UTMobileNetTraffic2021: A Labeled Public Network Traffic Dataset

Yuqiang Heng, Vikram Chandrasekhar, and Jeffrey G. Andrews, *Fellow, IEEE*

***Abstract*—A high-quality network traffic dataset is essential to the development of accurate network traffic classification algorithms. In this work, we present a new labeled public network traffic dataset with realistic mobile traffic from a wide range of popular applications. An automated platform is constructed to generate and collect data traffic from specified applications in a controlled environment. The dataset contains over 21 million packets from more than 29 hours of mobile traffic with application and activity-level labels. We provide an application classification example using machine learning (ML) models trained on the proposed dataset.**

***Index Terms*—Encrypted traffic classification, network traffic dataset, machine learning.**

I. Introduction

NETWORK traffic classification provides the foundation for a wide range of services, such as dynamic scheduling, quality of service (QoS) assurance, content-based billing and abnormality detection. While network traffic classification has been studied for over two decades, the explosion of mobile data traffic along with increasing packet encryption, have shifted the paradigm: traditional approaches that rely on deep packet inspection (DPI) or examining the port of packets have become less viable.

Machine learning (ML) and deep learning (DL) models extract meaningful patterns from the data and learn a mapping from the input features to the class labels. State-of-the-art ML and DL approaches can achieve high classification accuracy even with encrypted data traffic. Although ML and DL traffic classification models often use different input features, they are both data-driven and heavily rely on high-quality training data. While numerous recent works have proposed ML and DL based network traffic classifiers, there has been a lack of public labeled datasets. In this work, we present a public network traffic dataset named UTMobileNetTraffic2021 with application and activity level labels. With the assistance of 6 undergraduate UT Austin students who undertook this effort as a part of their senior design project, an automated platform was developed to generate and collect data traffic from a wide range of popular mobile applications in a controlled environment. The dataset consists of over 29 hours of data traffic generated by both the automated platform and human users. We also provide examples of application and activity classification using the proposed dataset, where we train ML models using flow-based features extracted from packet headers. The dataset and the classification example are published online.[1]

The rest of this letter is organized as follows. The related work and existing datasets are discussed in Section II. The experimental setup is explained in Section III. The data collection procedure and characteristics of the dataset are explained in Section IV. Example uses cases of the dataset are provided in Section V. Finally, the conclusion and final remarks are provided in Section VI.

II. Related Work

Network traffic classification is essential to various traffic engineering tasks. Early approaches relying on DPI and packet port numbers have become less favorable since modern network traffic is often encrypted. State-of-the-art approaches extract features from the packets and use ML models to classify the traffic. We refer to [1] and [2] for surveys on traffic classification techniques.

High-quality datasets are essential to ML-based traffic classification. However, sharing user data can cause privacy concerns, and labeling network traffic data is difficult and can be expensive. As a result, public labeled network traffic datasets are rare. In our previous work [3] where we proposed ML-based traffic classification models, a major challenge was the lack of modern high-quality data. Some recent works such as [4] use proprietary datasets, making it difficult for other researchers to reproduce their results. On the other hand, the datasets used in [5], [6] and [7] only contain service-level labels, which limit the granularity of the classification algorithm. As summarized in Table I, there are only a few public labeled datasets for application classification.

- **ISCXVPN2016:** The ISCXVPN2016 dataset contains labeled traffic data generated by 20 applications across 7 different service categories [8]. It contains both normal traffic and traffic over VPN. The dataset includes categories such as browsing, email, chat, streaming, file transfer, VoIP and P2P, and popular applications such as YouTube and Facebook. An updated ISCXTor2016 dataset is also published for Tor-encrypted traffic [9].
- **USTC-TFC2016:** The USTC-TFC2016 dataset contains both malware traffic data and benign traffic data [10]. The malware dataset contains various types of malware traffic collected from real network environments. The benign

Manuscript received May 14, 2021; revised June 25, 2021; accepted July 10, 2021. Date of publication July 19, 2021; date of current version August 21, 2021. The associate editor coordinating the review of this article and approving it for publication was H. H. Gharakheili. *(Corresponding author: Yuqiang Heng.)*

Yuqiang Heng and Jeffrey G. Andrews are with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA (e-mail: yuqiang.heng@utexas.edu).

Vikram Chandrasekhar was with Standard and Mobility Innovation Lab, Samsung Research America, Mountain View, CA 94043 USA. He is now with Amazon, Mountain View, CA 94041 USA.

