. Many other researchers are interested in the anomaly detection [31] and malware detection [33]. These works often play a vital role in the Intrusion Detection System (IDS) which is designed to protect the internet from various malicious attacks. To protect the internet users from network security problems automatically, less human intervention is desired. Some others devote themselves to the study of the classification of encrypted traffic [24].

In this paper, we consider the traffic classification problem from another perspective where we do not classify the traffic with respect to the protocols, but try to study the media types of the traffic. Generally, we are interested in finding out whether the current traffic is a video traffic or a audio traffic or whether it is a picture traffic or a text traffic. Because different media type of traffic require different amount of network bandwidth and transmission latency. For example, video streaming traffic is often huge and need a great deal of bandwidth, while audio traffic cannot tolerate large latency because some audio traffic can be VoIP traffic which needs to be transmitted in real time. The classification of traffic types rather than specific protocols can benefit the network management in a more flexible level and ISP can provide different kinds of service according to the traffic's bandwidth demands and latency demands. Therefore, the accurate classification of the media types of traffic is extremely important. We collect the traffic dataset which contains the four classes of labeled traffic from the real network environment. The dataset include more than 10000 records of flows that are almost evenly distributed among the four classes of traffic. In consideration of the deficiency of the traditional traffic classification method, deep learning method is adopted to manage the classification of the four kinds of traffic [5, 28]. As deep learning has achieved great successes in many classification problems [2, 39], we apply deep learning to the problem of media traffic classification. Specifically, two different deep learning modules, Multilayer Perceptron (MLP)

ICCDA' 2019, March 14-17, 2019, Kahului, Hawaii, USA

ACM ISBN 978-1-4503-6634-2.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[12] and Convolutional Neural Network (CNN) [20] is used in our traffic classification system. Both of which apply back propagation [34] to optimize the model parameters so that the practical output can approximate the ideal value as much as possible. A combination of packet level features and flow level features are employed to improve the learning performance. The performance of the two deep learning methods in traffic classification are compared in the evaluation. Evaluation results show that a very high classification accuracy can be achieved by using the deep learning method with the combined features. The precision and recall are also satisfactory. We find that MLP is very good at the classification of picture, text and video traffic while CNN has a preference for audio traffic. The relationship between the training time of the neural network for traffic classification and the neural network's architecture is also revealed at the end of the paper.

2 RELATED WORK

Traffic classification has been a hot topic since a long time ago. There has been a plenty of work on the problem. Previous work can been generally divided into three different classes, i.e., port based method [5, 17, 26, 28], payload based method [8, 9, 13, 18, 46] and statistic or machine learning based method [1, 24, 30, 32, 37, 38, 41–43, 45].

Port based method is effective in the early time because the port number is assigned by Internet Assigned Numbers Authority (IANA) [14] and the number is corresponding to packet header information, people can classify internet traffic by the assigned port number from the packet header. This method can achieve very high classification accuracy. However, with the increase of internet complexity, port number is no longer invariable and is often assigned randomly. Moreover, the port number is even disguised to avoid the attacks from internet in many cases. Port number based method cannot provide effective traffic classification nowadays [28].

Payload based method is also known as Deep Inspection (DI). It compares the payload of the internet traffic to a certain pattern which is corresponding to a certain kind of traffic [9]. If payload matches pattern, then the traffic can be classified to the certain class. This method can obtain quite high classification accuracy for the unencrypted traffic. Nonetheless, more and more internet traffic tend to be encrypted for the consideration of confidentiality. Payload based method cannot work for the classification of encrypted traffic. Moreover, some internet traffic has the choice of protocol encapsulation, which also leads to the ineffectiveness of payload based traffic classification. On the other side, payload based method needs to look into the packet content which also introduces the problem of privacy. The computation complexity is also very high since the number of traffic patterns is very large, therefore, it cannot handle the high speed traffic and large number flows [5].

Another widely use method for traffic classification is machine learning, or statistics based method. Machine learning has proven its effectiveness in many areas such as data mining, computer vision, bioinformatics and pattern recognition [2, 7, 21, 39]. A number of researchers bring machine learning to the classification of internet traffic [24, 32]. Machine learning generally includes two stages: training stage and test stage. It first extracts useful features from the source data and feed these features to the learning module in the training stage. After sufficient times of training, a mature learning module is formulated. In the testing stage, input the testing data to the trained module and one can get the classification result. A prosperous research direction here is to multiple layer artificial neural network as the learning module, this method is widely known as deep learning.

