Spectrum Based Wireless Radio Traffic Classification using Hybrid Deep Neural Network

Md. Habibur Rahman, *Student Member, IEEE,* Raihan Bin Mofidul,
and Yeong Min Jang, *Member, IEEE*
Department of Electronics Engineering, Kookmin University, Seoul 02707, Korea
Email: rahman.habibur@ieee.org; raihanbinmofidul@kookmin.ac.kr;
yjang@kookmin.ac.kr

***Abstract*—In recent years, traffic classification (TC) represents an important issue in managing and optimizing the wireless network capacity. With the growth of numerous wireless technologies, it has become more challenging to develop an efficient TC system. Deep learning (DL) based architecture provides feasible solution in today's complex and modern scenarios where even traffic is encrypted. Traditional TC using DL based architecture exploits the byte/protocol representation of the packet at the link layer (L2) or above on the same radio network domain. It limits the efficacy of the TC systems in wireless networks using shared spectrum. Therefore, designing TC based on spectrum band generated physical layer (L1) packet using DL based architecture has received significant research attention more recently. In this article, we propose a deep hybrid neural network that incorporates a deep convolutional network with a recurrent network to classify traffic at layer 7 (L7) (e.g., application characterization and application identification) of the radio network stack using L1 packets. The proposed network can capture spatio-temporal feature correlation and use multiscale feature map to avoid vanishing gradient problem. From the simulation, it is seen that the proposed classifier can achieve 98.25% accuracy and 86.28% accuracy for the task of application characterization and application detection, respectively. Simulation results unveil that our proposed network is very promising for classifying traffic at L7 using the L1 packet.**

***Index Terms*—Network traffic, radio spectrum, hybrid deep neural network, CNN, RNN.**

I. Introduction

With the rapid evolution of network traffic diversity, automatic traffic recognition and classification has become one of the indispensable task of network monitoring service (NMS). Automatic traffic analysis can provide insight to network operators specific security and quality of services (QoS) for enforcing on the analyzed traffic [1]. At present, tremendous growth of complementary wireless technologies is playing a leading role to produce huge amount of traffic offering access to millions of users and machines and hence understanding the network traffic pattern has turned into more burdensome. In addition, encryption of the traffic adds more difficulty in achieving high quality traffic classification (TC) along with securing high QoS to the users [2].

Traditional approaches for traffic detection mainly includes Port Number Based method and the Data Packet Inspection (DPI) Based method collecting information from the NMS which shows very poor performance in complex and modern traffic environment [3]. Deep learning (DL) architectures including convolutional neural network (CNN) and recurrent neural network (RNN) has provided feasible solution with outstanding achievements in wide-ranging fields from natural language processing and computer vision to communication and bio informatics. More recently, DL architecture is potentially studied and applied for various purpose in the network optimization and management area that includes, network state prediction, anomaly detection, cyber-attack detection, and TC. These DL architectures can significantly outperform the traditional methods for traffic analysis task.

Focusing on TC, several approaches based on DL have been designed and studied using Link Layer (L2) or above layers' packet flows. An end-to-end TC algorithm based on CNNs that converts raw traffic into images is proposed in [4] and [5]. The authors of [4] have used the several packets of the traffic flows to extract time series features and exploits them to identify the application or protocol type generated by them. A comparison of performance of the proposed Seq2Img model against four popular classifiers such as Supported Vector Machine (SVM), Multi-Layer Perceptron (MLP), Naive Bayes (NB), and Decision Tree (DT) are shown in [4]. Experimental results show that their proposed Seq2Img is almost 12% more accurate than other models when classifying applications. However, training such DL based network requires a large labeled dataset. A semi-supervised approach that pre-trains a 1DCNN model on an unlabeled dataset to infer traffic patterns is proposed in [6] to overcome the problem of data labeling. In their experiment, the used dataset contains time-series features of a fixed number of sampled packets from traffic flows. In [7], a comparative analysis among Dl based classifiers, e.g. a Stacked Denoising Auto Encoder (SDAE), a CNN, and a Long Short Term Memory (LSTM) and machine learning based classifier, e.g. Random Forest (RF) is presented. This study shows that Dl model can learn more insightful features and perform over 20% better than RF in terms of accuracy. A combination of two CNN layers followed by one LSTM layer with two fully connected layers at the end was proposed in [8]. Time series features are extracted from the headers of the first 20 packets exchanged during the flow lifetime and used to train the proposed network. To ensure data confidentiality

