Real-time Traffic Classification in Encrypted Wireless Communication Network

Yongming Chen*, Yuzhou Tong*, Gwee Bah Hwee*, Qi Cao†, Sirajudeen Gulam Razul‡, Zhiping Lin*
*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
†School of Computing Science, University of Glasgow Singapore, Singapore
‡Temasek Laboratories, Nanyang Technological University, Singapore
{yongming001, ytong004}@e.ntu.edu.sg, {ebhgwee, esirajudeen, ezplin}@ntu.edu.sg, qi.cao@glasgow.ac.uk

***Abstract*—Classification of traffic service types is a valuable function for wireless communication networks. Even though some progress has been made, the recognition of the type of the traffic services cannot be done in real time. In this paper, we propose a novel method for classifying traffic series in real time based on transfer learning techniques. We pre-train a deep learning model with long traffic series and fine-tune the model with short traffic series. In this way, the developed model achieves the capability of recognising traffic services in real time. In other words, the model can recognize traffic services by using short traffic series. We collect Downlink Control Information (DCI) from commercial LTE networks when using five common types of traffic services. Then we use the dataset to validate our method. Our experimental results show that, by using proposed method, LSTM accuracy rates will increase to 80% and 88.5% when the length of the traffic series is 5 seconds and 10 seconds respectively, which is higher than the baseline. The strategy is also suitable for one dimension convolution neural network (1D-CNN).**

***Index Terms*—LTE, Downlink Control Information, Traffic classification, Transfer learning**

I. Introduction

The rapid development of wireless communication networks makes mobile phones indispensable in our daily life. A variety of applications (apps) have been developed to meet service requirements of users. Thus, service recognition becomes possible by analyzing the app data traffic in the wireless network. Service recognition mainly has two aspects of benefits to the wireless communication network. First, service recognition gives a better understanding of the traffic demand in the network. This is beneficial to network operations and management [1]. Second, service recognition reveals the possibility of leaking private information through the public commercial network, which regards as a security problem that needs to be solved.

A large amount of work on mobile encrypted traffic classification leverages UDP/TCP port analysis or packet-level information [2]–[7]. However, in LTE/4G and 5G networks, the information in the network layer or transport layer cannot be acquired in a passive way because of the network encryption mechanism. This makes traffic classification in LTE and 5G networks a challenging task.

Recently, some studies take advantage of the LTE downlink decoding software (e.g. SRS Airscope [8] or OWL [9]) to analyze Downlink Control Information (DCI). DCI carries the important scheduling information, and it is sent to the user equipment (UE) from the base station through the Physical Downlink Control Channel (PDCCH). As DCI contains adequate user-specific traffic information, it enables traffic classification by a passive sniffer. In [10], a recurrent neural network (RNN) based model was reported to recognize traffic service types by decoding DCI. Trinh *et al.* introduced the method to recognize not only traffic services, but also the usage apps in both supervised and unsupervised ways based on machine learning [11]. It can achieve about 90% accuracy in the identification task of three types of traffic services with the traffic series longer than 20 seconds. Nevertheless, the accuracy rate will drop rapidly, if the traffic series is shorter than 20 seconds. This will emerge a problem when the users switch between different traffic services with a short usage duration. The reported model will fail to recognize the traffic service. Another interesting work is presented in [12], with the usage of the data link layer information. The machine learning based classifier can recognize 20 types of apps in both iOS and Android mobile phones. However, this work was done only in the lab simulation network and the traffic series is longer than 50 seconds.

In this paper, we present a novel method to real-time recognize the traffic services in the real world commercial LTE network based on transfer learning. We leverage the knowledge from a deep learning model pre-trained with the long traffic series to improve the model performance on the short traffic series. This allows the recognition of LTE traffic services to become real-time. A developed passive sniffer first decodes the DCI by using Airscope. After preprocessing and encoding the raw data, we then train the model with long traffic series and fine-tune with short traffic series. We demonstrate the effectiveness of the proposed method through real-world LTE data. Furthermore, we combine our method with user identification tracking method introduced in [13] to make our traffic monitor be able to long-term track for a specific user.

We summarize our main contributions as follows:

- We propose a novel method for effectively classifying short time series based on transfer learning techniques.
- A raw LTE traffic information encoding method is presented to reduce the complexity of the model training.
- We demonstrate the ability of the proposed method to learn through real-world LTE data to recognise five types of the traffic services in real time, including streaming

video, streaming music, social media, text chat and video calls.

