Mobile traffic classification through burst traffic statistical features

Cesar Vargas Anamuro, Xavier Lagrange

▶ **To cite this version:**

Cesar Vargas Anamuro, Xavier Lagrange. Mobile traffic classification through burst traffic statistical features. VTC2023-Spring 2023: IEEE 97th Vehicular Technology Conference, Jun 2023, Florence, Italy. 10.1109/VTC2023-Spring57618.2023.10200032 . hal-04154479

HAL Id: hal-04154479

https://imt-atlantique.hal.science/hal-04154479v1

Submitted on 6 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License

Mobile traffic classification through burst traffic statistical features

Cesar Vargas Anamuro, Xavier Lagrange
IMT Atlantique/IRISA
Rennes, France
{cesar.vargasanamuro, xavier.lagrange}@imt-atlantique.fr

Abstract

Mobile traffic classification is a topic of interest for researchers focused on improving the network capacity or for those seeking to identify potential risks to users' privacy. In recent years, traffic classification accuracy has significantly improved thanks to machine learning techniques. These techniques allow traffic identification even if it is encrypted, as in mobile networks. In this paper, we show that it is feasible to classify mobile traffic applications with high accuracy using downlink control information (DCI) messages and machine learning. The DCI messages are collected using a sniffer located near the base station. Then we extract the statistical features of the bursts and inter-burst periods of the traffic generated by mobile applications at the physical layer. This strategy uses few features and does not require a big dataset. We have tested our approach on a 4G cellular network testbed and a commercial 4G cellular network. The results show an accuracy greater than 92% and 95% for application and category classification, respectively.

1 Introduction

The rapid evolution of cellular networks in recent years has boosted the appearance of many new services and applications. Currently, smartphones play an essential role in everyday life of users. The popularity of these devices is due to the variety of applications for work, communication, or leisure activities.

Mobile traffic classification is the association of mobile traffic with the application that generates this traffic. Some researchers propose traffic classification to improve the network capacity [1] and the Quality of Service (QoS) requirements according to the application. The authors of [2] use traffic classification to estimate the amount of traffic generated by the most popular applications. Other studies focus on identifying potential risks to users' privacy [3], and energy saving [4].

In the literature, traffic classification techniques are typically divided into three categories: port-based, payload-based, and flow statistics-based [5]. Port-based techniques consist of checking the ports used by applications. Payload-based classification uses information extracted from the packet header and payloads, but this approach requires deep packet inspection. The two categories above are unusable when the traffic is encrypted below the transport layer, as in mobile traffic. Flow statistics-based techniques do not require deep packet inspection; thus, they can be used even if the traffic is encrypted. This approach only uses statistical flow features, such as packet inter-arrival times, average packet size, and the number of bytes and packets per flow.

The use of machine learning techniques to identify mobile applications has been the subject of numerous studies. Most of these studies focus on the information carried by higher protocol stack layers. Moreover, these studies use flow statistics-based techniques to overcome the challenge of encrypted mobile traffic. The authors of [6–9] use an approach based on extracting traffic burst features to identify traffic generated by mobile applications. In these studies, a flow consists of successive IP packets that share the same source and destination IP addresses, source and destination port numbers, and transport protocol. The authors of [6] propose the application classification over 802.11 traffic. In [7], the authors show the effect of time, different devices, and different versions of applications on the classification metrics. The authors of [8] analyze the impact of the burst threshold (the maximum period between two consecutive packets that belong to the same burst) on the application classification performance. These studies demonstrate that if the monitored traffic is divided into flows, it is possible to identify the mobile applications that generate the traffic with high accuracy. However, due to the encryption and authentication used by the cellular network, the information of the upper layers (IP addresses, port numbers, and transport protocol) is not accessible to users (sniffers) outside the communication. A sniffer outside the Mobile Network Operator (MNO) can only monitor the physical layer. For this reason, the authors of [2] propose the classification of mobile traffic through a Convolutional Neural Network (CNN) based on the information of the control messages transported by the Physical Downlink Control Channel (PDCCH). However, this approach requires a large number of labeled samples to achieve good classification performance. The dataset used to evaluate the proposed architecture comprises more than 11,000 mobile sessions generated

