IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 1, 2020, accepted August 18, 2020, date of publication September 9, 2020, date of current version September 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3022980

Encrypted Traffic Classification Based on Unsupervised Learning in Cellular Radio Access Networks

CAROLINA GIJÓN, MATÍAS TORIL, MARTA SOLERA, SALVADOR LUNA-RAMÍREZ, AND LUIS ROBERTO JIMÉNEZ
Communications Engineering Department, University of Málaga, 29071 Málaga, Spain

Corresponding author: Carolina Gijón (cgm@ic.uma.es)

This work was supported in part by the Spanish Ministry of Science, Innovation and Univessities under Grant RTI2018-099148-B-I00, and in part by the Spanish Ministry of Education, Culture and Sports under Grant FPU17/04286.

ABSTRACT Traffic classification will be a key aspect in the operation of future 5G cellular networks, where services of very different nature will coexist. Unfortunately, data encryption makes this task very difficult. To overcome this issue, flow-based schemes have been proposed based on payload-independent features extracted from the Internet Protocol (IP) traffic flow. However, such an approach relies on the use of expensive traffic probes in the core network. Alternatively, in this paper, an offline method for encrypted traffic classification in the radio interface is presented. The method divides connections per service class by analyzing only features in radio connection traces collected by base stations. For this purpose, it relies on unsupervised learning, namely agglomerative hierarchical clustering. Thus, it can be applied in the absence of labeled data (seldom available in operational cellular networks). Likewise, it can also identify new services launched in the network. Method assessment is performed over a real trace dataset taken from a live Long Term Evolution (LTE) network. Results show that traffic shares per application class estimated by the proposed method are similar to those provided by a vendor report.

INDEX TERMS Traffic classification, radio access network, trace, unsupervised learning, clustering.

I. INTRODUCTION

In the last years, the success of smartphones and tables has opened up a new world of exciting applications for mobile users. The global mobile application market size was valued at $106.27 billion in 2018, and projected to reach $407.31 billion by 2026 [1]. This trend will continue with future 5G systems, whose improved connectivity will be exploited to create innovative use cases. All these changes have forced cellular operators to change the way they manage their systems from a network-centric to a user-centric approach focused on Quality of Experience (QoE) [2].

For the above purpose, operators need proper tools to predict, monitor and control the QoE offered to their customers [3]. With recent advances in information technologies and data science, the newest traffic monitoring and analysis solutions can leverage the huge amount of information from network elements and interfaces in mobile networks [4]. To make the most of these tools, it is key to identify the service demanded by the user at all times. In this context, traffic classification aims to associate network traffic with the underlying generating application. In current cellular networks, accurate traffic classification can benefit many network management tasks, such as capacity planning, traffic policy and charging, troubleshooting or QoE management. For this reason, in Long Term Evolution (LTE) systems, each connection has a Quality-of-service Class Identifier (QCI), used for prioritizing services [5]. Such information is registered in measurements collected by radio network elements. However, even if some QCIs are associated to a single service, other QCIs comprise services of very different nature. In particular, QCIs 6, 8 and 9 comprise a mixture of multimedia, interactive and Transmission Control Protocol (TCP)-based services, namely instant messaging, streaming, web surfing or app download. Such a coarse granularity complicates any application-oriented task. Then, more precise traffic

The associate editor coordinating the review of this manuscript and approving it for publication was Qilian Liang.

classification methods are required. In future 5G networks, identifying the traffic mix will be key to design fine-grained slices with QoE control, mobile edge/multi-access computing and network functions optimized per service [6].

In legacy Internet Protocol (IP)-based networks, traffic was classified in the past by port number [7]. Such an approach is today unreliable due to the proliferation of new applications with non-standard or randomly generated ports [8]. As an alternative, payload-based methods (e.g., deep packet inspection) match the IP packet payload with a set of stored signatures to classify network traffic [9]. However, such an approach requires high storage and processing capacity, and it is useless for encrypted traffic [10]. To solve these limitations, several works tackle traffic classification by analyzing payload-independent flow characteristics. These techniques rely on the fact that traffic from different applications typically have distinct flow patterns (a.k.a., app fingerprints), which can be detected by Supervised Learning (SL) algorithms [11]. Unfortunately, SL-based classifiers exclusively consider services included in the training dataset, and are therefore unable to identify new services arising in the network. Moreover, these methods depend on large quantities of labeled data, which are difficult to obtain. For these reasons, the design of semi-supervised [12] or unsupervised [13] schemes is considered a promising research direction [14]. Nonetheless, in the particular case of mobile networks, both supervised or unsupervised flow-based traffic classification require probes that analyze traffic in the core network. In practice, operators are reluctant to install such probes because of the high associated costs. As an alternative, it is possible to process connection traces collected in the radio interface by means of big data analytics techniques. Such very detailed information can be used to classify traffic without investing in network probes.

In this work, an offline method for coarse-grained encrypted traffic classification in cellular radio access networks is presented. The method relies on unsupervised learning to classify traffic into broad service classes. Unlike classical approaches, based on IP traffic analysis by probes in the core network, the proposed method uses traffic descriptors from connection traces in the radio interface to perform the classification. Likewise, it can be applied in the absence of labeled data (seldom available in mobile networks) and identify new types of services launched in the network. Validation is performed over a dataset from a live LTE network. The main contributions of this work are: a) the definition of a set of traffic descriptors to classify and characterize traffic in the radio interface, and b) a method for coarse-grained encrypted traffic classification in absence of labeled data based on such descriptors. The proposed offline method is conceived to make the most of existing trace datasets for QoE-driven network planning and optimization.

The rest of the document is organized as follows. Section II presents an overview of related work. Section III introduces some key concepts related to the proposed classification method, described in section IV. Section V presents the validation of the method in a live LTE network. Finally, section VI summarizes the main conclusions.

II. RELATED WORK

Encrypted traffic classification has been extensively covered in the literature. In fixed networks, several flow-based methods have been proposed to classify traffic in real time by using the first packets of the flow (early classification) [15], [16] or offline based on the whole flow (late classification) [9]. These approaches have also been extended to wireless networks, by leveraging the ability of SL to identify app fingerprints. In [17], a device-fingerprinting scheme based on learning traffic patterns of background activities is proposed. The method uses support vector and k-nearest neighbors classifiers, trained with data from 20 users with different combinations of apps connected to a 3G network. In [18], six types of mobile applications are identified by analyzing the packet size and transmission direction of the first 20 packets as input features of a hidden Markov model. In [11], a framework for fingerprinting and identification of mobile apps is presented based on decision trees and support vector classifiers trained with statistical flow features grouped based on timing and destination IP address/port. In [19], the same framework is used to assess the degradation of classification performance due to changes in app fingerprints. In [20], an ensemble approach combining different state-of-the-art classifiers is proposed. Four classes of combination techniques are compared, differing in accepted classifiers' outputs, training requirements and learning scheme. Validation on a dataset of real user activity shows higher accuracy compared to the individual use of the considered classifiers.

App fingerprints vary significantly with time due to terminal evolution, app updates, user behavior, etc. Thus, classification models must be retrained with new data periodically [19]. To overcome this issue, other works propose classifiers based on deep learning, that work directly on input data by automatically distilling structured and complex feature representations at the expense of a higher training complexity and need for larger datasets [14]. In wireless networks, this approach has been considered via variational autoencoder networks [21], convolutional networks [22] or multi-modal classifiers [6], [23]. Nonetheless, as explained above, using SL flow-based classifiers in mobile networks requires a large training dataset and implies installing probes in the core network, which is undesirable for network operators. In this work, the former shortcoming is circumvented by unsupervised learning, already used for traffic classification in fixed networks [13], but not in mobile networks. For this purpose, the analysis relies on radio connection traces, which can be collected in the absence of probes in the core network.