[32] conducts the traffic classification over SDN using machine learning and try to provide internet operators with useful forecasts and monitoring. [45] designs a QoS (Quality of Service) aware architecture to classify the traffic in SDN, the proposed architecture combines machine learning and DI. DI is used to maintain a dynamic traffic database and periodically re-training can be done at the learning stage so that the authors can obtain a good classification accuracy. [24] presents *deep packet* to identify encrypted traffic and it can classify internet traffic to major classes as well as end-user applications. In addition, *deep packet* can distinguish VPN and non-VPN traffic. [1] introduces Support Vector Machine (SVM) and Naive Bayesian method to classify the network traffic. Statistical features and the information of flow correlation are used to improve the performance of traffic classification. [30] uses the sub-flows information to train the machine learning model, the sub-flows are all extracted from the original full flows. In this way, the authors are not possible to miss packets from the start of flows. The authors apply Decision Tree and Naive Bayes method to classify the IP Traffic. Besides, this method do not need to know the direction of flow. [42] adopts Stacked AutoâĂŘEncoder (SAE) to deal with feature learning and feature selection automatically, SAE can manage unlabeled data in the training thus is very popular in the unsupervised learning. The authors perform this method on 25 common protocols and achieve good classification results. The paper also considers the identification of unknown protocols. Above all the works, traditional machine learning method such as SVM or Bayesian method are easy to implement and widely used in the classification of different protocols. On the other side, deep learning method like CNN or SAE adopt the artificial neural network as the learning module and achieve good performance in the classification of specific kind of traffic [24].

3 METHODOLOGY

In recent years, researchers has witnessed the the strong ability of deep learning in many areas. Plenty of excellent work has been produced with the help of deep learning [2, 39]. Indicated by the success of deep learning, in this work, we apply the deep learning method to the classification of four different kinds of traffic, i.e., audio traffic, picture traffic, text traffic, video traffic. We design a traffic classification system that uses both the packet level and flow level features to improve the classification performance. Firstly, we collect the four kinds of traffic data from the real network environment. Then, we excavate useful features in the raw data, we note it as the preprocessing of the dataset. Thirdly, after the preprocessing, the extracted features are fed to train the learning module. Two different deep learning modules are adopted, MLP and CNN. We do not apply traditional machine learning method such as SVM here because it has difficulties in dealing with high dimension data and

Table 1: Packet Level Features.

Features	Description
minfps	Minimum forward packet size
minbps	Minimum backward packet size
maxfps	Maximum forward packet size
maxbps	Maximum backward packet size
meanfps	Mean forward packet size
meanbps	Mean backward packet size
medianfps	Median forward packet size
medianbps	Median backward packet size
stdfps	Standard deviation of forward packet size
stdbps	Standard deviation of backward packet size
minfpt	Minimum forward packet inter-arriving time
minbpt	Minimum backward packet inter arriving time
maxfpt	Maximum forward packet inter arriving time
maxbpt	Maximum backward packet inter arriving time
meanfpt	Mean forward packet inter arriving time
meanbpt	Mean backward packet inter arriving time
medianfpt	Median forward packet inter arriving time
medianbpt	Median backward packet inter arriving time
stdfpt	Std deviation of forward inter arriving time
stdbpt	Std deviation of backward inter arriving time
fprt	forward packet arriving time
bprt	backward packet arriving time
fpft	forward packet finishing time
bpft	backward packet finishing time

often turns out poor performance [1]. Fourthly, the trained learning modules are used to classify the traffic to different classes.

3.1 Data Collection

We collect the four kinds traffic data from the real network environment. 5-tuple (source IP, destination IP, source port, destination port, protocol) information is used to specify a traffic flow. As the concerned flows have two directions, we separate the flow with the forward flow and backward flow from each other. The forward and backward flow make a pair traffic. The two-directional flows are both used to extract useful features.

As described in the formal section, we obtain the raw traffic from the real network environment and use the refined data to train and test the designed neural network based traffic classification system. As the extracted features are various in very large scale by the numerical measurement, we need to preprocess the feature data before feeding it to the neural network. We deal with the feature data under normalization. Normalization is applied here to make sure that the feature data be processed under the same scale. Max-Min normalization [15] is adopted in our method. Then the normalized data is used to train and test the neural network.