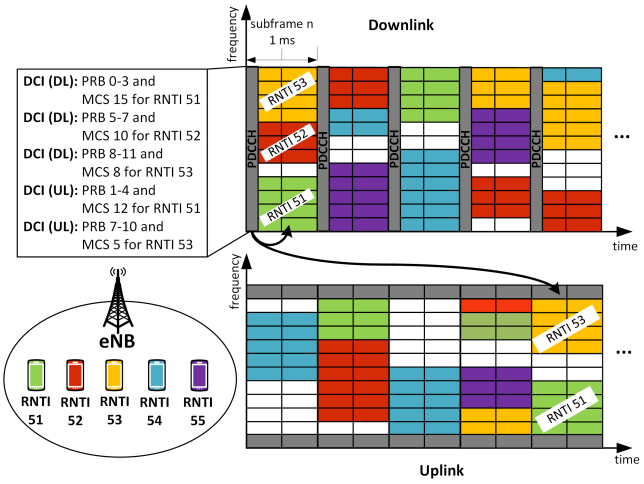

Figure 1: PRB allocation, network with five active terminals.

by a terminal connected to a commercial 4G cellular network. In each session, the terminal connects to the Base Station (BS) and generates traffic for a specific mobile application. This dataset is not available to the public.

In this paper, we present a high-accuracy and low-complexity (with limited features and labeled samples) mobile traffic classification scheme. Our approach is based on supervised machine learning and the statistical characteristics of the bursts and inter-burst periods of the traffic generated by mobile applications at the physical layer. We compare five well-known machine learning algorithms: Random Forest, Decision Tree, k-Nearest Neighbors, Logistic Regression, and Gaussian Processes. To evaluate the algorithm's performance, we use the k-fold cross-validation method.

Due to the lack of datasets containing real data from mobile networks, we first collected data from cellular mobile traffic using an LTE sniffer developed by our research team. The source code has been made publicly available at `https://gitlab.imt-atlantique.fr/cvargasa/lte_analyzer`. Our LTE sniffer, based on the open-source srsRAN library [10], captures all the messages broadcasted by the BS in clear text, among which the Downlink Control Information (DCI) messages are the concern of our study. We collected two datasets corresponding to the traffic generated by terminals connected to two 4G mobile networks. The first is a testbed based on the Amarisoft platform [11] implemented in the laboratory. The second is a commercial 4G cellular network belonging to a French MNO.

The remainder of the paper is organized as follows: Section 2 gives an overview of the control channel in LTE. Section 3 describes the data collection methodology used in this study. Section 4 discusses the obtained results. Finally, Section 5 concludes our work.

2 Control Channel in LTE

LTE transmissions are structured into 1-millisecond subframes. In the downlink (DL), every subframe is divided into data and control regions. The control region is occupied by several physical control channels, such as the PDCCH. The PDCCH is used to transport the DCI, which carries DL scheduling assignments and uplink (UL) scheduling grants for the terminals connected to the BS. A DCI message also contains the Modulation and Coding Scheme (MCS) index, which, together with the number of allocated Physical Resource Blocks (PRB), is used to derive the Transport Block Size (TBS). A 16-bit Radio Network Temporary Identifier (RNTI) identifies the destination terminal of DCI messages. The BS assigns an RNTI to the terminal once the Radio Resource Control (RRC) connection is established; the terminal uses the RNTI until the end of the RRC connection.

In every subframe, the BS informs connected terminals of the DL scheduling assignment (and the MCS index) for the current subframe n and the UL scheduling grant (and the MCS index) for the subframe $n+4$ [12]. As an example, Fig. 1 shows the resource allocation within a cell in which five terminals are connected to BS. The BS transmits five DCI in the PDCCH region of subframe n. In subframe n, PRBs 0-3, 5-7, and 8-11 in the DL are reserved for terminals 51, 52, and 53, respectively. In subframe $n+4$, PRBs 1-4 and 7-10 in the UL are reserved for terminals 51 and 53, respectively.

It should be noted that the DCI messages do not indicate any information that allows associating the RNTI with a specific terminal or the application running in the terminal. Hence the need to use statistical properties of the encrypted traffic for traffic classification.

In LTE, the RRC inactivity timer parameter fixes the maximum inactivity time allowed in the cell. This

Table 1: Mobile data traffic by application category.