Digital Object Identifier 10.1109/LNET.2021.3098455

[1]The dataset and code are available: https://github.com/YuqiangHeng/UTMobileNetTraffic2021.

traffic dataset contains traffic from 10 different applications collected through a network simulator but lacks some of the modern popular applications. The size of the benign dataset is also limited.

- **IMTD17:** The IMTD17 dataset contains mobile traffic generated by 12 Android applications [11]. The dataset was collected on a campus gateway. The application-specific traffic are generated by smartphones connected to WiFi. The dataset contains transformed features but not raw packet data from 1000 labeled traffic flows. The dataset includes popular applications in China but lacks applications widely used worldwide.
- **Unicauca18:** The Unicauc dataset contains traffic generated by 75 applications captured in a network section in a university environment [12]. It contains statistical features extracted from IP flows. The application labels are obtained using DPI.
- **MTD18:** The MTD18 dataset contains raw traffic data generated by 12 Android applications [13]. During data collection, a custom client software on a mobile device routes traffic to a VPN server and reports the socket information associated with each application. The server than obtains the ground truth labels by checking the socket information of collected traffic data. Similar to the USTC-TFC2016 and the IMTD17 datasets, this dataset mostly considers popular applications in China.
- **MIRAGE-2019:** The MIRAGE-2019 dataset contains mobile traffic data from 40 Android applications generated by human users [14]. The dataset includes per-packet features from the first 32 packets of each bidirectional flow, per-flow statistical features and the metadata of each flow. The labels are generated using the socket information or according to a heuristic: assigning the most common label.

Numerous public datasets have also been published for purposes other than application classification, including a dataset for YouTube traffic analysis [15], a dataset for Netflix traffic analysis and user experience inference [16] and a dataset for Internet of Things (IoT) device classification [17].

The advantages of the proposed UTMobileNetTraffic2021 dataset over these existing ones include:

- **A wide range of modern applications:** Compared to the datasets proposed in [10], [11] and [13] which only contains popular applications in China, the proposed dataset includes more widely used modern applications, ranging from social networking apps like Instagram, media consumption apps like YouTube and utility apps like Dropbox.
- **Accurate activity-level labels:** An automated traffic generation and data collection platform was built. Instead of relying on DPI or socket information to obtain the labels, users can specify the activities to perform on each application, allowing the data to be labeled with both the application and the activity being performed. The proposed dataset provides accurate labels at a finer granularity compared to the existing datasets.
- **Flexible feature engineering:** The proposed dataset captures TCP/IP packet headers and their timestamps in both the downlink and the uplink direction. Compared to the datasets in [11], [12] and [14] which provide either extracted features or processed data, the proposed dataset allows more flexible data analysis and feature engineering.

III. Experimental Setup

In order to construct an automated data collection platform, a Linux laptop running Ubuntu is used as the central controller for application activity generation and packet capturing. Data traffic is generated using 3 unlocked Samsung Galaxy S5 smartphones. The smartphones are manually rooted and the LineageOS Android distribution is installed so that the laptop can gain full access and control. The smartphones are connected to the Internet through a WiFi access point (AP) and are connected to the laptop through USB 2.0 wired connections. The laptop sends data collection commands such as launching applications on the smartphone and retrieving collected data through the USB connection. Each experiment was conducted at one of three locations. Location 1 is the Engineering Education and Research Center on the UT Austin campus. It is equipped with the Cisco Aironet 2800 series Wireless Access Points (WAP) connected to the wired network with 1 Gigabit Ethernet connections. The WAPs support the IEEE 802.11ac standard and use the 5 GHz band. Location 2 is the Prather Residence Hall (PRH), a student dormitory on the UT Austin campus. It is equipped with the Cisco Aironet 2702i series WAPs with 1 Gigabit Ethernet connections to the wired network. The WAPs support the IEEE 802.11a/g/n/ac standards and use the 2.4 GHz and the 5 GHz band. Location 3 is a house in a residential area in Houston. It uses the 50 megabits per second (Mbps) wired Internet plan from AT&T and is equipped with an AT&T WiFi gateway supporting the IEEE 802.11b/g/n/ac standards. The overall experimental setup is illustrated in Fig. 1.