With recent advances in data analytics, several works have considered the use of connection traces for network management in the context of self-organizing networks. Their ability to generate new indicators different from counters provided by vendors is extremely valuable for operators [24].

For instance, traces can be used in network planning to derive spectral efficiency curves required in cellular planning and simulation [25] or the spatiotemporal distribution of radio resources in a live network [26]. Likewise, traces can be used in the operational stage to tune network parameters (e.g., link adaptation offset [27], antenna tilt angle [28] or inter-system handover margin [29]) or find the root cause of problems (e.g., dropped connections [30]). In this work, information in radio connection traces is used to characterize connections from different service classes. To the authors' knowledge, no traffic classification method based on unsupervised learning over performance indicators in radio connection traces has been proposed.

III. KEY CONCEPTS

In this section, some key concepts for the proposed classification system are explained, namely radio connection traces and data encapsulation in LTE.

A. TRACES

Radio connection traces contain signaling events in the radio interface [31]. An event is a report including measurements and performance information (e.g., signal level, bit rate, etc.). Events are grouped in two categories: internal and external events. Internal events are generated by base stations (e.g., evolved Nodes B in LTE) and are specific to each vendor. In contrast, external events include signaling messages that the base station exchanges with other network equipment via standard protocols, such as Radio Resource Control (RRC) or S1 Application Protocol (S1AP). Events selected by the network operator are registered in a trace file per cell generated periodically after each reporting period (currently, 15 min). Such file is then sent to the Operation and Support System (OSS). Two types of trace files are distinguished: Cell Traffic Recording (CTR) and User Equipment Traffic Recording (UETR). While CTRs store events of all users in the cell anonymously, UETRs store information of a specific user selected by the operator [32]. In this work, for privacy reasons, CTRs are used to collect traffic descriptor statistics for all users in the network.

CTRs are binary files in ASN.1 format. To compute traffic descriptors for classification purposes, these files must be first converted into a readable format (e.g., a CSV file). Each file comprises events from users demanding services in a cell. An event includes timestamp, user identifier, cell identifier, QCI and a set of traffic parameters that differ depending on the event type. For ease of analysis, information in each file is divided per event type and synchronized. Later, user and node identifiers are used to build individual connections. A connection comprises information from a user demanding a certain service in a particular cell. Each connection includes user identifier, cell identifier and a set of traffic descriptors computed from information in events. In this work, the following traffic descriptors are considered:

- The RRC connection time, T_{RRC} [s]. A RRC connection starts when a service is requested and lasts until the user leaves the cell, the connection is closed explicitly or the user inactivity timer expires. Such a timer often has a default value of 10 s [33]. Thus, in a RRC connection of 13 s, the user may transmit during the first 3 s and the inactivity timer expires 10 s later. The connection time excluding that timer (if that is the cause of connection release) is here referred to as the effective connection time, T_{eff}.
- The total DownLink (DL) traffic volume at the packet data converge protocol level, V_{DL} [bytes].
- The UpLink (UL) traffic volume ratio, η_{UL} [%], computed as

$$\eta_{UL} = 100 \times \frac{V_{UL}}{V_{UL} + V_{DL}} . \tag{1}$$

- The DL traffic volume ratio transmitted in last Transmission Time Intervals (i.e, TTIs when the transmission buffer becomes empty), $\eta_{DL}^{lastTTI}$, computed as

$$\eta_{DL}^{lastTTI} = \frac{V_{DL}^{lastTTI}}{V_{DL}} . \tag{2}$$

- The DL activity ratio, τ_{DL}^{active}, computed as the ratio between active TTIs (i.e., those with data to transmit) and the effective duration of the connection,

$$\tau_{DL}^{active} = \frac{T_{DL}^{active}}{T_{eff}} . \tag{3}$$

- The DL session throughput, $TH_{DL}^{session}$ [bps], computed as the volume transmitted in the DL divided by the effective duration of the connection,

$$TH_{DL}^{session} = \frac{8V_{DL}}{T_{eff}} . \tag{4}$$

As shown in previous works [34], the above traffic descriptors can easily be computed per connection from information in common signaling events (e.g., connection setup, connection release, etc.). All of them are payload-independent, so they can be collected even if traffic is encrypted at application level. Moreover, most are ratios, showing similar values regardless of encryption scheme. Nonetheless, some of these descriptors are strongly influenced by radio link and network conditions (e.g., $\eta_{DL}^{lastTTI}$ and τ_{DL}^{active} depend on spectral efficiency, cell bandwidth, available user capacity, scheduling algorithm, etc.). Thus, connections of the same service might have different values of these descriptors. Likewise, connections of different services might have similar values of these indicators, making it difficult to isolate services. Hence, it is advisable to develop new traffic descriptors that are less dependent on network performance.

B. DATA ENCAPSULATION PROCESS IN LTE

To reduce design complexity, most networks are organized into protocol layers, each one built upon the one below. As a result, data generated by applications goes through an

TABLE 1. Traffic descriptors at different protocol layers for 5 different services in LTE.

		Measured					Theoretical
	Service	Instant messaging	Web (small objects)	Web (large objects)	Video streaming	App download	Full buffer
	Provider	WhatsApp	Freepik	Vimeo	YouTube	Google Play Store	–
Transport	Protocol	TCP	TCP	TCP	UDP	TCP	–
	Header [bytes]	32	32	32	8	32	32
	Max. DL payload [bytes]	147	1348	1348	1350	1348	1348
	Avg. DL packet length [bytes]	71	1139	1396	1189	1391	1348
	Max. DL packet length [bytes]	179	1380	1380	1358	1380	1380
	DL packets with MSS [%]	0	73	99	86	99	100
	No. of DL packets	27	56	2369	1988	30754	N_p
	No. of UL packets	30	39	1156	313	10991	N_p
	Ratio DL/UL packets	0.90	1.44	2.05	6.35	2.80	1
IP	Protocol header [bytes	20	20	20	20	20	20
	Max. packet length [bytes]	199	1400	1400	1378	1400	1400
PDCP	Total DL volume [kB]	1.9	63.8	3306.2	2362.9	42770.9	$1400 \times N_p$
	Total UL volume [kB]	2.5	4.9	61.9	80.08	571.5	$52 \times N_p$
	η_{UL} [%]	56.6	7.2	1.8	3.2	1.3	3.58

encapsulation process. Each layer adds a header and passes the data to the next layer, until the lowest layer is reached, where actual communication occurs through the physical medium.

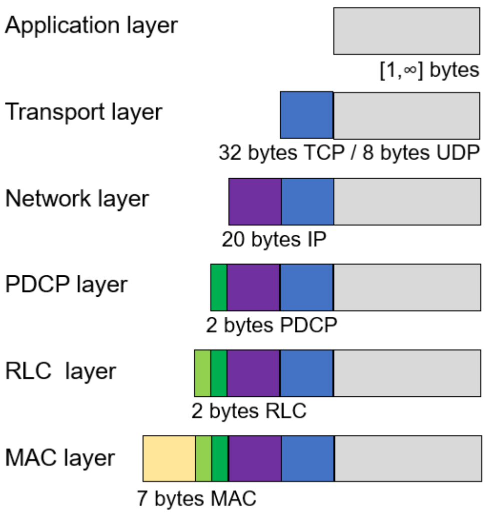

FIGURE 1. Example of packet encapsulation in the LTE user plane.