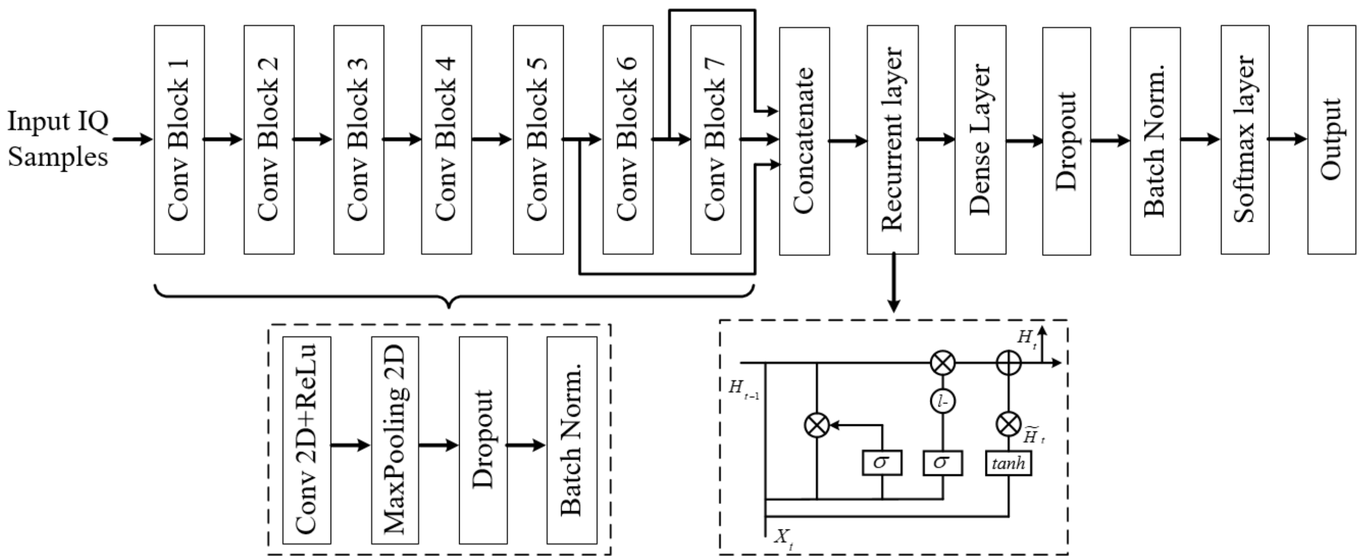


Fig. 1. architecture of the Proposed classifier.

users identity (MAC/IP addresses) was removed from the data packet. However, results suggest that the combination obtained the best results in terms of accuracy and F1-score. Performance of a semi-supervised model based on a Variational Auto Encoder (VAE) is studied to traffic detection in [9]. The authors exploits transformed image of data from the HTTP sessions (requests and responses) to train the network. The proposed VAE includes an MLP encoder and decoder as feature extractor from that images in an unsupervised manner.

However, all these aforementioned studies have considered the traffic from same network domain and the traffic data packet was taken from the L2 (or above). But, when users shared the same radio spectrum, users' traffic from one wireless network domain can be negatively impacted by users' traffic transmissions from other wireless networks which results in poor classification performances. Recently, some researchers have started working on physical layer (L1) packets to perform TC as a solution. The authors in [10] investigated the performance of RNN network in classifying the spectrum data. But, their achieved result is not satisfactory compared to byte-based TC systems. In [11], the authors have presented a traffic recognition approach directly from time-frequency image of the radio spectrum. Their approach can achieve an accuracy of $\geq 96\%$ on their generated data and outperform state-of-the-art methods based on IP-packets with DL. A novel framework to achieve TC at any layer on the radio network stack has been developed in [12]. They compared the performances of two spectrum-based DL-based classifiers, e.g. a novel Convolutional Neural Network (CNN) and a Recurrent Neural Networks (RNN) for traffic classification. But, both of the classifiers are not suitable in terms of time cost and computational memory.