The general workflow of the automated data collection platform is shown in Fig. 2. From the laptop controller, the user specifies the application and the activity to perform on the application. The laptop then gains root access of the smartphone through Android debug bridge (ADB), which is an API for third parties to access and control Android devices. The laptop launches TCPDUMP on the smartphone to start collecting data packets and then launches the specified application. The smartphone executes a series of actions on the application, which generates the corresponding network traffic. After all specified actions have been completed, TCPDUMP and the application are closed, and the collected data packets are transferred and stored on the laptop.

IV. Data Collection

We select 16 of the most popular mobile applications listed on [18], so that the collected data is representative of the modern data consumption patterns. Typical user activities are selected for each application, such as browsing and posting for Reddit and sending and opening emails for Gmail. For each activity, a series of actions are implemented through the Android API to emulate a sequence of user interactions with the smartphone. Examples of actions include scrolling

TABLE I
SUMMARY OF EXISTING PUBLIC DATASETS FOR APPLICATION CLASSIFICATION

Dataset	Data Granularity	Label Granularity	Features	number of applications	labeled dataset size	Year
ISCX [8]	transport layer	application	raw data	20	12 million packets	2016
USTC-TFC2016 [10]	transport layer	application	raw data	10	2 million packets	2016
IMTD17 [11]	application layer	application	flow features	12	1,000 HTTP sessions	2017
Unicauca18 [12]	transport layer	application	flow features	75	3.5 million flows	2017
MTD18 [13]	transport layer	application	raw data	12	20.5 million packets	2018
MIRAGE-2019 [14]	transport layer	application	packet & flow features	40	27.8 million packets	2019
UTMobileNetTraffic2021	transport layer	application & activity	packet header information	16	21 million packets	2019

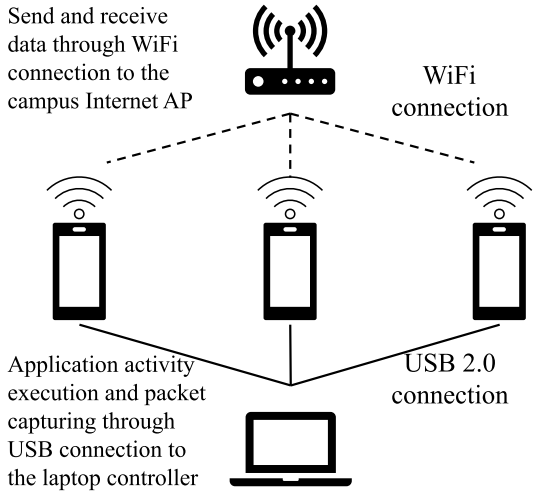

Fig. 1. Data Collection Setup.

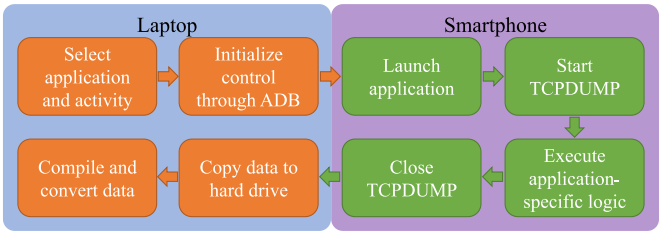

Fig. 2. Data Collection Workflow.

the smartphone screen, clicking on part of the displayed content to go to a different screen, and waiting for the displayed media to play for a certain time.

When collecting data, a BASH script launches the application on the smartphone through ADB, waits for the application to finish loading, then starts recording network packets through TCPDUMP on the smartphone. Background tasks are cleared before starting data collection for a new application, and each application is closed after data collection is complete. This minimizes the amount of background traffic in the collected data so that the majority of the traffic is generated by the target application. After packet recording is complete, the raw pcap data files are transferred to the laptop, where they are processed using TShark and converted to the csv format. Modern mobile traffic is often encrypted, which makes DPI infeasible and traffic classification more challenging in general. While