Fig. 1 shows an example of the encapsulation scheme in the user plane of the LTE radio interface. The upper level is the application layer, which contains application-specific protocols (e.g., Hypertext Transfer Protocol -HTTP-, File Transfer Protocol -FTP-, etc.). These protocols generate data packets of very different sizes. Below the application layer is the transport layer, which is responsible for transferring data between application peers. The primary two protocols on this layer are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). UDP is a stateless and connectionless option, providing fast, unreliable data transfer, suitable for streaming services. In contrast, TCP is stateful and connection-oriented, providing reliable transmission by guaranteeing in-order data delivery and retransmissions, suitable for web or file transfer. In both cases, application data packets are broken into smaller more manageable pieces. The maximum size of these pieces (a.k.a., Maximum Segment Size -MSS-) is usually restricted by the maximum transfer unit of the underlying network. In TCP, flow control uses a sliding window whose size limits how many bytes may be sent (one or more segments). When a segment is correctly received, the receiver sends an acknowledgment packet (ACK) and informs about how many bytes can still be received. Below the transport layer is the network layer, responsible for connecting devices with the Internet Protocol (IP) [35]. In the link layer, the Packet Data Converge Protocol (PDCP) transports IP datagrams, provides header compression (if required), ciphering and integrity protection. Below PDCP, Radio Link Control (RLC) segments and concatenates PDCP packets to adapt them to the transport block size in the Medium Access Control (MAC) layer. RLC has three modes of operation: transparent mode, unacknowledged mode and acknowledge mode. The latter mode is often used to deliver packets through dedicated logical channels (i.e., data traffic) [36].

The performance of the above protocols is strongly influenced by the type of service requested by the user. Different applications have different traffic characteristics and communication patterns. For instance, app or file downloads generate large packets, while messaging services generate infrequent small packets. To support this statement, Table 1 breaks down traffic descriptors at different protocol layers for 4 of the most demanded services in LTE, namely instant messaging via WhatsApp, web browsing (in two different webpages), video streaming via YouTube and app download via Google Play Store. Data in the table is obtained by analyzing traffic from live applications captured in a mobile terminal connected to a commercial LTE network. As expected, WhatsApp reports the lowest TCP packet size, with an average packet size of 71 bytes. In fact, no packet fills the transport MSS. For the rest of services (i.e., data-hungry services), the percentage of packets that fill the transport MSS varies. In app download, video streaming and web with large objects, application data chunks are large enough to fill payload in most transport packets ($\geq 86\%$). In contrast, in the case of web browsing in simple webs, only 73% of packets fill TCP payload, revealing the presence of some application data chunks with smaller size (e.g., small objects).

Different packet sizes of data-hungry services have an impact on the value of η_{UL} descriptor. This indicator reflects in which direction (i.e., UL, DL or both) data traffic is transmitted in a connection. Connections with η_{UL} close to 0%/100% belong to asymmetric download/upload services, respectively, while connections with η_{UL} close to 50% correspond to symmetric services. For download connections (the most frequent in data-hungry services), the value of η_{UL} can be approximated analytically by considering a connection with arbitrarily large application data chunks, where all transport packets are completely filled (i.e., a full buffer service). Such an example is included in column 'Full Buffer' in Table 1. V_{DL} at PDCP level is computed as the maximum TCP payload (i.e., 1348 bytes in LTE, according to measurements in Table 1) plus 32 + 20 bytes of TCP and IP headers. Likewise, V_{UL} is approximated by the size of ACK packet (52 bytes). Thus, η_{UL} results in 3.58%. Connections with η_{UL} less than that threshold belong to download services characterized by large data chunks (e.g., app download). In contrast, connections with a higher η_{UL} belong to upload services (e.g., file upload), symmetric services (e.g., video conference) or download services with smaller data chunk size (e.g., web browsing with small objects). Such a threshold is supported by measurements in Table 1. It is observed that Google Play Store, YouTube and the large web show η_{UL} below 3.58%. In contrast, the simple web shows η_{UL} above 3.58%, and WhatsApp shows $\eta_{UL} \approx 50\%$, since it is a symmetric service.

It should be pointed out that, in the analytical bound obtained for full buffer service, it is assumed that: a) there is no header compression in PDCP, which is valid for most data traffic in LTE [36], b) TCP protocol is used in the transport layer, and c) each TCP packet is acknowledged by an ACK. The latter assumptions are not always true in current networks. On the contrary, results for app download service in Table 1 show that 30,754 packets are sent in the DL and only 10,991 ACKs are sent in the UL (i.e., 1 UL ACK message acknowledges 2.8 DL packets on average). Likewise, YouTube uses UDP protocol. If some of these conditions do not hold (e.g., there is header compression, less ACKs are sent, or a different transport protocol is used), a lower value of η_{DL} will be obtained. Thus, it can be stated that connections filling most transport packets cannot have η_{DL} higher than 3.58%.

IV. CLASSIFICATION METHOD

This section describes the proposed traffic classification method. The aim of the method is: a) to divide traffic into broad application groups (e.g., messaging services, web browsing, streaming services, etc.) using information from radio connection traces provided by network operators, and b) to report the main features of each group.

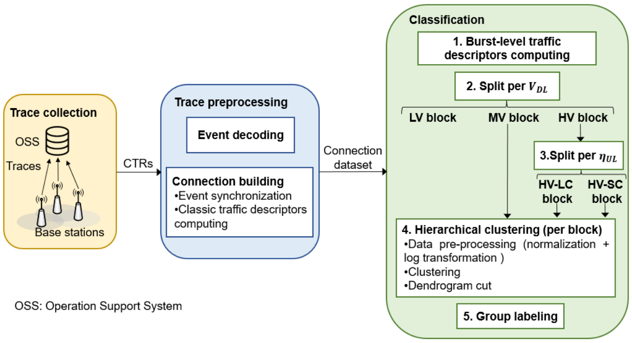

FIGURE 2. Proposed classification method.

Method structure is shown in Fig. 2. Once traces are collected and processed, a new set of traffic descriptors modeling radio connections at burst level is computed per connection and added to the dataset. Then, the dataset is broken up into disjoint groups by a 4-step procedure. It will be shown later that services offered in mobile networks are unevenly demanded (e.g., instant messaging is more demanded than file download). Performing clustering over an imbalanced dataset can lead to the classes with less members being shadowed by those with more members [37]. A common solution is to re-sample the dataset by under-sampling the classes with more data points, but this process requires labeled data, which is seldom available in mobile networks due to the difficulty of combining data from the radio access and core domains. To circumvent this problem, the connection dataset is first split into broad connection classes based on a priori knowledge. Then, connections in each broad class, from services with comparable demand, are divided into clusters by means of unsupervised clustering. Finally, the obtained groups are labeled manually by analyzing the properties of each group. A more detailed explanation of each step is given next.

A. TRAFFIC MODELING IN THE RADIO INTERFACE

Traffic carried during a connection consists of one or more data chunks sent from/to the network. As explained above, chunks generated at the application layer can be fragmented into smaller packets in lower layers. Then, as a result of packet scheduling, packets belonging to the same data chunk can be transmitted in several data bursts over the radio interface, where traces are collected [36]. Thus, a connection in the radio interface consists of a series of data bursts, characterized by three parameters: the number of bursts, N_{DL}^{burst}, the duration per burst, $T_{DL}^{burst}(n)$, and the volume per burst, $V_{DL}^{burst}(n)$ (where n denotes the burst index, since burst duration and volume may vary across bursts). Those parameters strongly depend on the service. For instance, when downloading a large file, a single data chunk is available at once at the application layer, so less bursts are likely to be transmitted than when downloading a web page comprising many small objects. Thus, the values of the above parameters can be used to isolate different services in the radio interface.