Once the normalization of the data is completed, we separate the whole data into two sets, i.e., the training set and the testing set. To make the classification results more convincing, we apply 10-fold cross validation to deal with the dataset. Specifically, the dataset is partitioned into 10 parts. Then 9 parts of the data are used for training and the last 1 part is used for testing, in this way we can obtain a classification result. Repeat this progress until every part of the dataset has been used for testing. Lastly, we compute the average the 10 times' testing results and obtain the final classification result.

Table 2: Flow Level Features.

Features	Description
numpf	Number of packets in forward flow
numpb	Number of packets in backward flow
numbf	Number of bytes in forward flow
numbb	Number of bytes in backward flow
numppsf	Number of packets per second in forward flow
numppsb	Number of packets per second in backward flow
numbpsf	Number of bytes per second in forward flow
numbpsb	Number of bytes per second in backward flow
srcip	Source IP address
dstip	Destination IP address
srcport	Source port
dstport	Destination port
prot	Protocols

3.2 Feature Selection

As the collected traffic raw data cannot be directly fed into the deep learning modules like images, we need to extract useful features from the raw data.

Unlike other works that target on traffic classification use only packet level features such as packet number, packet length, we adopt both the packet level features as well as the flow level features to improve the classification performance. Flow level features such as flow size can describe the traffic more directly in the classification of whether the traffic is video or picture. We have also observed that the traffic classification accuracy increased largely since we add the flow (sub-flow) level features.

Since we have obtained the flows of two directions (forward and backward) in the data preprocessing stage, we can extract the flow (sub-flow) features in both the two directions. For example, in the packet level, for the forward direction flows, we calculate the size of the packets as an important feature because it can reflect the traffic type to a certain degree. We also calculate the same features for the backward flows. Besides the packet size, we find that packet arriving time and packet finishing time are also important features in the classification. For different types of traffic, the packet arriving time and packet finishing time could be various, and both of which are counted in the forward direction and the backward direction. Apart from the packet arriving time and finishing time, we also employ the packet inter arriving time as the classification feature. Packet inter arriving time means the time gap between two adjacent packets. For different kinds of traffic, the time gap between packet could be quite different, which makes the packet inter arriving time plays an important role in the traffic classification of our design. The packet inter arriving time is also calculated in both the forward direction and backward direction.

On the other hand, at the flow level, we have also found out a number of important features in the classification. For example, the number of packets in a flow. As we have observed that the number

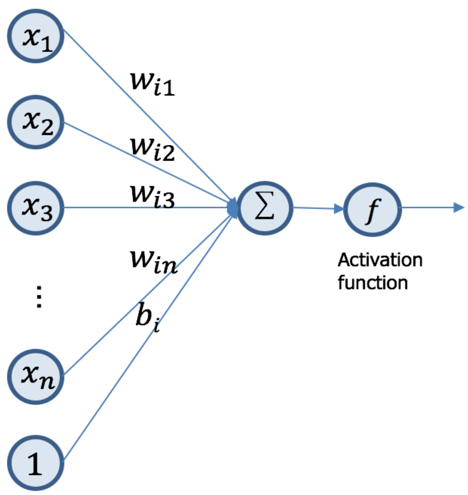

Figure 1: The architecture inside a neuron.

of packets is diverse in a certain flow, therefore, the number of packets is chosen as an important feature. Moreover, we also notice that the number of bytes of a flow also improves the classification accuracy as a feature. Furthermore, the number of packets per second and number of bytes per second are also used as the features in our design. The number of packets and number of bytes are both calculated in the forward flow direction and the backward flow direction, and so are the packets per second and number of bytes per second.

For the features such as packet size and packet inter arriving time in the packet level, we obtain the minimum, maximum, mean, median, standard deviation value of it. For instance, the minimum packet inter arriving time and maximum packet inter arriving time as well as the mean packet inter arriving time and median packet inter arriving time, and the standard deviation of the packet inter arriving time are calculated as the classification features. We also adopt the 5-tuple information as the input features in the flow level. We summarize the packet level features and flow level features used in the classification in Table 1 and Table 2 respectively.