In this paper, we proposed a cost-efficient hybrid deep neural network combining recurrent neural network with CNN to perform TC at layer 7 (L7) (e.g., application identification and application characterization) spectrum band generated L1 packet. Compared to the similar work in [12], the proposed network is capable of extracting spatio-temporal features from the in-phase and Quadrature (IQ) samples of L1 packet data and exploit them to achieve high accuracy with less parameters, less memory, faster execution of the training. Besides, we have introduced skip connection in the network to preserve residual information and prevent the vanishing gradient problem. To the end of this paper, simulation results have been demonstrated to unveils the efficacy of our proposed network.

II. Spectrum Based Traffic Classification Methodology

The problem of traffic recognition at L7 using spectrum data from the L1 packet can be formulated as: given a representation of the spectrum $x \epsilon X$ where X is set of raw IQ samples, the goal of a traffic classifier is to predict the class label of traffic at L7 taking x as input and inform NMS for proceeding to analysis of the traffic flowing in the physical medium. However, performing TC exploiting L1 packets is very arduous because of the heterogeneity (e.g., various modulation scheme or various packet length for the the same payload). The overall procedures of TC comprises several steps.

When traffic generators, i.e. use any software or any applications on their wireless devices, they generate traffic and transmits it to the gateway [12]. These generated traffic includes a distinctive pattern at any radio stack level. Our proposed DL based classifiers learn recognition of the traffic of L7 capturing that pattern. After obtaining the raw spectrum samples, the samples are normalized and grouped before being fed to the DL model. The raw spectrum consists of IQ samples which can be corrupted by noise and interference. Therefore, filtering techniques are adopted to remove the effects of noise and interference. Afterwards, the samples are zero padded for short sequences and truncated for long sequence to normalize the length of all L1 packets to a given fix value. To train our proposed classifier, this task is very important. finally this fixed length data sequences are used to train and optimize the proposed architecture. Once, the proposed classifier are optimized, then this classifier become ready to perform TC at layer 7. At L7, the same classifier can perform two types of classification with high accuracy. The first type of classification task is coarse-grained task where the classifier determine the type of application inside the transmitted packet (e.g., audio or video). In contrast, the second type classification task is fine-grained task where the goal of the classifier is to discriminate between the actual applications generating the L7 traffic.

A. Network Architecture

The architecture of the proposed classifier is illustrated in Fig. 1. The classifier comprises seven "Conv Block", each blocks is composed of 2D convolutional layers (conv) where the convolution layers are all composed of 64 filters, with a kernel size of 32×1 and the rectified linear unit (ReLu) activation function, a 2×2 maxpooling layer, which partly helps to prevent overfitting and reduce the amount of parameters and computation in the network. A dropout layer with drop rate of 0.1 and also a batch normalization layer is inserted after convolutional and maxpooling layers. A dropout layer can prevent vanishing gradient problem in some scale and the batch normalization helps to enhance the classifier training speed through normalization of the layers inputs.

Followed by the CNN block, a RNN layer is inserted to detect temporal changes within a given IQ sequence and to reduce the spectral variance.The RNN can remember previous inputs using a cell mechanism. It allows our model to create and recall a complex history of traffic pattern. More specifically, we employed a GRU network (i.e., a special RNN architecture), which has the ability to learn long-term dependencies [13]. The GRU layer is constructed using 20 units. The features extracted from sixth and seventh are concatenated to generate input of the GRU layer.

The final part of the classifier is made up of a dense layer, dropout layer, batch normalization layer, softmax layer, and final dense layer. Theses layers are adept at mapping features into a separable space, and are used in most CNN models for classification tasks. The dense layer is made up of 1024 units with a ReLu activation function. The drop out rates of the dropout layer is also 0.1. A final dense layer with -classes number of neurons followed by a softmax layer to produce a higher-order feature representation into the n classes of traffic IQ samples.