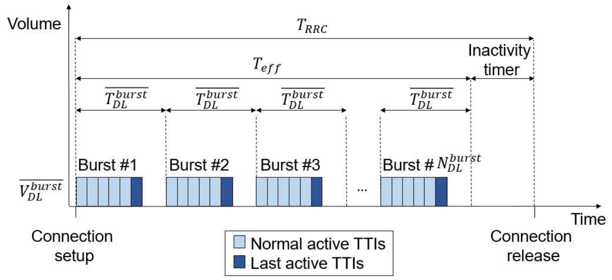

FIGURE 3. Connection model in the radio interface.

Unfortunately, radio connection traces do not explicitly register information at a burst level. As an alternative, burst level parameters can be estimated per connection from the set of traffic descriptors described in section III-A by assuming that all bursts are equal (i.e., have the same burst volume and duration), as shown in Fig. 3. First, the activity ratio of a connection k is expressed as

$$\tau_{DL}^{active}(k) = \frac{T_{DL}^{active}(k)}{T_{eff}(k)} = \frac{N_{DL}^{burst}(k)\,\overline{N_{burst\_DL}^{activeTTI}}(k)}{T_{eff}(k)} = \frac{\overline{N_{burst\_DL}^{activeTTI}}(k)}{\overline{T_{DL}^{burst}}(k)}, \quad (5)$$

where $\overline{N_{burst\_DL}^{activeTTI}}(k)$ is the average number of active TTIs per burst in DL in connection k. Likewise, by assuming that all the $N_{DL}^{activeTTI}(k)$ active TTIs in DL in a connection transmit the same data volume, $V_{DL}^{TTI}(k)$, the total volume transmitted in last TTIs in DL in the connection can be expressed as

$$V_{DL}^{lastTTI}(k) = N_{DL}^{burst}(k)\;V_{DL}^{TTI}(k) = N_{DL}^{burst}(k)\,\frac{V_{DL}(k)}{N_{DL}^{activeTTI}(k)} = N_{DL}^{burst}(k)\frac{V_{DL}(k)}{N_{DL}^{burst}(k)\,\overline{N_{burst\_DL}^{activeTTI}}(k)} = \frac{V_{DL}(k)}{\overline{N_{burst\_DL}^{activeTTI}}(k)}, \quad (6)$$

where it has been taken into account that there is only 1 last TTI per burst, and hence the number of last TTIs in DL in a connection is $N_{DL}^{burst}(k)$. Thus, the share of volume in last TTIs is given by

$$\eta_{DL}^{lastTTI}(k) = \frac{V_{DL}^{lastTTI}(k)}{V_{DL}(k)} = \frac{\frac{V_{DL}(k)}{\overline{N_{burst\_DL}^{activeTTI}}(k)}}{V_{DL}(k)} = \frac{1}{\overline{N_{burst\_DL}^{activeTTI}}(k)}. \quad (7)$$

By replacing (7) in (5), the average burst duration can be computed as

$$\overline{T_{DL}^{burst}}(k) = \frac{1}{\tau_{DL}^{active}(k)\,\eta_{DL}^{lastTTI}(k)}. \quad (8)$$

Then, the number of bursts is estimated as

$$N_{DL}^{burst}(k) = \frac{T_{eff}(k)}{\overline{T_{DL}^{burst}}(k)}, \quad (9)$$

and finally the average burst size is computed as

$$\overline{V_{DL}^{burst}}(k) = \frac{V_{DL}(k)}{N_{DL}^{burst}(k)}. \quad (10)$$

In the above equations, it is assumed that: a) every burst has the same number of active TTIs, and b) every active TTI transmits the same volume. Both statements may not be true for some connections due to changing radio conditions, TCP ramp-up or services with varying burst size (e.g., multiple objects in a web page). Nonetheless, N_{DL}^{burst}, $\overline{T_{DL}^{burst}}$ and $\overline{V_{DL}^{burst}}$ capture the general behavior of the connection, which should be enough to identify the class of services it belongs to.

B. SPLIT PER DL VOLUME

The DL volume, V_{DL}, allows to separate data-hungry services from non-data-hungry services. Specifically, connections can be split into 3 blocks:

- High Volume (HV) block, comprising connections with $V_{DL} \geq 256$ kB, belonging to data-hungry services. Such a threshold is the 5th percentile of web page size in mobile version according to a comprehensive analysis of the 400 top web sites in Alexa ranking [38] performed with the WebPageTest tool [39]. Moreover, such a threshold it is below the size of the initial data chunk of any audio or video in major streaming platforms [40], [41].
- Medium Volume (MV) block, comprising connections with 300 B $< V_{DL} <$ 256 kB. This block contains connections from applications consuming less data. The lower 300 bytes threshold is the minimum data volume exchanged by applications providing instant messaging service (Telegram, Viber, etc.), which is the less data-demanding of the most popular services in current mobile networks [42]. Such a threshold is also the maximum size of push notifications used by mobile applications to inform users of new events and updates [43].
- Low volume (LV) block, comprising connections with $V_{DL} <$ 300 B. This block contains traffic from signaling or push-up notifications.

C. SPLIT PER TRANSPORT SEGMENT SIZE

Different data-hungry services have different size of data chunks at the application layer. As explained in section III-B, such a behavior has an impact on the UL/DL volume ratio. Thus, η_{UL} can be used to split connections in HV block in two sub-blocks: a) HV-LC block, comprising connections with Heavy data Volume and Large data Chunks that tend to make the most of payload size at the transport layer, and b) HV-SC, comprising connections with Heavy data Volume and some Small data Chunks that may not fill transport packets. In section III-B, $\eta_{UL}\approx 3\%$ was computed as an upper bound for the former services.

D. AGGLOMERATIVE HIERARCHICAL CLUSTERING

Connections in MV and HV-LC are divided into groups by means of Agglomerative Hierarchical Clustering (AHC) [44]. AHC groups data points in clusters based on their similarity. The algorithm starts by treating each data point as a singleton cluster. Then, (dis)similarity between every pair of data points in the dataset is computed with a given distance metric, and the two closest clusters merged into a single cluster by means of a linkage function based on such similarity information. This process is repeated until all clusters merge into one root cluster. The result is a tree-based representation of the data, referred to as *dendrogram*.

Among the existing clustering algorithms, AHC is chosen because: a) it is able to manage datasets with clusters of different sizes (remember that, in mobile networks, services are unevenly demanded) and density (connections from a service can have very similar traffic descriptors or not), b) it does not require to specify the number of clusters in advance (in the considered problem, such information is unknown), and c) the dendrogram itself is valuable to understand the data.

Most clustering algorithms do not work effectively in high dimensional space due to the so-called *curse of dimensionality* [45]. Moreover, in clustering algorithms based on distance such as AHC, as the number of input features grows, the distances among data points become all approximately equal, and no meaningful clusters can be formed [46]. To avoid these undesirable effects, a reduced subset of the considered traffic descriptors are used as input features to AHC. Ideally, the selected traffic descriptors must fulfill that: a) they take different values for different services, b) they are insensitive to network conditions, and c) they do not provide redundant information. A preliminary analysis (not shown here for brevity) reveals that the subset comprising T_{RRC}, V_{DL}^{burst} and N_{DL}^{burst} fulfills these criteria. Then, only these 3 traffic descriptors are used as input features to AHC.

AHC assume normally distributed data. A log-transformation is performed over the 3 input features to reduce data skewness. Moreover, traffic descriptors show very different ranges of values. For higher accuracy, data is normalized so that all variables are comparable. For this purpose, a feature scaling method is used [47]. The normalized value of each descriptor, i_{norm}, is computed as

$$i_{\text{norm}} = \frac{i - i_{\min}}{i_{\max} - i_{\min}}, \tag{11}$$

where i is the original value of the descriptor (after log-transformation) and $i_{\max}$ and $i_{\min}$ are the maximum and minimum values of the descriptor in the corresponding block of connections, respectively.