3.3 Learning Modules

Since MLP and CNN have shown great classification performance in many other areas [2, 39], we develop two traffic classification methods that based on MLP and CNN respectively by using both the packet level features and the flow level features [12, 20]. MLP is a feedforward network that made up of multiple layers of neurons. Generally, MLP has one input layer, one output layer and at least one hidden layer or middle layer. Every neuron in one layer is fully connected to the next layer's neurons. The architecture inside a neuron is shown in Figure 1. Assume that the values of the current layer's neurons are denoted as $\boldsymbol{x} = (x_1, \ldots, x_n)^T$, the weights vector and offsets vector between the current layer and the next layer's ith neuron are $\boldsymbol{w_i} = (w_{i1}, \ldots, w_{in})$ and $\boldsymbol{b_i}$ respectively. Then

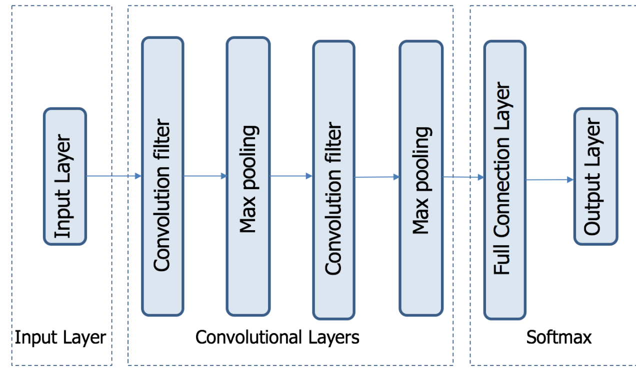

Figure 2: CNN based traffic classification module.

the next layer's ith neuron's value can be represented as

$$h_i(\boldsymbol{x}) = f(\boldsymbol{w_i x} + b_i). \tag{1}$$

Where $f()$ is the activation function, it could be *Logistic Function* $\sigma(x) = \frac{1}{1+\exp(-x)}$, or *Tanh Function* $\tanh(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$, or other activation functions such as *Rectified Linear Unit* (ReLU) $ReLU(x) = \max(0, x)$.

At the output layer we compare the predicted value of the neural network and the practical value, and use the *Cross Entropy* [6] as the cost function. Back propagation [34] is applied to train the neural network and obtain the optimal parameters that minimize the cost function.

Another learning module we apply is CNN. CNN is a kind neural network that contains the layer of convolutions. Suppose the value of current layer is denoted as $\boldsymbol{X}_i$, the kernel of the convolutional layer is denoted as $\boldsymbol{W}_i$, the activation function is ReLU, then output value of the convolutional layer can be represented as

$$\boldsymbol{Y}_i = ReLU(\boldsymbol{X}_i * \boldsymbol{W}_i). \tag{2}$$

Behind the convolutional layer, a pooling layer is there to enhance the neural network. Pooling is important to increase the stability of the neural network in case that a tiny change of the input may cause a significant difference in the output. The introduce of pooling can also reduce the computation complexity. Moreover, pooling can avoid overfitting in the learning progress. Here, we use max pooling which can be represented as

$$maxpooling\begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix} = \max(x_1, x_2, x_3, x_4). \tag{3}$$

The framework of CNN for traffic classification is shown in Figure 2 in our design. After the two layers of convolution and two layers of pooling, a full connection layer is "fully connected" to the output layer. Like MLP, *Cross Entropy* is chosen as the cost function, also back propagation is applied to train the neural network so that the value of cost function is minimized.

3.4 System Design

The designed traffic classification system collects traffic data from the real network environment and gives out the classification results at the output end. Then, meaningful management of the network could be done according to the traffic classification results, which is beyond the scope of this paper and will be left for future work. It should be paid attention that, after the collection of the network traffic, the raw data cannot be directly fed into the neural network for classification because the traffic data is not suitable to be formed in a matrix format. Therefore, the raw traffic data need to be preprocessed in advance. We refine the flow information which is specified by the 5-tuple. Then useful features as described in Subsection 3.2 are extracted from the flow information. We train the neural network with the extracted features and obtain the traffic classification results by feeding the trained neural network with the testing data.

4 EVALUATIONS

In this section, we present the experiments' results in detail. In order to measure the classification performance, we need to employ several metrics [35] to evaluate the algorithm. Basically, we have four types of internet traffic, i.e., audio, picture, text and video, take the identification of the video traffic for the example. Here we have 4 different decision scenarios, True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). TP means the video traffic is correctly identified as the video kind, TN means the the non-video traffic is not identified as the video kind, FP means the non-video traffic is identified as the video kind, FN means the video traffic is identified as the non-video kind. Accordingly, the measuring metrics are given as follows

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{4}$$

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{5}$$

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{6}$$

Accuracy reflects the overall effectiveness of the classifier, *Precision* specifies the rate that the correctly identified samples (TP) in the samples that are practically identified (TP+FP), *Recall* specifies the rate that the the correctly identified samples (TP) in the samples that should be identified (TP+FN).