The remainder of the paper is organized as follows. Section II introduces the LTE technique background of our proposed traffic services classifier. Section III presents our data acquisition system, the proposed raw traffic data encoder and our deep learning model for short traffic series classification. Section IV demonstrates the experimental setting and results and comparison results to other methods. Section V concludes the paper.

II. Background of LTE

We briefly introduce the LTE network infrastructure, protocol stack and identifiers which serve as background of our proposed traffic services classifier.

A. LTE Network Infrastructure

The LTE Network infrastructure is shown in Fig. 1. The LTE network contains multiple user equipment (UE), including mobile phones and other user mobile devices. The base stations, Evolved NodeB (eNodeB) is the intermediate connectors. Because the messages from the eNodeB to the UEs are broadcast in the air, this gives us an opportunity to get the unencrypted information. The core network, Evolved Packet Core (EPC), is in charge of establishing the point-to-point connection from the UEs to the Internet. The traffic monitor presented in this paper consists of the software defined radios (SDR) and a computer is deployed between the UE and the eNodeB.

B. LTE Protocol Stack

The LTE protocol stack contains the physical layer (L1), data link layer (L2) and network layer (L3). The functions of the L3 layer include system message broadcasting, paging, wireless connection management, etc. The L2 layer is responsible for data encryption, data segmentation and concatenation, etc. L1 layer completes physical layer processes including coding, modulation, and multi-antenna mapping. The PDCCH, an important physical channel in the physical layer, contains DCI. DCI carries resource allocation information, which is useful to identify the app or traffic service usage on the victim's device [10], [11].

C. LTE Identifiers

There are several identifiers used in LTE network. International Mobile Subscriber Identity (IMSI) is a unique ID that globally identifies a SIM card. Because of IMSI's high sensitivity, when the mobile subscriber first accesses the network, the network allocates an Temporary Mobile Subscriber Identifier (TMSI) to the subscriber. Meanwhile, eNodeB transmits the TMSI without encryption when setting up the Radio Resource Control (RRC) connection [13]. This makes getting TMSI possible. Even though TMSI is refreshed periodically, its longer life time than Cell Radio Network Temporary Identifier (C-RNTI) gives us the opportunity to track a specific user for a long time.

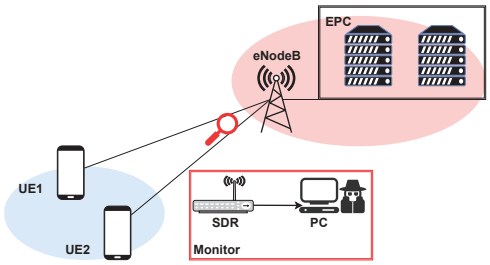

Fig. 1. LTE Network Infrastructure and the position of the deployed traffic monitor

TABLE I
Names and categories of the apps for experiment

Traffic service type	Representative app
Streaming Video	TikTok
Streaming Music	Spotify
Social Media	Instagram
Test Chat	Whatsapp
Video Calls	Whatsapp

III. Proposed Methodology

A. Data acquisition System

For data collection, we set our experiment mobile phones to connect to the commercial LTE network. We then launch the downlink traffic monitor to collect the broadcast information from the base station connected by user devices. For this, we use the SDR, USRP X310, equipped with the downlink sniffer software, SRS AirScope. AirScope enables us to get all DCI messages and MAC layer packets, including the RRC connection setup message. The traffic monitor position is between the eNodeB and UEs as shown in Fig. 1.

We first upload a large size file to make a burst in uplink data rate. In this way, we can determine the initial RNTI of the experimented mobile phone. Then we control the mobile phone to use the specific apps. We select five common traffic service types in mobile networks and set the mobile device to use the representative apps in each type. The list of the selected apps is shown in TABLE I. We collect 100 traffic traces for each app and each trace lasts for 50 seconds. After that we map multiple RNTIs with the TMSI. The corresponding DCI messages are filtered by using the RNTIs chain to get all DCI messages of the experimented mobile phone. In this paper, we use uplink and downlink Transport Block Size (*TBS*) in DCI messages as the raw traffic data.