For robustness, the optimal point to cut the dendrogram (i.e., the best number of clusters, N_{clust}) is found per block by checking the average silhouette score and the Calinski–Harabasz (CH) score across a wide range of cut points. Silhouette score assigns a mark between -1 and 1 to each sample in the dataset. Positive values show that a sample is well classified, whereas negative values indicate that the sample is more similar to a different cluster [48]. In contrast, CH score computes the ratio between the within-cluster dispersion and the between-cluster dispersion [49]. In both cases, the higher value, the better.

It should be pointed out that connections in LV block consist on signaling and notifications, which are often neglected in network dimensioning and QoE management. Likewise, HV-SC block is expected to include a mix of services whose traffic patterns are not distinguishable by information in traces. Thus, AHC is not applied into these blocks.

E. GROUP LABELING

Finally, the services included in each group are deduced by analyzing the median value of traffic descriptors for connections in the group.

V. PERFORMANCE ASSESSMENT

The proposed classification method is validated using connection traces from a live LTE network. For clarity, assessment methodology is first described and results are presented later.

A. ASSESSMENT METHODOLOGY

The dataset is generated from anonymous traces collected from 10 am to 11 am (busy hour) in 145 cells covering 125 km^2 in an urban area of a live LTE network. This data should be representative of the entire network traffic because: a) the time period represents a significant share of daily network traffic, and b) the area includes financial, residential and recreational districts, with different user profiles, which should reduce the influence of time of day. Events provided by the vendor in traces are:

- INTERNAL_PROC_INITIAL_CTXT_SETUP. Event reporting connection start time.
- INTERNAL_PROC_UE_CTXT_RELEASE. Event reporting connection release time and cause.
- INTERNAL_PER_UE_TRAFFIC_REP. Periodic event reporting the active number of TTIs in both UL and DL.
- INTERNAL_PER_UE_RB_TRAFFIC_REP. Periodic event with total data volume in UL and DL and data volume transmitted in last TTIs.

From those events, all the considered traffic descriptors can be computed.

Event decoding is performed by a proprietary tool provided by the network operator, and then connection building is carried out in Java for computational efficiency. The resulting dataset consists of 184,349 connections. It is expected that most traffic is encrypted by the time the dataset was collected based on reports published by popular content providers (e.g., Google [50]). As a consequence, QCI is the only information available regarding service type. The dataset comprises 11.5% of connections with QCI 1 (Voice-over-LTE), 0.1% with QCI 5 (IP Multimedia Subsystem signaling) and 88.4% with QCIs from 6 to 9 (multimedia and TCP-based services). The latter class, comprising 162,965 connections, is divided into application groups by the proposed classification method. Such a method, referred to as Enhanced Agglomerative Hierarchical Clustering (E–AHC), is compared with a naïve method, referred to as Basic Agglomerative Hierarchical Clustering B–AHC). In B–AHC, AHC is applied to the connection dataset directly (i.e., without any previous split per V_{DL} or η_{UL}). This approach, considered as a benchmark, may be taken by a practitioner with no prior knowledge on mobile networks.

AHC is implemented with the *Cluster Analysis* toolbox in Matlab [51]. In both B–AHC and E–AHC, a *ward* linkage function is used, which minimizes the total within-cluster variance by merging the pair of clusters with minimum between-cluster distance at each step. The Euclidean distance is used as distance metric [52].

In the absence of labeled data, which would require using network probes, the method is validated by checking that the groups created are consistent with the typical mobile traffic mix reported by a vendor the year when traces were collected [53].

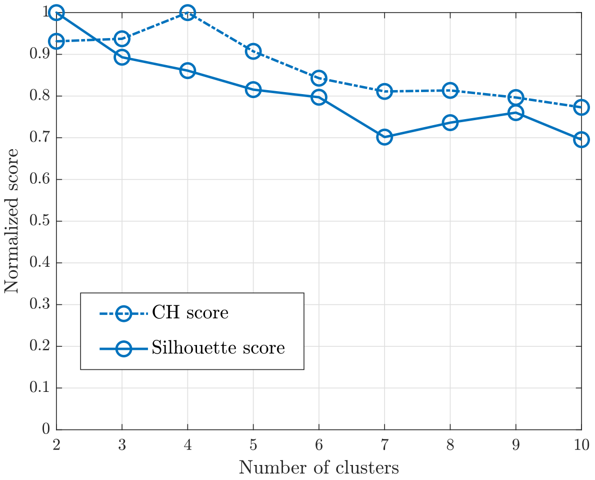

FIGURE 4. B–AHC performance with different number of clusters.

B. RESULTS: B–AHC

Fig. 4 shows the average silhouette score and the CH score obtained with B–AHC for different cuts in the dendrogram (i.e., N_{clust} choices). For a better visualization, values for each indicator are normalized by their maximum value. It is observed that, in general, the value of both metrics tend to decrease as the number of clusters increases. The higher (i.e., the best) value of CH index is obtained when $N_{clust} = 4$, whereas the silhouette value for the same choice is near to the best value (i.e., the relative value is 0.86). Thus, the connection dataset is split AHC into 4 service groups.

TABLE 2. Groups in B–AHC method.

Group	Group 1	Group 2	Group 3	Group 4
No. connections	35488	55224	51782	20471
T_{RRC} [ms]	10618	10537	14148	28460
V_{DL} [bytes]	211	288	6111	243493
η_{UL} [%]	0.65	0.47	0.36	0.10
$\eta_{DL}^{lastTTI}$ [%]	1	1	1	0.33
τ_{DL}^{active} [%]	2.4	2.3	1.2	1.9
$TH_{DL}^{session}$ [kbps]	2.27	3.41	13.97	132.04
$\overline{N_{DL}^{burst}}$	7	9	35	83
$\overline{T_{DL}^{burst}}$ [ms]	132	80	191	114
$\overline{V_{DL}^{burst}}$ [bytes]	36	32	173	2925
% of total DL volume	0.13	0.06	1.29	98.52

Table 2 breaks down the results for B-AHC with $N_{clust} = 4$. For each group, the following information is provided: a) the number of connections, b) the median value of traffic descriptors of connections in the group, and c) the percentage of DL volume carried by connections in the group. Results show that connections in groups 1 and 2 present very similar characteristics (short connections with reduced volume transmitted in last TTIs). Thus, all these connections should have been grouped into a single cluster. Moreover, group 4, comprising long data-intensive connections, has 98.52% of the total carried traffic in the DL. According to [53], no service had such an amount of traffic by the time the dataset was collected (nor currently). The large number of connections in this group (e.g., 12.56% of the total) suggests that it contain connections from several data-hungry services. These inconsistencies point out that, as expected, AHC is not performing well because the number of connections in some services is extremely large, causing that clustering is focused only on that particular service. To confirm that bad results are not due to the AHC algorithm, the experiment is repeated with other well-known clustering, namely k-means and DBSCAN [54].

The above shortcomings are solved by the proposed E–AHC method by dividing the dataset into blocks of connections based on a priori knowledge.

C. RESULTS: E–AHC

In E–AHC, the dataset is first divided into 3 blocks based on connection data volume in the DL (LV, MV and HV blocks). This split results in MV block (medium volume) with the highest number of connections (104,227 connections, 63.99% of the total), LV block (low volume) with 48,615 connections and HV block (high volume) with the lowest number of connections (7,032, a 4.32% of the total). Then, the latter block is divided according to η_{UL} value in 2 blocks: HV-SC (small data chunks), comprising 7,032 connections, and HV-LC (large data chunks), with only 3,091 connections.