However, the total parameter of the proposed classifier is 8.3×10^5 where the no. of trainable parameters are 8.27×10^5.

TABLE I
SAMPLE DISTRIBUTION PER CLASSIFICATION TASK.

Classification type 1	No. of samples	Classification type 2	No. of samples	Total samples
Audio	39053	Spotify	13822	140665
		Tunein	10229	
		Gpodcast	15002	
Video	56253	Youtube	16671	
		Netflix	18268	
		Twitch	21314	
No-app Type	45359	No-app	45359	

B. Datasets Description

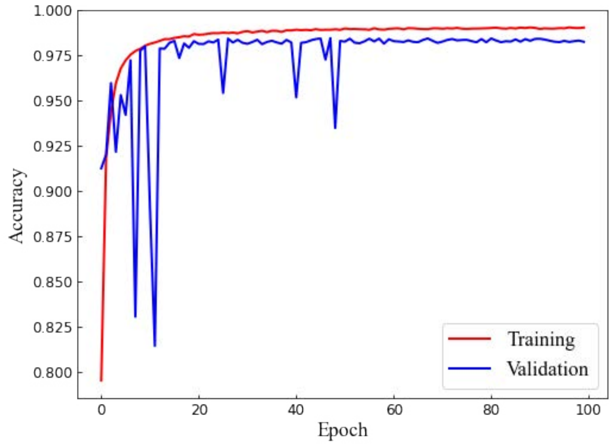
In this letter, we use a publicly available an open source dataset that contains 802.11 standard-compliant L1 waveforms [12]. The waveforms are generated by different 802.11 technologies (b, g, n), which results in different transmission schemes such as Direct-Sequence Spread Spectrum (DSSS) in 802.11b and Orthogonal Frequency Division Multiplexing (OFDM) in 802.11g/n, different types of L2 frames (management, control and data), and multiple MCS (modulations such as Binary Phase Shift Keying (BPSK) and Complementary Code Keying (CCK) for 802.11b and BPSK and Quadrature Phase Shift Keying (QPSK), 16-Quadrature Amplitude Modulation (QAM), 64-QAM for 802.11g/n with coding rates of 1/2, 3/4, and 5/6 according to the standard and modulation selected). Moreover, the payload carried by this L1 packets (information at L2 and above) were generated using real traces of L7 application running on a mobile device and connected to a secured 802.11 Access Point (AP) with Wi-Fi Protected Access (WPA)-2. Table I depicts the sample distribution for each classification task. The dataset contains a total of 140665 L1 packets.

C. Implementation Details

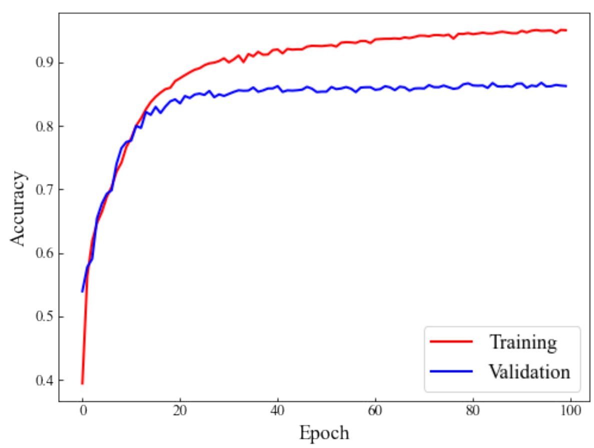
To evaluate the performance of the proposed classifier, we have splitted the whole datasets into training (80%), validation or testing (20%) sets. All the training and testing programs have been implemented in anaconda python 3.7 on a system equipped with 3.80 GHz CPU, 256 GB RAM, and a single NVIDIA Quadro RTX 6000 GPU. The proposed network learning rate is set to be .0001 and the mini-batch size is set as 64 during training. 100 iterations, i.e. epochs are used to optimize the proposed classifier. Moreover, "Adam" optimizer is used to perform optimization of the network. In addition, we have also scaled the data to the range of $[-1, 1]$ for training the network. In order to evaluate performance of our proposed classifier, we have taken four performance metrics, (e.g., accuracy, precision, recall, and F1 score) are measured. For a binary class problem, theses metrics of the classifier are defined as,