TABLE II
SUMMARY OF RECORDED DATA FEATURES

Header	Stored features
Frame	frame.number, frame.time, frame.len, frame.cap_len
Link	sll.pkttype, sll.hatype, sll.src.eth, sll.unused, sll.etype
IP	ip.hdr_len, ip.dsfield.ecn, ip.len, ip.id, ip.frag_offset, ip.ttl, ip.proto, ip.checksum, ip.src, ip.dst
TCP	tcp.hdr_len, tcp.len, tcp.srcport, tcp.dstport, tcp.seq, tcp.ack, tcp.flags.ns, tcp.flags.fin, tcp.window_size_value, tcp.check_sum, tcp.urgent_pointer, tcp.option_kind, tcp.option_len, tcp.options.timestamp.tsval, tcp.options.timestamp.tsecr
UDP	udp.scrport, udp.dstport, udp.length, udp.checksum, gquic.puflags.rsv, gquic.packet_number

packet payloads may still contain useful information such as the Transport Layer Security (TLS) handshake fields despite of the encryption, they are removed for ease of data storage and for privacy concerns. No user-specific information or meta data is included in the dataset for privacy reasons. The dataset includes features extracted from packet headers shown in Table II for each individual packet. During the data collection stage, the packets are recorded as a raw time series instead of being grouped into flows, which are defined by sequences of packets sharing the same TCP/IP 5 tuple, i.e., the source and destination IP addresses, the source and destination ports and the protocol. One activity on one application is performed during each data collection experiment. All packets captured in the same experiment are labeled with the corresponding application and activity.

Three different sets of data were collected. In the deterministic automated dataset, the action parameters for each activity are fixed. For instance, when performing the activity of scrolling news feed on Facebook, the BASH script will always scroll the feed 3 times, wait for 5 seconds, and repeat for 5 times. Although the actions are fixed, the context and content displayed are up to the application. In the randomized automated dataset, the action parameters for each activity are randomized, such as the number of scrolls, the wait time and the number of repetitions for Facebook news feed scrolling.

TABLE III
SUMMARY OF DATA DISTRIBUTION

App	Activities	Duration (hours)	Packets	Flows	Bytes (GB)
Facebook	scroll news feed, search page	2.12	973.83k	8.81k	1.44
Twitter	scroll feed, post tweet	1.72	233.11k	11.02k	0.14
Reddit	browse, post	1.60	1803.77k	23.97k	8.31
Instagram	search browse, send message	2.56	662.37k	7.99k	1.16
Pinterest	tap board	1.53	945.46k	9.06k	1.14
YouTube	search, play video	1.05	269.53k	14.73k	0.48
Netflix	browse home, watch video	2.49	3998.80k	10.73k	5.10
Hulu	scroll home, watch video	1.71	628.38k	10.68k	0.58
Spotify	play music, search music	2.37	452.31k	13.24k	0.32
Pandora	play music, search music	0.31	196.38k	4.15k	0.15
Google Maps	browse, directions, download map, explore	4.97	10013.61k	19.34k	12.65
Google Drive	upload, download	1.75	1082.56k	5.97k	1.70
Dropbox	upload, download	1.66	311.64k	10.68k	0.78
Gmail	send email, open email	1.10	60.86k	4.40k	0.25
Messenger	chat	1.65	58.89k	4.23k	0.53
Hangout	chat	1.01	106.33k	5.99k	0.04

This makes the collected data more diverse and realistic. The third dataset is generated by human users and is the most realistic in terms of representing user activity. It includes two subsets: an application-specific dataset and an activity-specific dataset. In the former, human users perform each activity using applications in Table III. In the latter, human users use each application normally without constraints on the activities to perform. The total number of packets, number of flows and duration of collected data for each application is shown in Table III. The entire dataset consists of over 29 hours of collected data for all applications.

V. EXAMPLE USE CASE: APPLICATION AND ACTIVITY CLASSIFICATION

The proposed dataset provides a meaningful tool to analyzing modern mobile network traffic. In particular, the labeled dataset can be used to develop traffic classification algorithms. Two example use cases on application classification and activity classification are presented. The purpose of this section is to provide a simple example and benchmark on how the proposed dataset can be used rather than to achieve state-of-the-art performance in the classification tasks.