Category	Applications
Audio streaming	Spotify, Youtube music
Social networking	Instagram, Tiktok
Video calling	Messenger, Skype
Video streaming	Youtube, Vimeo
Voice calling	Messenger, Skype

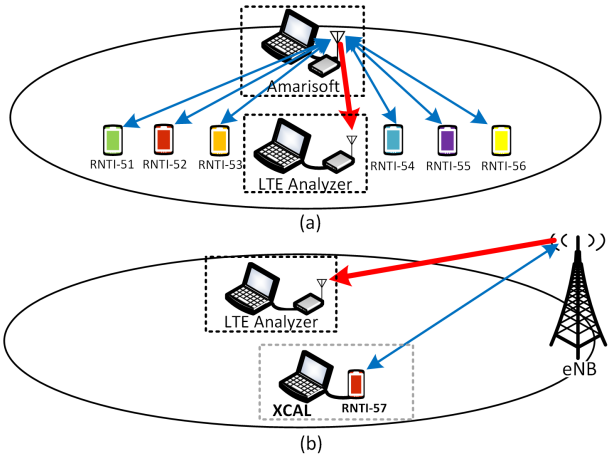

Figure 2: Data collection scenarios, (a) testbed and (b) commercial MNO. Terminals connected to the network run a specific application.

parameter depends on the MNO configuration (usually set to a few seconds). The value is not broadcast but can be inferred by analyzing mobile traffic measurements. The BS terminates the connection when the silence duration exceeds the inactivity timer parameter. In the case of burst traffic, the traffic generated by a given application can be divided into several RRC connections (different RNTI values).

3 Data Collection Methodology

3.1 Data Collection Scenarios

DCI messages were collected using an LTE analyzer (sniffer) developed by our research team. This sniffer generates a comma-separated values (CSV) file, where each row is the decoded information of a DCI. Columns contain a Unix timestamp, the RNTI associated with the DCI, the transmission direction, the MCS, and the TBS. We generate traffic using terminals connected to the cellular network that run a specific application. At the same time, the LTE analyzer collects the DCI messages inside the cell.

We consider only the categories of applications that generate the highest volume of mobile traffic [13], which we divide into five categories: audio streaming, social networking, video calling, video streaming, and voice calling (see Section 3.3). We analyze two different applications for each category, detailed in Table 1.

In this study, we consider two data collection scenarios. The first is a 4G network testbed based on the Amarisoft solution, which provides eNodeB (eNB) and core network functionalities running on a personal computer (see Fig. 2 (a)). The advantage of this testbed is that we can control all the terminals that connect to the cellular network. DCI messages collected by the LTE analyzer are labeled according to the executed application. We use six smartphones belonging to five different brands (Motorola, iPhone, Samsung, OnePlus, and Nexus). The second scenario corresponds to a commercial cellular network (see Fig. 2 (b)). In this case, labeling the connections is more difficult since the CSV file generated by the LTE analyzer contains the DCI messages of all terminals connected to the eNB. Thus, we have to filter only the DCIs corresponding to the terminal we use to generate traffic. To do this, we use a commercial drive-test tool called Accuver XCAL [14], which records the messages exchanged between the terminal and the eNB, among others, the random access response (RAR). The RAR message carries the RNTI assigned to the terminal. Using this information, we can correlate the application executed by the terminal and the DCI messages collected with the LTE analyzer.

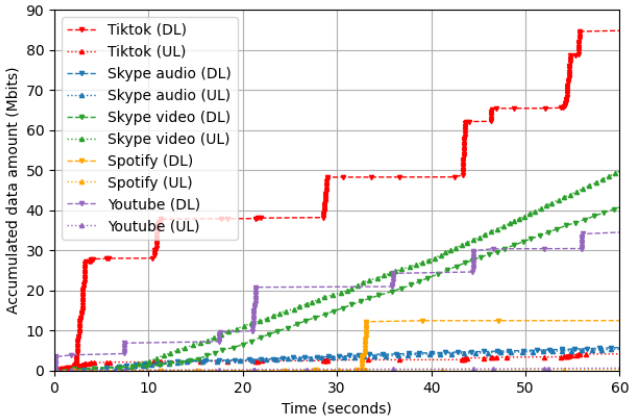

Figure 3: Accumulated data amount versus time.

3.2 Labeled Datasets

We generated two datasets from the two scenarios considered in this study. The first corresponds to DCI messages collected from the testbed. We fixed the bandwidth to 15 MHz in Frequency Division Duplex (FDD) mode in the 2.6 GHz band. The RRC inactivity timer was set to 10 seconds. The testbed dataset contains 2,100 labeled connections from ten applications (210 per application). The second dataset corresponds to the connections within a commercial MNO. The measurements were collected in a cell of the SFR MNO in the city center of Rennes, France. The MNO fixes the bandwidth to 15 MHz in FDD mode in the 2.6 GHz band, and the RRC inactivity timer parameter varies between 5 and 10 seconds. The commercial MNO dataset contains 2,100 labeled connections (210 per application).