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

$$Precision = \frac{TP}{(TP + FP)}$$

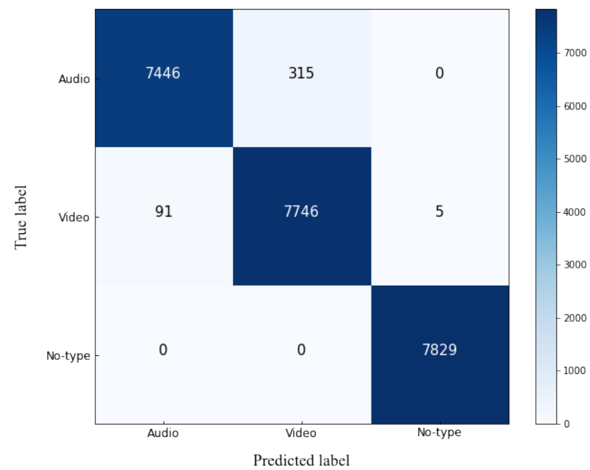


(a)

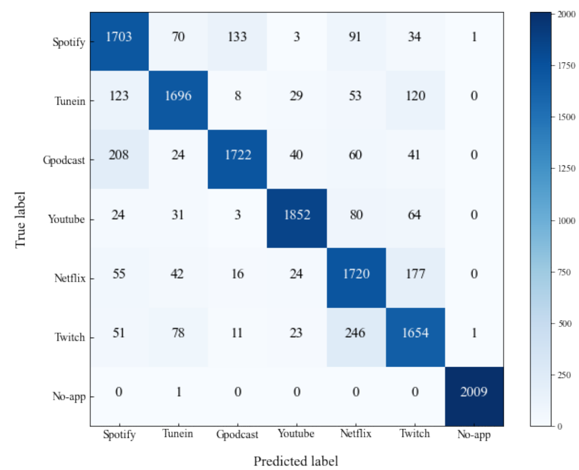


(b)

Fig. 2. The training and validation accuracy plot of proposed classifier for the TC task (a) application characterization (b) application identification.



(a)



(b)

Fig. 3. Confusion matrix of proposed classifier for the TC task (a) application characterization (b) application identification.

$$Recall = \frac{TP}{(TP + FN)}$$

$$F1 = 2 \times \frac{precision.recall}{precision + recall}$$

where, true positive, true negative, false positive, false negative are abbreviated as TP, TN, FP, FN, respectively. In the multi-class cases, several averaging techniques are used to extend these binary metrics to multi-class. In our experiment, binary classification metrics are measured employing one-vs-rest strategy for each class. Afterwards, we compute the weighted average of individual binary metric since class imbalance exists in our experiment.

D. Results AND Discussions

In this section, the overall performance of the proposed classifier are investigated for both of the task, (e.g., application characterization and application identification) in terms of accuracy, precision, recall, and F1 score. Fig. 2 depicts training and validation accuracy against the no. of epoch for both of the task. For the task application characterization, the training accuracy plot have increased with the increasing of epoch. After 100 iterations, the achieved accuracy is 98.25%. From validation accuracy plot, we can see that the classifier tends to suffer from vanishing gradient problem at the beginning of the training. But, the network significantly recover from that problem and get stable with higher number of epochs. On the other hand, the achieved accuracy is 86.28% for the task application identification. In this case, there is no sign of getting suffered from vanishing gradient problem during training.

Fig. 3 depicts the confusion matrix for both of the classification task. For the first type of classification task, i.e. application characterization, the no. of detected true positive class for “Audio”, “Video”, and “No-type” are 7829, 7746, and 7446, respectively, which indicates that the performance of classification audio and No-type app traffic recognition is good compared to the video traffic recognition. Besides, the value of weighted average precision, recall, and F1 score is 0.985, 0.984, and 0.986, respectively in this case which is very promising. For the second classification task, i.e. application identification, the highest detected true positive class is for “Youtube” class after the detection rate of “No-app” class and the lowest identification is found for the “Twitch” class. The weighted average precision, recall, and F1 score is 0.866, 0.863, and 0.864, respectively.