TABLE 3. Groups in E–AHC method.

Block	LV	MV			HV-LC			HV-SC
Group	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8
No. connections	48615	52798	37624	13805	834	1205	1052	7032
T_{RRC} [ms]	10458	11248	17890	12337	62555	18279	21404	46335
V_{DL} [bytes]	144	797	11220	8026	11956919	1814067	2106020	97390
η_{UL}	5	48.9	35.5	25.4	2.3	2.4	2.3	7.2
$\eta_{DL}^{lastTTI}$	1	1	1	0.40	0.24	0.03	0.17	0.23
τ_{DL}^{active} [%]	2.6	1.7	1	2	9.2	10.4	6.7	2.3
$TH_{DL}^{session}$ [kbps]	2.25	5.44	16.46	37.73	2146.53	2319.8	1480.3	251.1
$\overline{N_{DL}^{burst}}$	4	18	54	18	820	24	136	165
$\overline{T_{DL}^{burst}}$ [ms]	127	67	125	144	57	363	80	203
$\overline{V_{DL}^{burst}}$ [bytes]	33	40	199	496	12793	84223	16305	6487
% of DL volume	0.02	0.15	2.57	0.74	41.81	9.21	8.33	37.17
Service	Push notifications	Instant messaging	Instant messaging	File sharing	Streaming	Full buffer services	Web browsing	Web browsing and RRSS

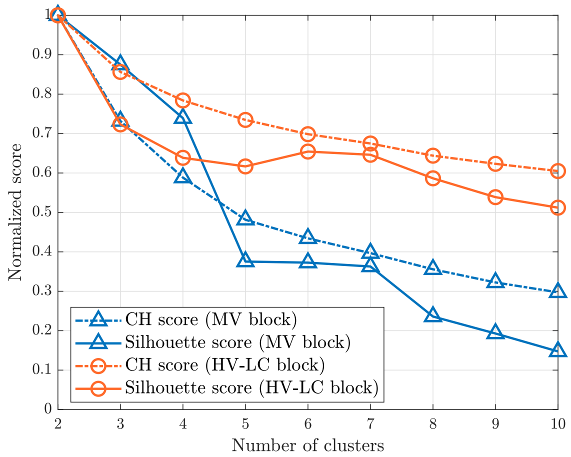

FIGURE 5. E–AHC performance with different number of clusters.

Fig. 5 shows the relative average silhouette and CH scores obtained when cutting the dendrograms of MV and HV-LC blocks at different N_{clust} values. The highest values of both scores are obtained with $N_{clust} = 2$. However, this solution is discarded, since it provides a too coarse classification. For $N_{clust} = 4$, CH score in MV block has a value of less than 0.6 compared to the maximum, which is unacceptable. Likewise, in HV-LC block, a deeper analysis of silhouette score (not shown here) reveals that the number of samples with a negative silhouette score value (i.e., which should be assigned to a different cluster) strongly increases at that point, which is undesirable. Larger number of clusters lead to worse performance. Thus, $N_{clust} = 3$ is selected as the cut point for both MV and HV-LC blocks.

Table 3 presents the 8 connection groups obtained at the end of the classification process. For each group, it provides: a) the block to which the group belongs, b) the number of connections, c) the median value of traffic descriptors of connections in the group, d) the percentage of the total DL volume carried by connections in the group and e) the underlying service, guessed by analyzing such values. Groups are analyzed next.

Connections in LV block (≈30% of the total) make up group 1. This group consists of very short connections (T_{RRC}<11 s and, hence, T_{eff}≈1 s) with few data (≈150 B in both UL and DL), all transmitted in last TTIs ($\eta_{DL}^{lastTTI} = 1$). As a consequence of the low transmitted data, session throughput is very low (≈2 kbps). Such a description fits with push notifications, consisting of lightweight audio or visual cues sent by specific servers (e.g., Google Cloud Messaging Server) to inform users about unread messages or updates in applications [43]. This group may also include some radio connections comprising only a TCP FIN or RESET packet, appearing when these packets are delayed more than the user inactivity timer [33]. In this case, a TCP connection is split in 2 connections over the radio interface (one with the main TCP data flow and another with the FIN or RESET message). Note that this group with push notifications is the second largest in the mobile network under analysis.

MV block is split in groups 2 to 4. Group 2 presents the highest number of connections (about 33% of the total) with a short RRC connection time, low traffic volume (≈800 B) and 100% of data transmitted in last TTIs. The fact that T_{RRC} is very close to the inactivity timer suggests that these connections consist of a single data chunk at the application layer. Moreover, $\eta_{UL} = 49\%$, revealing that connections belong to a symmetric service, i.e., users send and receive data. All these characteristics can be associated to instant messaging services (e.g., WhatsApp) [42].

Group 3 has less connections than group 2 (23% of the total) with longer duration (≈8 s without considering the inactivity timer) and a higher but still limited volume (≈11 kB). The fact that data is transmitted in last TTIs and the extremely low activity ratio in the DL (1%) show that data consists of small data chunks scattered in time (in fact, $N_{DL}^{burst} = 54$). $\eta_{UL} = 35\%$, showing that a considerable amount of the total data is transmitted in the UL. Thus, these connections are likely due to several interactions between user and network. This behavior is also typical of instant messaging services, where several messages are received/sent before the inactivity timer expires, so all those messages are part of the same connection. Note that connections in groups 2 and 3 make up 56% of samples in the dataset, which is consistent with the fact that instant messaging services are the most demanded services in mobile networks nowadays [55].

Connections in group 4 are shorter than those of group 3 ($T_{RRC} = 12.3$ s), with similar DL volume (8 kB) but lower UL volume ratio ($\approx$25%). The average burst volume is much higher than in group 3 ($\overline{V_{DL}^{burst}} = 199$ B and 496 B in groups 3 and 4, respectively), showing an increase of data chunk length. In fact, only 40% of data is transmitted in last TTIs. This group may be associated with small data files (e.g., images, audio recordings, documents, etc.) commonly shared by e-mail, messaging applications or social networks.

HV-LC block, comprising data-hungry services with large data chunks at the application layer (i.e., η_{DL}<3%), is split in groups 5 to 7. Group 5 presents the lowest number of connections in the dataset (0.05% of the total) with the longest length ($T_{RRC}\approx$62 s) and the highest DL data volume (12 MB), which is transmitted in many bursts. In fact, despite the reduced number of connections, this group accounts for 41.81% of the total download traffic in the network. The large duration and DL volume and the presence of bursty traffic suggest that this group includes connections from audio and video streaming applications (e.g., YouTube, Netflix, Spotify, etc.). It is worth noting that the median value of $TH_{session}^{DL}$ in this group in higher than expected, since 2150 kbps is approximately the rate of high-definition video [56]. It should be pointed out that, at the initial phase of a streaming session, a significant part of the video/audio file (e.g., 40 s) is downloaded at full speed to avoid re-buffering events. Then, download speed decreases, approaching the playout rate [40], [41]. Thus, $TH_{session}^{DL}$ for short videos can be considerably higher than the playout rate. A deeper analysis of data (not shown here) reveals that $TH_{session}^{DL}$ for connections in this group tends to decrease as T_{RRC} increases, which is consistent with the fact that, in longer videos, download speed tends to playout rate.