Both examples use the randomized automated dataset which contains 14 of the 16 applications shown in Table III, excluding Google Maps and Hangout. Per-flow features are extracted from the TCP/IP headers of downlink packets, including the total number of packets, the total number of bytes, the minimum, maximum, mean and standard deviation of packet sizes in a flow, the minimum, maximum, mean and standard deviation of the packet inter-arrival times (IAT) and the flow duration. The same set of features is used for both application and activity classification. The extracted dataset is preprocessed to remove the mean and scale to unit variance for each feature. Three ML models are considered: Random Forest Classifier (RFC), K-Nearest Neighbor (KNN) classifier and XGBoost classifier. The ML models are implemented using the scikit-learn Python package and the default hyperparameters are used. The accuracy and the F1 score are two important metrics for classification problems. Let TP_i, TN_i, FP_i and FN_i denote the true positives, true negatives, false positives and false negatives of class *i*, the accuracy is given by

$$\text{accuracy} = \frac{\sum_i \text{TP}_i + \text{TN}_i}{\sum_i \text{TP}_i + \text{TN}_i + \text{FP}_i + \text{FN}_i}. \tag{1}$$

The F1 score of class *i* is given by

$$\text{F1 score} = \frac{2\text{TP}_i}{2\text{TP}_i + \text{FP}_i + \text{FN}_i}. \tag{2}$$

TABLE IV
APPLICATION CLASSIFICATION ACCURACY (%)

Classifier	RFC	KNN	XGB
Accuracy	78.9 ± 2.0	70.0 ± 1.8	79.1 ± 1.9

A. Application Classification

The mean and standard deviation of the accuracy from 10-fold cross-validation are shown in Table IV. The RFC and the XGBoost classifier which are both ensemble models achieve an average accuracy of around 79% and outperform the KNN by a significant margin. The mean and standard deviation of the F1 scores for each application are shown in Fig. 3. Again, the RFC and the XGBoost classifier outperform the KNN in terms of the F1 score for most applications. The ML models can classify applications such as YouTube, Spotify, Dropbox and Pandora relatively well, achieving F1 scores of over 0.8 for these applications. On the other hand, the ML models can only achieve F1 scores of less than 0.6 for applications such as Gmail and Messenger. This might be caused by a lack of training data since these applications have fewer packets and flows in the dataset.

B. Activity Classification

Assuming that the application is known, an activity classifier is trained for each application to detect the activity performed. The applications with multiple types of captured

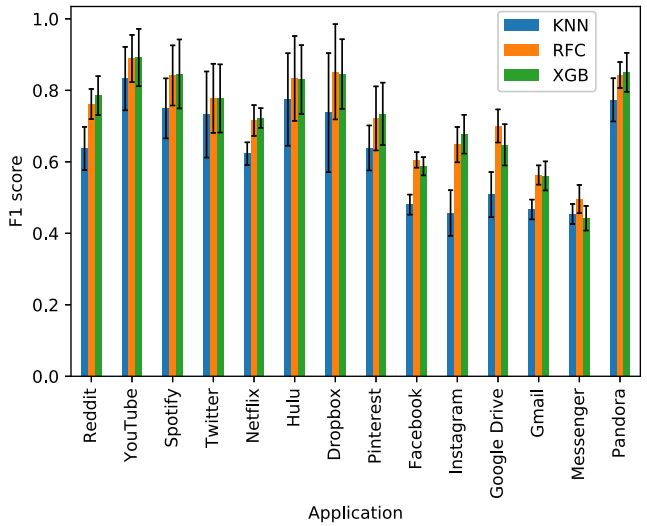

Fig. 3. F1 score mean and standard deviation of application classification from 10-fold cross-validation.

TABLE V
ACTIVITY CLASSIFICATION ACCURACY PER APPLICATION (%)

App	# of classes	RFC	KNN	XGB
Instagram	2	99.4 ± 1.7	97.2 ± 3.7	95.9 ± 5.9
Hulu	2	93.8 ± 3.1	91.1 ± 4.6	92.6 ± 3.5
Netflix	2	85.8 ± 7.7	80.2 ± 8.8	87.3 ± 7.7
YouTube	2	82.8 ± 7.0	78.4 ± 5.4	82.1 ± 6.5
Gmail	2	82.1 ± 9.8	78.4 ± 7.6	82.6 ± 10.3
Dropbox	2	82.0 ± 4.2	74.4 ± 8.1	81.3 ± 8.1
Spotify	2	79.3 ± 5.8	73.8 ± 5.5	79.3 ± 6.2
Google Drive	2	74.5 ± 13.3	66.1 ± 5.6	67.9 ± 10.1
Pandora	2	73.5 ± 4.5	68.3 ± 4.0	73.3 ± 4.4
Reddit	2	70.2 ± 7.4	70.2 ± 7.1	70.0 ± 7.1
Twitter	2	68.6 ± 4.5	66.8 ± 3.5	68.1 ± 6.1
Facebook	2	65.6 ± 11.7	70.8 ± 8.4	62.8 ± 11.8

activities in the randomized automated dataset are considered. The mean and standard deviation of the accuracy from 10-fold cross-validation of each application are shown in Table V. The ML models can classify activities relatively well for applications such as Instagram and Hulu but poorly for Facebook and Twitter. Activities of an application like Instagram are intuitively easier to be distinguished from one another: browsing photos and videos generates heavy traffic while sending messages does not. On the other hand, for an application like Facebook, scrolling the news feed and searching for Facebook pages generate more similar traffic patterns and are more difficult to classify using per-flow statistical features.