During data collection, the applications run for periods that vary between 3 and 10 minutes. From the datasets, we only use the following information: the timestamp, the RNTI, the direction (UL or DL), and the TBS. This information allows us to analyze the data amount variation for each connection as a function of time in both directions.

3.3 Application Grouping

We can group mobile applications based on how users interact with them, as many of these applications share common features. In voice or video calling applications, the user transmits and receives data almost in the same amount. Other applications, such as streaming and social networking, mainly generate DL traffic. While users interact highly with social networking applications, streaming applications do not require high user interaction.

Fig. 3 shows examples of cellular traffic generated by Tiktok, Skype voice, Skype video, Spotify, and Youtube. We notice that each application needs a specific throughput. Voice and video calling generate frequent transport blocks in both directions. Video is much more throughput-demanding than voice calling. These characteristics facilitate the identification of these applications. In contrast, the other applications generate burst traffic mainly in the DL. We noticed that during inter-burst periods, there are some small transport blocks, which probably carry signaling messages. Tiktok has more signaling messages than Youtube. We also see that the burst size is more variable on Tiktok than on Youtube. Spotify has only one burst and a long inter-burst period. This is because these types of applications download the content, and then the physical channel remains inactive while the user listens to the content.

3.4 Feature Extraction

We use an approach based on the statistical properties of bursts and inter-burst periods. A burst is defined as consecutive transport blocks that have an inter-arrival time shorter than a given threshold (named burst threshold). The burst threshold is a key parameter in separating traffic into bursts. In this paper, we evaluate different burst threshold values (between 10 ms and 2 seconds) and select the value that shows the best results.

From the datasets, we observe that the connection durations vary from a few seconds to a few minutes. For connections longer than three minutes, we only consider the DCI messages of the first three minutes. We consider the following connection features:

- UL and DL throughput
- Burst sizes and burst durations in the DL
- Inter-burst durations in the DL

Table 2: Category Classification, results are in the format avg. ±std. obtained over 5-folds.

	Testbed				Commercial MNO			
Method	Accuracy	Precision	Recall	F-Score	Accuracy	Precision	Recall	F-Score
Random Forest	0.95±0.01	0.94±0.03	0.88±0.04	0.91±0.02	0.95±0.01	0.94±0.03	0.91±0.05	0.93±0.03
Decision Tree	0.89±0.02	0.82±0.03	0.81±0.05	0.82±0.04	0.92±0.01	0.9±0.03	0.91±0.03	0.9±0.01
k-Nearest Neighbors	0.90±0.01	0.88±0.05	0.86±0.03	0.86±0.03	0.94±0.01	0.93±0.03	0.88±0.04	0.91±0.03
Logistic Regression	0.90±0.01	0.86±0.04	0.82±0.05	0.84±0.02	0.94±0.01	0.9±0.03	0.9±0.04	0.89±0.02
Gaussian Processes	0.77±0.01	0.38±0.03	0.99±0.01	0.55±0.03	0.87±0.02	0.41±0.02	0.99±0.01	0.58±0.02

- Transport block inter-arrival time in the DL
- Instantaneous DL throughput
- Number of transport blocks during the inter-burst period

We extract the statistical properties (mean, standard deviation, median, 10th, 25th, 75th, and 90th percentiles) of the burst sizes, burst durations, inter-burst durations, transport block inter-arrival time, instantaneous DL throughput, and the number of transport blocks during the inter-burst period. In total, we extract 44 features.

4 Classification Performance

4.1 Performance Metrics

The performance of the models is evaluated using four well-known metrics, accuracy $A_c = \frac{T_p+T_n}{T_p+T_n+F_p+F_n}$, precision $P = \frac{T_p}{T_p+F_p}$, recall $R = \frac{T_p}{T_p+F_n}$, and F-Score $F = \frac{RP}{R+P}$, where T_p, T_n, F_p, and F_n are the true positives, true negatives, false positives, and false negatives, respectively.