Groups 6 and 7 in HV-LC block comprise shorter connections than group 5 ($T_{RRC}\approx$20 s) with lower V_{DL} ($\approx$2 MB). The new burst indicators reveal that, in connections in group 6, data is transmitted in a few very long bursts over the air interface (the heaviest in the dataset). As a consequence, the activity ratio in the DL and session throughput are the highest (10.4% and 2.3 Mbps, respectively). These features fit with full buffer services, such as app download, software update or large file download via FTP, where the user demands as many resources as possible until all the data is transmitted. In contrast, group 7 comprises connections with a large number of bursts ($N_{DL}^{burst} = 136$ in group 7, compared to 24 in group 6) and lower DL activity ratio (6.7%) and session throughput ($\approx$1.48 Mbps). The higher ratio of last TTIs (0.17 in group 7, compared to 0.03 in group 6) points out the presence of small data bursts, which is confirmed by the lower $\overline{V_{DL}^{burst}}$ (6.5 kB in group 7, compared to 16.3 kB in group 6). Because of the presence of bursts with different sizes, and the median value of V_{DL}, very similar to the median size of mobile web pages in Alexa ranking, this group is labeled as web browsing.

Finally, connections in HV-SC block make up group 8. Since $\eta_{DL}^{lastTTI} = 0.23$, it is deduced that connections in this group have medium size data chunks. The median value of T_{RRC} is 46 s. The reduced DL activity ratio (2.3%) and the low session throughput ($\approx$250 kbps) point out that such a duration is due to several user interactions. This group may contain a mix of services, such as web browsing (e.g., web with many small objects, or multi-page sessions) or social networks, where a wide range of services (e.g., instant messaging, file sharing, short video streaming, etc.) can be demanded in a single connection.

TABLE 4. Share of DL traffic volume.

Service	Vendor report	Proposed method
Streaming	54.6 %	41.8 %
Web browsing	6 %	8.3 %
Full buffer services	6.9 %	9.2 %
Social networks & others	32.5 %	40.7 %

In the absence of labeled data, the classification shown in Table 3 is validated by comparing the results with mobile traffic statistics published by a vendor [53]. Table 4 shows the percentage of traffic per application type carried worldwide in 2016 [53] (i.e., when traces were collected) and that obtained by E–AHC. According to [53], audio/video streaming services carry most of the traffic (54.6%) in current networks. This figure is consistent with results from E–AHC, which ascribe 41.8% of traffic to these services (group 5). In [53], 6% of traffic is assigned to web browsing, whereas the proposed classification system assigns 8.3% of traffic to this service (group 7). Software update, application download and file sharing services comprise 6.9% of traffic in [53], compared to the 9.2% of traffic assigned to full buffer services (group 6) by E–AHC. Finally, [53] includes two groups called *Social Networks* and *Others* carrying 32.5% of traffic. Both groups include traffic of a different nature (e.g., instant messaging, short videos, small file sharing, etc.), equivalent to groups 1,2,3, 4 and 8 in A-EHC, carrying 40.7% of volume in the DL. Nonetheless, note that the classification performed here is based on traces from a particular network, and percentages may slightly differ from those reported worldwide by the vendor.

VI. CONCLUSION

In this work, a novel scheme for coarse-grained encrypted traffic classification in mobile networks has been proposed. Unlike previous flow-based approaches, based on expensive traffic probes in the core network, classification is based on traffic descriptors computed from connection traces collected on the air interface. To avoid the influence of network conditions, a new set of network-independent indicators describing typical application burst behavior per connection has been developed. The model is based on unsupervised learning, namely agglomerative hierarchical clustering, so that it can be applied in absence of labeled data.

Validation has been performed with a dataset from a live LTE network. Results have shown the limitations of classical clustering algorithms due to the uneven demand of services in mobile networks, where push notifications and instant messaging prevail over other services. To circumvent this problem, it is essential to exploit a priori knowledge before applying unsupervised clustering for traffic classification. The classification performed by the proposed method is consistent with the traffic share reported for current live networks, showing that traffic classification can be performed without installing expensive probes in the core network.

With the proposed method, radio optimization teams can make the most of existing trace datasets to build application performance maps for network benchmarking purposes. Since this task is carried out offline, computational efficiency is not critical. Moreover, the methodology can easily be extended to other radio access technologies, and is especially suitable for future 5G systems, where little knowledge is available about the type of services to come.

ACKNOWLEDGMENT

The authors would like to thank Juan Luis Bejarano Luque and Nuria González- Serrato (University of Málaga) for their valuable contribution to the experiments carried out in this work.

REFERENCES

[1] P. Borasi and S. Baul, “Mobile application market by marketplace and app category: Global opportunity analysis and industry forecast,” Allied Market Res., Pune, India, White Paper A01969, 2019.

[2] P. Le Callet, S. Möller, and A. Perkis, “Qualinet white paper on definitions of quality of experience. technical report version 1.2,” Eur. Netw. Qual. Exper. Multimedia Syst. Services (COST Action IC), White Paper, Mar. 2013.

[3] E. Liotou, D. Tsolkas, N. Passas, and L. Merakos, “Quality of experience management in mobile cellular networks: Key issues and design challenges,” *IEEE Commun. Mag.*, vol. 53, no. 7, pp. 145–153, Jul. 2015.

[4] A. J. Garcia, M. Toril, P. Oliver, S. Luna-Ramirez, and R. Garcia, “Big data analytics for automated QoE management in mobile networks,” *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 91–97, Aug. 2019.

[5] *Policy and Charging Control Architecture*, document TS 23.203, 3GPPP, Version 15.4.0, 2018.

[6] A. Nakao and P. Du, “Toward in-network deep machine learning for identifying mobile applications and enabling application specific network slicing,” *IEICE Trans. Commun.*, vol. E101.B, no. 7, pp. 1536–1543, 2018.

[7] *Internet assigned numbers authority (IANA)*. Accessed: Jul. 2, 2020. [Online]. Available: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

[8] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, “Transport layer identification of P2P traffic,” in *Proc. 4th ACM SIGCOMM Conf. Internet Meas. (IMC)*, 2004, pp. 121–134.

[9] T. Bujlow, V. Carela-Español, and P. Barlet-Ros, “Independent comparison of popular DPI tools for traffic classification,” *Comput. Netw.*, vol. 76, pp. 75–89, Jan. 2015.

[10] M. Finsterbusch, C. Richter, E. Rocha, J.-A. Muller, and K. Hanssgen, “A survey of payload-based traffic classification approaches,” *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1135–1156, 2nd Quart., 2014.

[11] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, “AppScanner: Automatic fingerprinting of smartphone apps from encrypted network traffic,” in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Mar. 2016, pp. 439–454.

[12] 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 *Proc. 10th USENIX Workshop Offensive Technol. (WOOT)*, 2016, pp. 1–10.

[13] J. Erman, M. Arlitt, and A. Mahanti, “Traffic classification using clustering algorithms,” in *Proc. SIGCOMM Workshop Mining Netw. Data (MineNet)*, 2006, pp. 281–286.

[14] 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.

[15] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, “Traffic classification on the fly,” *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 2, pp. 23–26, 2006.

[16] Y. Liu, J. Chen, P. Chang, and X. Yun, “A novel algorithm for encrypted traffic classification based on sliding window of flow’s first n packets,” in *Proc. 2nd IEEE Int. Conf. Comput. Intell. Appl. (ICCIA)*, Sep. 2017, pp. 463–470.

[17] T. Stöber, M. Frank, J. Schmitt, and I. Martinovic, “Who do you sync you are?: Smartphone fingerprinting via application behaviour,” in *Proc. 6th ACM Conf. Secur. Privacy Wireless Mobile Netw. (WiSec)*, 2013, pp. 7–12.

[18] I.-C. Hsieh, L.-P. Tung, and B.-S.-P. Lin, “On the classification of mobile broadband applications,” in *Proc. IEEE 21st Int. Workshop Comput. Aided Model. Design Commun. Links Netw. (CAMAD)*, Oct. 2016, pp. 128–134.