B. Encoder

In order to decrease the complexity of the machine learning model, we design an encoder to compress the raw traffic series to the same length. A sliding window length of 0.1 second is used to move rightwards from the beginning of the traffic series to the end without overlapping. The spilt traffic series is then used to calculate eight dimensions features, which are *mean TBS in uplink*, *mean TBS in downlink*, *max TBS in uplink*, *max TBS in downlink*, *total TBS in uplink*, *total TBS in*

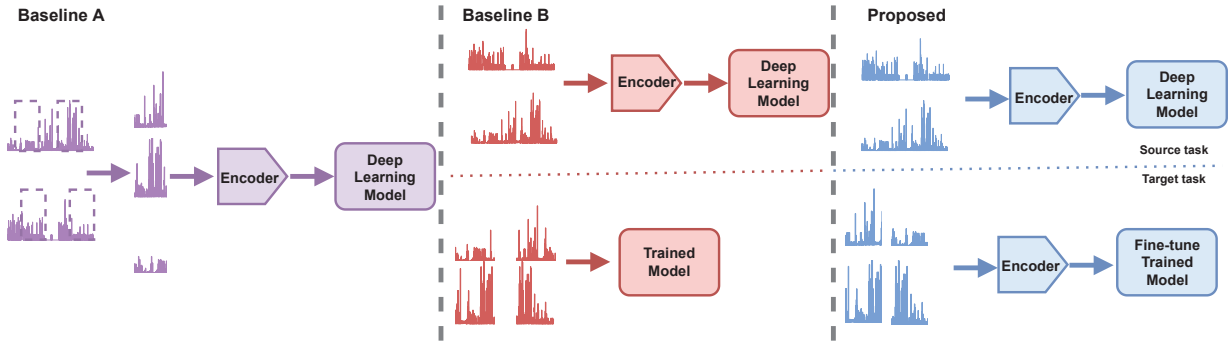

Fig. 2. Experiment and proposed model architecture. Baseline A: Traditional training deep learning model with using short traffic series randomly sliced from long traffic series. Baseline B: Training the model with long traffic series and testing with short traffic series. Proposed: Pre-training the model with long traffic series and fine-tuning pre-trained model with short traffic series

downlink, *total numbers of packets arrived in uplink* and *total numbers of packets arrived in downlink*. Next, the encoded data will be inputted to the deep learning model.

C. Model

In this section, we introduce our training method to make the deep learning model capable of recognising short traffic series based on transfer learning techniques.

Pan *et al.* introduced an elegant definition of transfer learning in [14]. Consider a domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$, where $\mathcal{X}$ is a feature space, $P(X)$ means the marginal probability distribution and $X = \{x_1, x_2, \ldots, x_n\} \in \mathcal{X}$. The task can be defined as $\mathcal{T} = \{\mathcal{Y}, f(.)\}$, where $\mathcal{Y} = \{y_1, y_2, \ldots, y_m\}$ is the label space and $f(.)$ is the objective predictive function $f : \mathcal{X} \rightarrow \mathcal{Y}$. Given a source domain $\mathcal{D}_S$ and source task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and target task $\mathcal{T}_T$, transfer learning is designed to help learn the $f_T(.)$ with the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$.

In this paper, $\mathcal{D}_S$ and $\mathcal{D}_T$ are traffic series in different time durations. We only collect data from LTE networks, so $\mathcal{X}_S = \mathcal{X}_T$. But different traffic series lengths will make $P_S(X) \neq P_T(X)$. Thus, we leverage the knowledge in the model pre-trained by long traffic series to enable training the models for short traffic series. We propose to use Long-Short Term Memory (LSTM) networks [15] with fully connected layer to this task. Thus, the objective predictive function will be:

$$f_S(.) = f_S^{FC}(f_S^{LSTM}(.)) \tag{1}$$

where $f_S^{FC}(.)$ and $f_S^{LSTM}(.)$ are the fully connected layer and LSTM layers trained by long traffic series, respectively. For an input sequence, $S = \{s_1, s_2, s_3, ...s_T\}$, the forward process of an LSTM cell at time t is shown in Eq. (2) - Eq. (7).

$$i_t = \sigma(W_{is}s_t + W_{ih}h_{t-1} + b_i) \tag{2}$$

$$f_t = \sigma(W_{fs}s_t + W_{fh}h_{t-1} + b_f) \tag{3}$$

$$o_t = \sigma(W_{os}s_t + W_{oh}h_{t-1} + b_o) \tag{4}$$

$$\tilde{c}_t = tanh(W_{cs}s_t + W_{ch}h_{t-1} + b_c) \tag{5}$$

$$m_t = f_t \odot m_{t-1} + i_t \odot \tilde{c}_t \tag{6}$$

$$h_t = o_t \odot tanh(m_t) \tag{7}$$

where i_t, f_t and o_t are the input gate, forget gate and output gate respectively. $\tilde{c}_t$ is the intermediate state; m_t is a memory cell and h_t is the hidden state. $tanh(.)$ and $\sigma(.)$ are activation functions and $\odot$ is pointwise multiplication. The procedure of the model training is as below.