In order to test and evaluate the performance of the models, we use the k-fold cross-validation method. That is, the data is split into k equal subsets. The training data is selected from $k-1$ of these subsets, and the remaining subset is used as the test set. Performance statistics are calculated across all k experiments. We compare five well-known machine learning algorithms, namely, Random Forest, Decision Tree, k-Nearest Neighbors, Logistic Regression, and Gaussian Processes. In this work, the classification problem was solved using the Sci-kit Learn library [15].

4.2 Category classification

In this section, we show the results of the mobile traffic classification per category. The performance results for both datasets are shown in Table 2. We note that Random Forest is the algorithm that achieves the best performance for both datasets. This is likely due to its robustness to outliers, which are typically caused by connections with a small number of bursts. Random Forest achieves an accuracy of 95% for both testbed and commercial MNO. These results show that the extraction of bursts and inter-burst periods characteristics is an approach that highly accurately classifies mobile traffic per category.

Fig. 4 shows the confusion matrix to understand the misclassification of the commercial MNO connections made by the Random Forest algorithm. We see that the voice calling applications are classified with an accuracy of 99%. This result is expected due to the particular characteristics of these applications. On the other hand, we see that video streaming has an accuracy of 94%.

We noticed that the traffic generated by audio streaming applications presents the worst results with an accuracy of 90%. This is because these applications generate a burst and then several seconds of silence. When the silence period is too long, the BS terminates the RRC connection, and the terminal performs the random access procedure at the next burst. Consequently, the traffic generated by audio streaming applications is divided into several short connections, making it challenging to identify. We also see that this category is confused with video streaming. This is because sometimes streaming audio applications show advertising videos.

4.3 Application classification

In this section, we present the performance evaluation of application identification. This type of classification is more challenging because the applications that belong to the same category have similar characteristics. Table 3 shows the performance of the metrics for five different classifiers. For both datasets, Random Forest

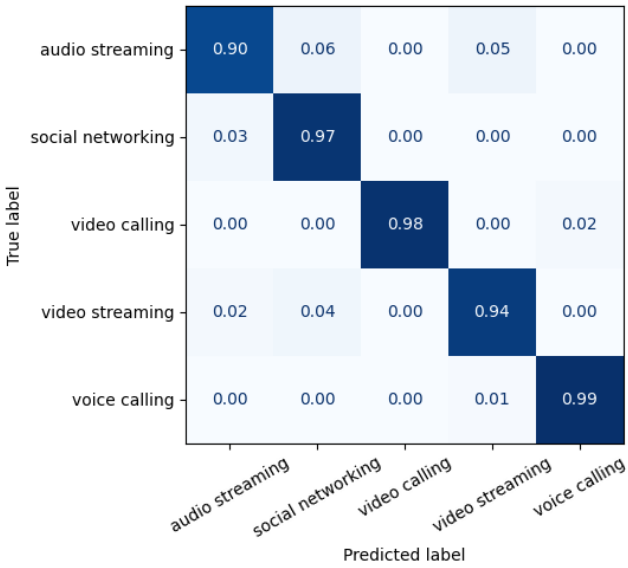

Figure 4: Confusion matrix for category identification using Random Forest, commercial MNO dataset.

Table 3: Application Classification, results are in the format avg. ±std. obtained over 5-folds.

	Testbed				Commercial MNO			
Method	Accuracy	Precision	Recall	F-Score	Accuracy	Precision	Recall	F-Score
Random Forest	0.94±0.01	0.98±0.02	0.89±0.07	0.93±0.04	0.92±0.02	0.97±0.02	0.83±0.05	0.89±0.02
Decision Tree	0.87±0.01	0.94±0.04	0.89±0.04	0.91±0.01	0.87±0.01	0.9±0.05	0.76±0.06	0.77±0.04
k-Nearest Neighbors	0.89±0.02	0.91±0.04	0.77±0.09	0.82±0.04	0.88±0.01	0.88±0.07	0.88±0.02	0.87±0.04
Logistic Regression	0.90±0.02	0.97±0.03	0.84±0.07	0.86±0.06	0.91±0.02	0.93±0.05	0.83±0.04	0.86±0.04
Gaussian Processes	0.77±0.01	0.28±0.03	0.96±0.03	0.43±0.04	0.83±0.01	0.3±0.05	0.99±0.01	0.46±0.06