Furthermore, we represent the harmonic mean of *Precision* and *Recall* as F_1, which is given as follows

$$\begin{aligned} \frac{2}{F_1} &= \frac{1}{P} + \frac{1}{R}, \\ F_1 = \frac{2P * R}{P + R} &= \frac{2TP}{2TP + FP + FN}. \end{aligned} \tag{7}$$

Where P is the *Precision* value, and R is the *Recall* value.

We have four different types of traffic in the dataset, audio, picture, text and video. The designed MLP based and CNN based traffic classification method are performed on the collected dataset. *Accuracy*, *Precision*, *Recall* and F_1 value are plotted out to demonstrate the effectiveness of the proposed traffic classification method. To reveal the intrinsic mechanism of the neural network, we further exhibit the relationship that how is the training time varying with the number of neurons of MLP.

We have found in the experiments that both the MLP based traffic classification and CNN based traffic classification achieve very good performance. It should be noticed that MLP has two hidden layers with 128 neurons and 512 neurons respectively, CNN has two

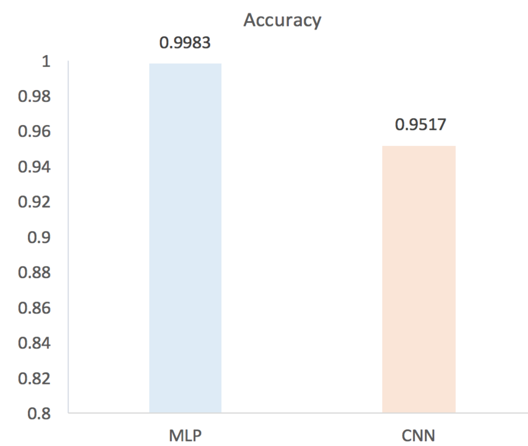

Figure 3: The traffic classification accuracy of the MLP based method and CNN based method.

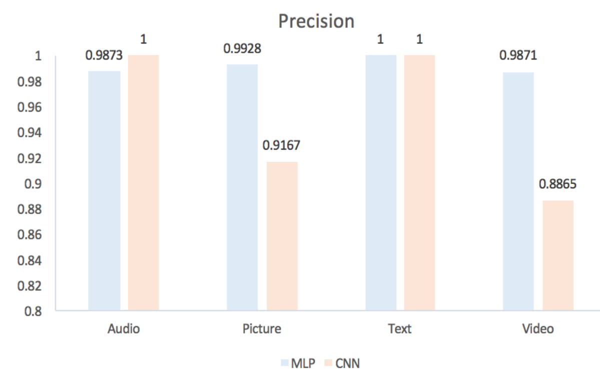

Figure 4: The traffic classification precision of the MLP based method and CNN based method.

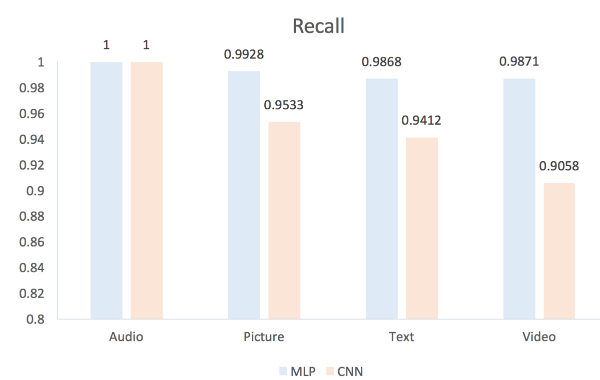

Figure 5: The traffic classification recall of the MLP based method and CNN based method.

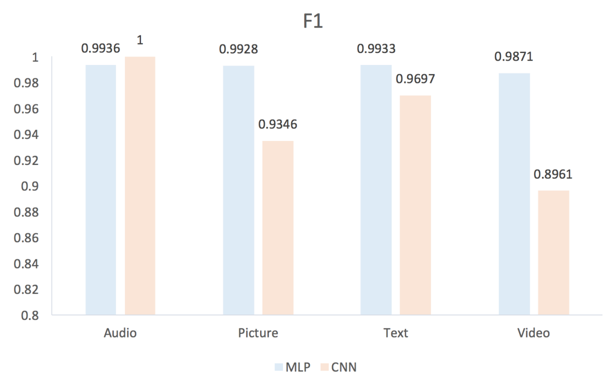

Figure 6: The traffic classification F_1 of the MLP based method and CNN based method.