VI. CONCLUSION

We present a public labeled dataset for network traffic classification. An automated platform is constructed to generate and collect mobile traffic data from specified applications in a controlled environment. The dataset consists of mobile data from a wide range of 16 popular applications generated by both the automated platform and human users. The dataset is labeled with application and activity level labels. Examples on application and activity classification are provided, where ML models are trained using flow-based features extracted from the dataset.

ACKNOWLEDGMENT

The authors thank C. Gill, K. Chau, N. Charles, A. Ma, J. M. Noh and J. Poudel of UT Austin for their assistance in developing the data collection platform and collecting the dataset. The support of Samsung, including smartphone donations for the experiments, is also gratefully acknowledged.

REFERENCES

[1] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," *Int. J. Netw. Manag.*, vol. 25, no. 5, pp. 355–374, 2015.

[2] P. Wang, X. Chen, F. Ye, and Z. Sun, "A survey of techniques for mobile service encrypted traffic classification using deep learning," *IEEE Access*, vol. 7, pp. 54024–54033, 2019.

[3] V. Chandrasekhar, Y. Heng, J. Cho, J. Lee, J. Zhang, and J. G. Andrews, "Experience-centric mobile video scheduling through machine learning," *IEEE Access*, vol. 7, pp. 113017–113030, 2019.

[4] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for Internet of Things," *IEEE Access*, vol. 5, pp. 18042–18050, 2017.

[5] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *Proc. PAM*, 2005, pp. 41–54.

[6] V. Tong, H. A. Tran, S. Souihi, and A. Mellouk, "A novel QUIC traffic classifier based on convolutional neural networks," in *Proc. IEEE GLOBECOM*, 2018, pp. 1–6.

[7] R. Alshammari and A. N. Zincir-Heywood, "Can encrypted traffic be identified without port numbers, IP addresses and payload inspection?" *Comput. Netw.*, vol. 55, no. 6, pp. 1326–1350, 2011.

[8] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features," in *Proc. ICISSP*, 2016, pp. 407–414.

[9] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *Proc. ICISSP*, 2017, pp. 253–262.

[10] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proc. IEEE ICOIN*, 2017, pp. 712–717.

[11] D. Li, Y. Zhu, and W. Lin, "Traffic identification of mobile apps based on variational autoencoder network," in *Proc. IEEE CIS*, 2017, pp. 287–291.

[12] J. S. Rojas, Á. Rendón, and J. C. Corrales, "Consumption behavior analysis of over the top services: Incremental learning or traditional methods?" *IEEE Access*, vol. 7, pp. 136581–136591, 2019.

[13] R. Wang, Z. Liu, Y. Cai, D. Tang, J. Yang, and Z. Yang, "Benchmark data for mobile app traffic research," in *Proc. EAI MobiQuitous*, Nov. 2018, pp. 402–411.

[14] G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapé, "MIRAGE: Mobile-app traffic capture and ground-truth creation," in *Proc. IEEE ICCCS*, Oct. 2019, pp. 1–8.

[15] T. Karagkioules *et al.*, "A public dataset for YouTube's mobile streaming client," in *Proc. IEEE TMA*, Jun. 2018, pp. 1–6.

[16] S. C. Madanapalli, H. H. Gharakhieli, and V. Sivaraman, "Inferring Netflix user experience from broadband network measurement," in *Proc. IEEE TMA*, Jun. 2019, pp. 41–48.

[17] A. Sivanathan *et al.*, "Classifying IoT devices in smart environments using network traffic characteristics," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1745–1759, Aug. 2019.

[18] *Statista: Unique U.S. Visitors to Mobile Apps 2016*. Accessed: Mar. 1, 2019. [Online]. Available: http://www.statista.com/statistics/250862/unique-visitors-to-the-most-popular-mobile-apps-in-the-us/