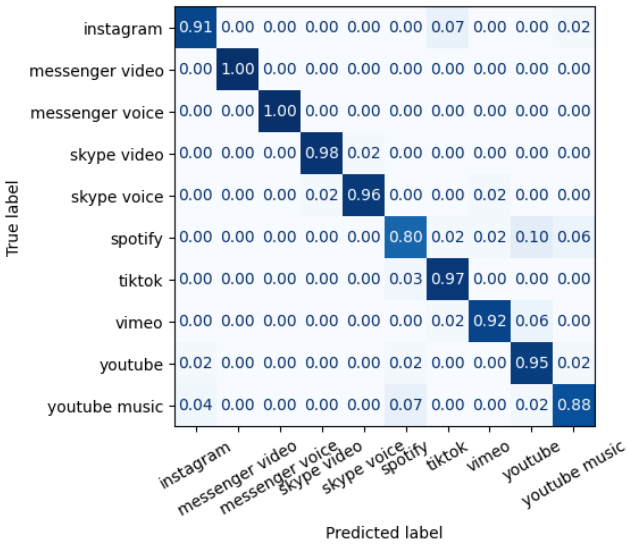

Figure 5: Confusion matrix for application identification using Random Forest, commercial MNO dataset.

is the algorithm that produces the best results with an accuracy of 94% and 92% for the testbed and the commercial MNO dataset, respectively.

Fig. 5 shows the confusion matrix to analyze which types are mismatched in the identification process. We note that the traffic generated by messenger is 100% identified in both voice and video calling. Skype traffic is identified with an accuracy of 96% and 98% in voice and video calling, respectively. We identify Tiktok and Youtube traffic with an accuracy of 97% and 95%, respectively. Spotify, with an accuracy of 80%, shows the worst results. This could be because, in some cases, connections are composed of a single burst, making it difficult to identify them.

5 Conclusion

In this paper, we focus on mobile traffic classification based on extracting characteristics of the bursts and inter-burst periods of the encrypted traffic generated by mobile applications. We compare classical machine learning algorithms such as Random Forest, Decision Tree, k-Nearest Neighbors, Logistic Regression, and Gaussian Processes. In a 4G cellular network testbed, we achieved an accuracy of 95% and 94% per category and application, respectively. On the other hand, considering a commercial cellular network, we obtained 95% and 92% accuracy per category and application, respectively. The results show that Random Forest achieves the best accuracy. Voice and video calling applications are identified with high accuracy. Audio streaming is the most challenging application category to identify due to the long inter-burst periods that characterize this category.

An extension of this work is the analysis of the effects of the RRC inactivity timer parameter in the application classification.

References

[1] B. Yamansavascilar, M. A. Guvensan, A. G. Yavuz, and M. E. Karsligil, "Application identification via network traffic classification," in *2017 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2017, pp. 843–848.

[2] 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.

[3] 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.

[4] A. Azari, F. Salehi, P. Papapetrou, and C. Cavdar, "Energy and resource efficiency by user traffic prediction and classification in cellular networks," *IEEE Transactions on Green Comm. and Networking*, vol. 6, no. 2, pp. 1082–1095, 2021.

[5] J. Zhang, X. Chen, Y. Xiang, W. Zhou, and J. Wu, "Robust network traffic classification," *IEEE/ACM transactions on networking*, vol. 23, no. 4, pp. 1257–1270, 2014.

[6] 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 comm. and network security (CNS)*. IEEE, 2015, pp. 433–441.

[7] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, "Robust smartphone app identification via encrypted network traffic analysis," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 63–78, 2017.

[8] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Multi-classification approaches for classifying mobile app traffic," *Journal of Network and Computer Applications*, vol. 103, pp. 131–145, 2018.

[9] T. Stöber, M. Frank, J. Schmitt, and I. Martinovic, "Who do you sync you are? smartphone fingerprinting via application behaviour," in *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, 2013, pp. 7–12.

[10] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. J. Leith, "srslte: An open-source platform for lte evolution and experimentation," in *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*, 2016, pp. 25–32.

[11] "Amarisoft: Amari lte 100, software lte base station on pc," https://www.amarisoft.com, accessed: December 2022.

[12] E. Dahlman, S. Parkvall, and J. Skold, *4G LTE/LTE-Advanced for mobile broadband*. Oxford: Academic press, 2011.

[13] "Ericsson mobility report june 2022," https://www.ericsson.com/en/reports-and-papers/mobility-report/reports/november-2022, accessed: December 2022.

[14] "Accuver xcal," https://www.accuver.com, accessed: December 2022.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.