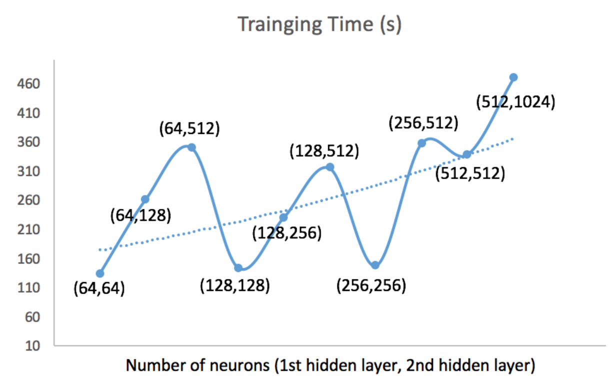

Figure 7: The training time of different MLP architectures to achieve the accuracy of 95%.

convolutional layers with 8 and 16 kernels and a fully connected layer with 128 neurons. MLP based traffic classification method obtains the classification accuracy of as high as more than 99%, the accuracy of CNN based method is about 95%. The classification accuracy is presented in Figure 3. The training time for MLP and CNN is 314s and 611s respectively. At the testing stage, the classification results are obtained within 0.019s and 0.022s by the MLP based method and CNN based method respectively.

The *Precision* and *Recall* as well as F_1 value are also satisfactory. The *Precision* and *Recall* result can be found in Figure 4 and Figure 5. It can be observed that the *Recall* of MLP based traffic classification method has outperformed that of CNN based method for all kinds of the concerned traffic. As for *Precision*, MLP based method is also better than CNN based method except for the audio traffic. Because MLP based method does not have a convolutional layer so that it is easier to train a MLP to obtain a good classification performance. However, as CNN based method adopts max pooling to improve stability so it is more suitable in the classification of traffic that has correlation information, which is more likely to appear in audio traffic. Therefore, CNN based method has a preference for the classification of audio traffic. With regard to picture and video traffic, the *Precision* of CNN based method is about 91% and 88% respectively, much less than that of MLP based method. CNN is not suitable to be used in the classification of picture and video traffic but it can make quite a difference in the classification of audio traffic.

Furthermore, F_1 value is presented in Figure 6. As F_1 harmonizes the values of *Precision* and *Recall* and can reflects the performance of the classification methods more directly. It can be found that the F_1 value of MLP based method exceeds that of CNN in all traffic types except for audio. This is because CNN based method is especially good at the classification of audio traffic as described above.

The training time of MLP is revealed in Figure 7. We count the training time of MLP in the condition that the neural network can achieve 95% of classification accuracy at the testing stage. We conduct the experiments under a number of different neural network architectures. Specifically, we vary the number of neurons of MLP, all the tested MLP architectures have two hidden layers. We denote the number pair (a, b) as the number of neurons of the first hidden layer and the second hidden layer. For example, $(64, 128)$ means the first hidden layer has 64 neurons and the second hidden layer has 128 neurons. Basically, it could be observed that the training time increases with the number of neurons in large-scale. However, we notice the training time is in trough of wave when the two hidden layers has the same number of neurons and is in wave crest if the number of the second hidden layer is about four times that of the first layer's. For the scenario that the number of neurons of the second hidden layer is about two times that of the first layer's, the training time is closely distributed around the trend curve (dotted line). It is meaningful that if we want to reduce the training time in the traffic classification when applying MLP based method, we can try to make the hidden layers have about the same number of neurons in the condition that the learning architecture can achieve the satisfactory performance.

5 CONCLUSIONS

In this paper, we apply deep learning method to the classification of four different types of media traffic (audio, picture, text and video) and provide precise classification accuracy to the four kinds of traffic. We use both the packet level features and flow level features to enhance the classification results. We collect the traffic data from the real network environment and design two deep learning based methods (MLP based method and CNN based method) to classify the target traffic. The designed learning architectures can achieve satisfactory classification performance. MLP has outperformed CNN in the classification of picture, text and video traffic, CNN is very good at the classification of audio traffic. Moreover, we have found that the training time can be reduced if the number of neurons of the hidden layers are close. As a matter of fact, we have not applied neural network with more that two layers in the consideration of computation complexity. More complicated neural network architecture for traffic classification and computation method to improve the processing time are left for future work.

ACKNOWLEDGMENTS

The authors would like to thank the colleagues in NSLab Tsinghua University for productive discussions. The authors would also like to thank the anonymous referees for their valuable comments and helpful suggestions. This work is supported by National Natural

Science Foundation (No. 61872212) and National Key Research and Development Program (No.2016YFB1000101).