- We first train the LSTM with fully connected layer with the encoded long traffic series, whose length is with 50 seconds. From this we get the $f_S(.) = f_S^{FC}(f_S^{LSTM}(.))$ on $\mathcal{D}_S$.
- We transfer the pre-trained model in the previous step to the short traffic series. When training the transferred model, we set the learning rate of the LSTM layer to an extremely small value. In this way, we could freeze the LSTM layers and maintain its knowledge learning from the long traffic series.
- The pre-trained fully connected layer is discarded, and a new fully connected layer will be trained to fit the short traffic series. Finally, we will get $\mathcal{T}_T$ on $\mathcal{D}_T$. The objective predictive function $f_T(.)$ in $\mathcal{T}_T$ will be:

$$f_T(.) = f_T^{FC}(f_S^{LSTM'}(.)) \tag{8}$$

where $f_T^{FC}(.)$ means the fully connected layer trained by short traffic series and $f_S^{LSTM'}(.)$ means the LSTM layers from $\mathcal{T}_S$ with minor changes.

Then the softmax function is used to get the probability of each class i shown in Eq. (9).

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{9}$$

where $\vec{z} = \{z_1, z_2, \ldots, z_K\}$ is the output of the full connected layer and K is the number of the classes. Moreover, we demonstrate this training method can also be suitable for one dimension convolution neural network (1D-CNN). The detailed presentation is omitted here due to page limitation.

IV. Experiments and Results

A. Experiments

In order to validate our proposed approach, we compare the performance of the proposed method with two baseline

methods. The procedures of the experiment is shown in Fig. 2.

- Baseline A: It is the most traditional method to train the model. We use a fixed length time window randomly slicing the long traffic series to generate short series. Then we directly input the short series to the encoder and deep learning model for training and testing. This method is used in [11]. Baseline A is to validate the necessity of learning different objective functions $f(.)$ in $\mathcal{D}_S$ and $\mathcal{D}_T$.
- Baseline B: It is to train the model using the long traffic series but to test the model performance using short time series. This is to validate whether the model trained by long traffic series has the capacity to classify short traffic series, which means whether $P_S(X)$ is equal to $P_T(X)$.
- Proposed: It is our proposed transfer learning method.

All models use the same dataset to train and test, and the proportion of the test data is 20%. The accuracy rates on the test data are reported as the metrics of the model's performance. For the model trained by each method, we test the performance on seven different traffic series lengths, including 2 seconds, 5 seconds, 10 seconds, 20 seconds, 30 seconds, 40 seconds, and 50 seconds (i.e., the whole traffic series).

B. Results

The experiment results are shown in Fig. 3 and Fig. 4. It can be seen that the accuracy rates of the whole traffic series are high, which is 92% in LSTM and 96% in 1D-CNN. This means that the traffic service can be effectively recognized by using a deep learning model when traffic series are long enough. However, the accuracy rates drop rapidly in both deep learning models trained by methods A and B. In method B, the faster decreasing accuracy rates mean that the model trained by long traffic series does not have the ability to recognize short traffic series. For 1D-CNN, the accuracy rate even drops to 20%, nearly the chance level for five class classification tasks. This is because the size of the fully connected layer in 1D-CNN is fixed. But the length of the traffic series becomes shorter when testing the model using short traffic series. In order to use pre-trained 1D-CNN, the short traffic series need to be padded with zeros, which introduces noise to data and deteriorates the model performance.

For the proposed method, it can be seen that the performance is better than the other two methods. This illustrates that fine-tuning the pre-trained model by short traffic series gives better performance. The knowledge from the pre-trained model is able to guide to learn how to recognize short traffic series, which is lacking in both Method A and B. The accuracy rates can reach 81% and 88.5% in LSTM by using traffic series length at 5 and 10 seconds. This improves the real-time performance of the traffic classifier. In addition, fine-tuning the 1D-CNN achieves better performance because we only keep the CNN layers of the pre-trained model and the size of the fully connected layer is flexible for various short traffic series. Meanwhile, the performance of the fine-tuned model is more stable than the other two methods. Especially in 1D-CNN, the accuracy rate can maintain at 80% when traffic series length is 10 seconds.