III. Conclusion

In this paper, we have proposed a hybrid deep neural network combining CNN with RNN in order to investigate the its performance on classifying traffic at L7 layer using L1 packets. The proposed network can extract both spatial and temporal features from IQ samples of the L1 packet data and can significantly increase the learning accuracy exploiting those extracted features. Besides, the proposed network is very efficient efficient in terms of time cost and computational memory. The detail of network architecture and training method for the proposed classifier have been presented. The simulation results have also been discussed to evaluate the performance. The overall results validate the proposed network is very promising in traffic classification at L7 using the L1 packet. Future work will focus on enhancing the accuracy significantly for the classification task of L7 layer as well as investigate the performance for the classification task of other layer as well.

Acknowledgment

This work was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2022-2018-0-01396) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

References

[1] Y. Zeng, H. Gu, W. Wei and Y. Guo, “$Deep-Full-Range$: A Deep Learning Based Network Encrypted Traffic Classification and Intrusion Detection Framework,” *IEEE Access*, vol. 7, pp. 45182-45190, 2019.

[2] M. Roughan, S. Sen, O. Spatscheck *et al.*,“Class-of-service mapping for qos: A statistical signature-based approach to ip traffic classification,” in *Proc. of the 4th ACM SIGCOMM Conference on Internet Measurement, ser.*IMC ’04. ACM, 2004, pp. 135–148.

[3] S. Kumar, S. Dharmapurikar, F. Yu, P. Crowley and J. Turner, “Algorithms to accelerate multiple regular expressions matching for deep packet inspection,” *ACM SIGCOMM Comput. Commun. Rev.,* vol. 36, pp. 339-350, 2006. vol. 65, no. 15, pp. 4075–4089, Aug. 2017.

[4] Z. Chen, K. He, J. Li *et al.*, ““Seq2img: A sequence-to-image based approach towards ip traffic classification using convolutional neural networks,” in *Proc. 2017 IEEE International Conference on Big Data (Big Data),* pp. 1271–1276, 2017.

[5] W. Wang, M. Zhu, J. Wang *et al.*, “End-to-end encrypted traffic classification with one-dimensional convolution neural networks,” in *Proc. 2017 IEEE International Conference on Intelligence and Security Informatics (ISI),* pp. 43–48, 2017.

[6] S. Rezaei and X. Liu, “How to achieve high classification accuracy with just a few labels: A semi-supervised approach using sampled packets,” *arXiv preprint arXiv:1812.09761,* 2018.

[7] X. Wang, S. Chen, and J. Su, “Real network traffic collection and deep learning for mobile app identification,” *Wireless Communications and Mobile Computing,* vol. 2020, 2020.

[8] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, *et. al.* “Network traffic classifier with convolutional and recurrent neural networks for internet of things,” in *IEEE Access,* vol. 5, pp. 18 042–18 050, 2017.

[9] D. Li, Y. Zhu, and W. Lin, “Traffic identification of mobile apps based on variational autoencoder network,” in *Proc. 2017 13th International Conference on Computational Intelligence and Security (CIS),* pp. 287–291, 2017,.

[10] T. J. O’Shea, S. Hitefield, and J. Corgan, “End-to-end radio traffic sequence recognition with recurrent neural networks,” in *Proc. 2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP),* pp. 277–281, 2016.

[11] M. Camelo, T. De Schepper, P. Soto *et al.* “Detection of traffic patterns in the radio spectrum for cognitive wireless network management,” in *Proc. 2020 IEEE International Conference on Communications (ICC),* jan. 2020.

[12] M. Camelo, P. Soto and S. Latré, “A General Approach for Traffic Classification in Wireless Networks using Deep Learning,” *IEEE Transactions on Network and Service Management,* November. 2021.

[13] R. Utrilla, E. Fonseca, A. Araujo, and L. A. Dasilva, “Gated recurrent unit neural networks for automatic modulation classification with resource-constrained end-devices,” *IEEE Access,* vol. 8, pp. 112783–112794, 2020.