REFERENCES

[1] R. Aggarwal and Nanhay Singh. 2017. A New Hybrid Approach for Network Traffic Classification Using Svm and Naïve Bayes Algorithm.

[2] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. 2015. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology* 33, 8 (2015), 831.

[3] Mina Tahmasbi Arashloo, Yaron Koral, Michael Greenberg, Jennifer Rexford, and David Walker. 2016. SNAP: Stateful network-wide abstractions for packet processing. In *Proceedings of the 2016 ACM SIGCOMM Conference*. ACM, 29–43.

[4] Tom Auld, Andrew W Moore, and Stephen F Gull. 2007. Bayesian neural networks for internet traffic classification. *IEEE Transactions on neural networks* 18, 1 (2007), 223–239.

[5] Alberto Dainotti, Antonio Pescape, and Kimberly C Claffy. 2012. Issues and future directions in traffic classification. *IEEE network* 26, 1 (2012).

[6] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. 2005. A tutorial on the cross-entropy method. *Annals of operations research* 134, 1 (2005), 19–67.

[7] Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 69–78.

[8] Alessandro Finamore, Marco Mellia, Michela Meo, and Dario Rossi. 2010. Kiss: Stochastic packet inspection classifier for udp traffic. *IEEE/ACM Transactions on Networking (TON)* 18, 5 (2010), 1505–1515.

[9] Michael Finsterbusch, Chris Richter, Eduardo Rocha, Jean-Alexander Muller, and Klaus Hanssgen. 2014. A survey of payload-based traffic classification approaches. *IEEE Communications Surveys & Tutorials* 16, 2 (2014), 1135–1156.

[10] Open Networking Fundation. 2012. Software-defined networking: The new norm for networks. *ONF White Paper* 2 (2012), 2–6.

[11] Bo Han, Vijay Gopalakrishnan, Lusheng Ji, and Seungjoon Lee. 2015. Network function virtualization: Challenges and opportunities for innovations. *IEEE Communications Magazine* 53, 2 (2015), 90–97.

[12] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks* 2, 5 (1989), 359–366.

[13] Nan Hua, Haoyu Song, and TV Lakshman. 2009. Variable-stride multi-pattern matching for scalable deep packet inspection. In *INFOCOM 2009, IEEE*. Citeseer, 415–423.

[14] IANA. [n. d.]. https://www.iana.org/. ([n. d.]).

[15] T Jayalakshmi and A Santhakumaran. 2011. Statistical normalization and back propagation for classification. *International Journal of Computer Theory and Engineering* 3, 1 (2011), 1793–8201.

[16] Chengjun Jia, Zhe Fu, Xiaohe Hu, Shui Cao, Liang Wang, and Jun Li. 2018. Multi-core HTB for bandwidth sharing. In *Proceedings of the 2018 Symposium on Architectures for Networking and Communications Systems*. ACM, 169–171.

[17] Thomas Karagiannis, Andre Broido, Nevil Brownlee, Kimberly Claffy, and Michalis Faloutsos. 2003. File-sharing in the Internet: A characterization of P2P traffic in the backbone. *University of California, Riverside, USA, Tech. Rep* (2003).

[18] Hyang-Ah Kim and Brad Karp. 2004. Autograph: Toward Automated, Distributed Worm Signature Detection.. In *USENIX security symposium*, Vol. 286. San Diego, CA.

[19] Alok Kumar, Sushant Jain, Uday Naik, Anand Raghuraman, Nikhil Kasinadhuni, Enrique Cauich Zermeno, C Stephen Gunn, Jing Ai, Björn Carlin, Mihai Amarandei-Stavila, et al. 2015. BwE: Flexible, hierarchical bandwidth allocation for WAN distributed computing. In *ACM SIGCOMM Computer Communication Review*, Vol. 45. ACM, 1–14.

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[21] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 609–616.

[22] Zhi Liu, Shijie Sun, Ju Xing, Zhe Fu, Xiaohe Hu, Jianwen Pi, Xiaofeng Yang, Yunsong Lu, and Jun Li. 2018. MN-SLA: a modular networking SLA framework for cloud management system. *Tsinghua Science and Technology* 23, 6 (2018), 635–644.

[23] Zhi Liu, Xiang Wang, Weishen Pan, Baohua Yang, Xiaohe Hu, and Jun Li. 2015. Towards efficient load distribution in big data cloud. In *2015 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 117–122.