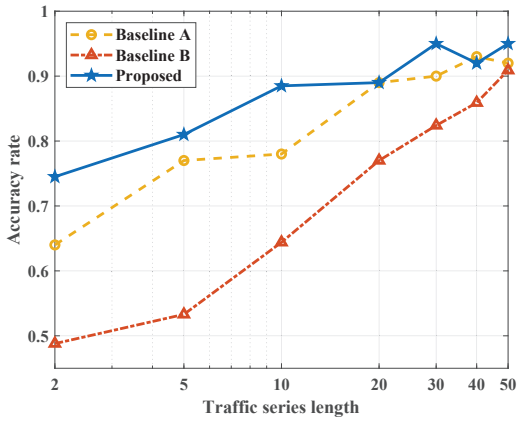

Fig. 3. Accuracy rates *vs* traffic series length on LSTM

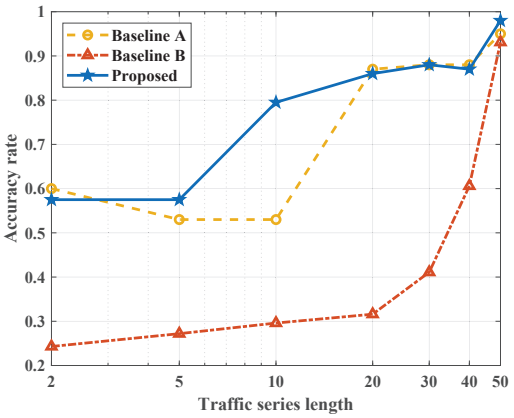

Fig. 4. Accuracy rates *vs* traffic series length on 1D-CNN

V. Conclusion

In this paper, we present a transfer learning method to deal with real-time traffic service classification in LTE. From the results, it is observed that the proposed method can achieve a high accuracy rate even though the traffic series is extremely short. Comparing with the two baseline methods, we can validate the effectiveness of our proposed strategy. The accuracy rates of LSTM are 80% and 88.5% when traffic series length is 5 and 10 seconds respectively. Furthermore, when compared to traditional methods, the performance of 1D-CNN can be improved, and the performance is more stable for shorter traffic series.

For the equivalent capabilities of the 5G wireless communication networks, as the DCI format and PDCCH are similar in LTE and 5G, the model proposed in this paper could be extended to 5G networks.

References

[1] W. Wu, L. Jiang, C. He, D. He, and J. Zhang, "RavenFlow: Congestion-Aware Load Balancing in 5G Base Station Network," in *2020 IEEE*

International Symposium on Circuits and Systems (ISCAS). IEEE, 2020, pp. 1–5.

[2] H. F. Alan and J. Kaur, "Can Android applications be identified using only TCP/IP headers of their launch time traffic?" in *Proceedings of the 9th ACM conference on security & privacy in wireless and mobile networks*, 2016, pp. 61–66.

[3] Q. Wang, A. Yahyavi, B. Kemme, and W. He, "I know what you did on your smartphone: Inferring app usage over encrypted data traffic," in *2015 IEEE conference on communications and network security (CNS)*. IEEE, 2015, pp. 433–441.

[4] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde, "Can't you hear me knocking: Identification of user actions on android apps via traffic analysis," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, 2015, pp. 297–304.

[5] B. Saltaformaggio, H. Choi, K. Johnson, Y. Kwon, Q. Zhang, X. Zhang, D. Xu, and J. Qian, "Eavesdropping on Fine-Grained User Activities Within Smartphone Apps Over Encrypted Network Traffic," in *10th USENIX Workshop on Offensive Technologies (WOOT 16)*, 2016.

[6] T. van Ede, R. Bortolameotti, A. Continella, J. Ren, D. J. Dubois, M. Lindorfer, D. Choffnes, M. van Steen, and A. Peter, "Flowprint: Semi-supervised mobile-app fingerprinting on encrypted network traffic," in *Network and Distributed System Security Symposium (NDSS)*, vol. 27, 2020.

[7] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, "Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2016, pp. 439–454.

[8] Software Radio Systems, "SRS AirScope." [Online]. Available: https://www.srs.io/products/ [Accessed on Oct. 30, 2022].

[9] N. Bui and J. Widmer, "OWL: A reliable online watcher for LTE control channel measurements," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, 2016, pp. 25–30.

[10] J.-W. Son, S. Lee, and M.-h. Han, "Supervised Service Classification using Downlink Control Indicator in LTE Physical Downlink Control Channel," in *2021 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2021, pp. 1533–1536.

[11] H. D. Trinh, A. F. Gambin, L. Giupponi, M. Rossi, and P. Dini, "Mobile traffic classification through physical control channel fingerprinting: a deep learning approach," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1946–1961, 2020.

[12] L. Zhai, Z. Qiao, Z. Wang, and D. Wei, "Identify What You are Doing: Smartphone Apps Fingerprinting on Cellular Network Traffic," in *2021 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2021, pp. 1–7.

[13] D. Rupprecht, K. Kohls, T. Holz, and C. Pöpper, "Breaking lte on layer two," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 1121–1136.

[14] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[15] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.