[24] Mohammad Lotfollahi, Ramin Shirali, Mahdi Jafari Siavoshani, and Mohammdsadegh Saberian. 2017. Deep Packet: A Novel Approach For Encrypted Traffic Classification Using Deep Learning. *arXiv preprint arXiv:1709.02656* (2017).

[25] Alok Madhukar and Carey Williamson. 2006. A longitudinal study of P2P traffic classification. In *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium on*. IEEE, 179–188.

[26] Alok Madhukar and Carey Williamson. 2006. A longitudinal study of P2P traffic classification. In *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium on*. IEEE, 179–188.

[27] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review* 38, 2 (2008), 69–74.

[28] Andrew W Moore and Konstantina Papagiannaki. 2005. Toward the accurate identification of network applications. In *International Workshop on Passive and Active Network Measurement*. Springer, 41–54.

[29] Andrew W Moore and Denis Zuev. 2005. Internet traffic classification using bayesian analysis techniques. In *ACM SIGMETRICS Performance Evaluation Review*, Vol. 33. ACM, 50–60.

[30] T Nguyen and Grenville Armitage. 2006. Synthetic sub-flow pairs for timely and stable IP traffic identification. In *Proc. Australian Telecommunication Networks and Application Conference*.

[31] Salima Omar, Asri Ngadi, and Hamid H Jebur. 2013. Machine learning techniques for anomaly detection: an overview. *International Journal of Computer Applications* 79, 2 (2013).

[32] Mohammad Reza Parsaei, Mohammad Javad Sobouti, Seyed Raouf Khayami, and Reza Javidan. 2017. Network traffic classification using machine learning techniques over software defined networks. *International Journal of Advanced Computer Science and Applications* 8, 7 (2017), 220–225.

[33] RK Rahul, T Anjali, Vijay Krishna Menon, and KP Soman. 2017. Deep learning for network flow analysis and malware classification. In *International Symposium on Security in Computing and Communication*. Springer, 226–235.

[34] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature* 323, 6088 (1986), 533.

[35] Muhammad Shafiq and Xiangzhan Yu. 2017. Effective packet number for 5G IM WeChat application at early stage traffic classification. *Mobile Information Systems* 2017 (2017).

[36] Yiyang Shao, Yibo Xue, and Jun Li. 2014. PPP: Towards parallel protocol parsing. *China Communications* 11, 10 (2014), 106–116.

[37] Yiyang Shao, Baohua Yang, Jingjie Jiang, Yibo Xue, and Jun Li. 2014. Emilie: Enhance the power of traffic identification. In *2014 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 31–35.

[38] Yiyang Shao, Luoshi Zhang, Xiaoxian Chen, and Yibo Xue. 2014. Towards time-varying classification based on traffic pattern. In *2014 IEEE Conference on Communications and Network Security*. IEEE, 512–513.

[39] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[40] Pu Wang, Shih-Chun Lin, and Min Luo. 2016. A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs. In *2016 IEEE International Conference on Services Computing (SCC)*. IEEE, 760–765.

[41] Pan Wang, Feng Ye, Xuejiao Chen, and Yi Qian. 2018. Datanet: Deep learning based encrypted network traffic classification in sdn home gateway. *IEEE Access* 6 (2018), 55380–55391.

[42] Zhanyi Wang. 2015. The applications of deep learning on traffic identification. *BlackHat USA* (2015).

[43] Baohua Yang, Guangdong Hou, Lingyun Ruan, Yibo Xue, and Jun Li. 2011. Smiler: Towards practical online traffic classification. In *2011 ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems*. IEEE, 178–188.

[44] Baohua Yang, Guodong Li, Yaxuan Qi, Yibo Xue, and Jun Li. 2010. DFC: Towards Effective Feedback Flow Management for Datacenters. In *2010 Ninth International Conference on Grid and Cloud Computing*. IEEE, 98–103.

[45] Changhe Yu, Julong Lan, JiChao Xie, and Yuxiang Hu. 2018. QoS-aware Traffic Classification Architecture Using Machine Learning and Deep Packet Inspection in SDNs. *Procedia computer science* 131 (2018), 1209–1216.

[46] Zhenlong Yuan, Yibo Xue, and Yingfei Dong. 2013. Harvesting unique characteristics in packet sequences for effective application classification. In *Communications and Network Security (CNS), 2013 IEEE Conference on*. IEEE, 341–349.