[19] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, “Robust smartphone app identification via encrypted network traffic analysis,” *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 63–78, Jan. 2018.

[20] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, “Multi-classification approaches for classifying mobile app traffic,” *J. Netw. Comput. Appl.*, vol. 103, pp. 131–145, Feb. 2018.

[21] D. Li, Y. Zhu, and W. Lin, “Traffic identification of mobile apps based on variational autoencoder network,” in *Proc. 13th Int. Conf. Comput. Intell. Secur. (CIS)*, Dec. 2017, pp. 287–291.

[22] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescape, “Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges,” *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 2, pp. 445–458, Jun. 2019.

[23] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapè, “MIMETIC: Mobile encrypted traffic classification using multimodal deep learning,” *Comput. Netw.*, vol. 165, Dec. 2019, Art. no. 106944.

[24] I. de-la Bandera, M. Toril, S. Luna-Ramírez, V. Buenestado, and J. M. Ruiz-Avilés, “Complex event processing for self-optimizing cellular networks,” *Sensors*, vol. 20, no. 7, p. 1937, 2020.

[25] M. Toril, R. Acedo-Hernández, A. Sánchez, S. Luna-Ramírez, and C. Úbeda, “Estimating spectral efficiency curves from connection traces in a live LTE network,” *Mobile Inf. Syst.*, vol. 2017, pp. 1–11, Jan. 2017.

[26] A. Sánchez, R. Acedo-Hernández, M. Toril, S. Luna-Ramírez, and C. Úbeda, “A trace data-based approach for an accurate estimation of precise utilization maps in LTE,” *Mobile Inf. Syst.*, vol. 2017, pp. 1–10, Jan. 2017.

[27] A. Duran, M. Toril, F. Ruiz, and A. Mendo, “Self-optimization algorithm for outer loop link adaptation in LTE,” *IEEE Commun. Lett.*, vol. 19, no. 11, pp. 2005–2008, Nov. 2015.

[28] V. Buenestado, M. Toril, S. Luna-Ramírez, J. M. Ruiz-Avilés, and A. Mendo, “Self-tuning of remote electrical tilts based on call traces for coverage and capacity optimization in LTE,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 4315–4326, May 2017.

[29] C. Gijon, M. Toril, S. Luna-Ramirez, and M. L. Mari-Altozano, “A data-driven traffic steering algorithm for optimizing user experience in multi-tier LTE networks,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 9414–9424, Oct. 2019.

[30] A. Gomez-Andrades, R. Barco, I. Serrano, P. Delgado, P. Caro-Oliver, and P. Munoz, “Automatic root cause analysis based on traces for LTE self-organizing networks,” *IEEE Wireless Commun.*, vol. 23, no. 3, pp. 20–28, Jun. 2016.

[31] *Subscriber and Equipment Trace: Trace Data Definition and Management*, document TS 32.423, 3GPPP, Version 15.0.0, 2018.

[32] *Subscriber and Equipment Trace: Trace Concepts and Requirements*, document TS 32.421, 3GPPP, Version 15.0.0, 2018.

[33] J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z. M. Mao, S. Sen, and O. Spatscheck, “An in-depth study of LTE: Effect of network protocol and application behavior on performance,” in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 363–374, 2013.

[34] V. Buenestado, J. M. Ruiz-Aviles, M. Toril, S. Luna-Ramírez, and A. Mendo, “Analysis of throughput performance statistics for benchmarking LTE networks,” *IEEE Commun. Lett.*, vol. 18, no. 9, pp. 1607–1610, Sep. 2014.

[35] K. R. Fall and W. R. Stevens, *TCP/IP Illustrated: The Protocols*, vol. 1. Reading, MA, USA: Addison-Wesley, 1994.

[36] S. Sesia, M. Baker, and I. Toufik, *LTE—UMTS Long Term Evolution: From Theory to Practice*. Hoboken, NJ, USA: Wiley, 2011.

[37] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proc. 14th Int. Conf. Mach. Learn. (ICML)*, 1997, pp. 179–186.

[38] A. Alexa. *The Top 500 Sites on the Web*. Accessed: Jun. 30, 2020. [Online]. Available: https://www.alexa.com/topsites

[39] *WebPageTest Tool*. Accessed: Jun. 30, 2020. [Online]. Available: https://www.webpagetest.org/

[40] A. Schwind, F. Wamser, T. Gensler, P. Tran-Gia, M. Seufert, and P. Casas, "Streaming characteristics of spotify sessions," in *Proc. 10th Int. Conf. Qual. Multimedia Exper. (QoMEX)*, May 2018, pp. 1–6.

[41] P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, and J. M. Lopez-Soler, "Analysis and modelling of YouTube traffic," *Trans. Emerg. Telecommun. Technol.*, vol. 23, no. 4, pp. 360–377, Jun. 2012.

[42] S. E. Coull and K. P. Dyer, "Traffic analysis of encrypted messaging services: Apple iMessage and beyond," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 5–11, Oct. 2014.

[43] U. Acer, A. Mashhadi, C. Forlivesi, and F. Kawsar, "Energy efficient scheduling for mobile push notifications," in *Proc. 12th EAI Int. Conf. Mobile Ubiquitous Syst., Comput., Netw. Services*, 2015, pp. 100–109.

[44] L. Rokach and O. Maimon, "Clustering methods," in *Data Mining and Knowledge Discovery Handbook*. Boston, MA, USA: Springer, 2005, pp. 321–352.

[45] H. P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Trans. Knowl. Discovery Data*, vol. 3, no. 1, pp. 1–58, Mar. 2009.

[46] M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of clustering high dimensional data," in *New Directions in Statistical Physics*. Berlin, Germany: Springer, 2004, pp. 273–309.

[47] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.

[48] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

[49] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Statist.-Theory Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[50] Google. *HTTPS Encryption on the Web*. Accessed: Jun. 10, 2020. [Online]. Available: https://transparencyreport.google.com/https/overview

[51] L. Hubert, H.-F. Köhn, and D. Steinley, "Cluster analysis: A toolbox for MATLAB," in *The SAGE Handbook of Quantitative Methods in Psychology*. London, U.K.: SAGE Publications Ltd., 2009, pp. 444–512.

[52] J. H. Ward, Jr., "Hierarchical grouping to optimize an objective function," *J. Amer. Stat. Assoc.*, vol. 58, no. 301, pp. 236–244, Mar. 1963.

[53] Ericcson. *Mobile Trafic By Application Type in 2018*. Accessed: Jul. 10, 2020. [Online]. Available: https://www.ericsson.com/TET/trafficView/loadBasicEditor.ericsson

[54] C. C. Aggarwal and C. K. Reddy, *Data Clustering: Algorithms and Applications*. Boca Raton, FL, USA: CRC Press, 2013.

[55] Y. Fu, H. Xiong, X. Lu, J. Yang, and C. Chen, "Service usage classification with encrypted Internet traffic in mobile messaging apps," *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2851–2864, Nov. 2016.

[56] YouTube. *Live Encoder Settings, Bitrates, and Resolutions*. Accessed: Jun. 15, 2020. [Online]. Available: https://support.google.com/youtube/answer/2853702?hl=en

CAROLINA GIJÓN received the B.Sc. degree in telecommunication systems engineering and the M.Sc. degree in telecommunication engineering from the University of Málaga, Spain, in 2016 and 2018, respectively, where she is currently pursuing the Ph.D. degree. Her research interests include self-organizing networks, machine learning, and radio resource management.

MATÍAS TORIL received the M.S. degree in telecommunication engineering and the Ph.D. degree from the University of Málaga, Spain, in 1995 and 2007, respectively. Since 1997, he has been a Lecturer with the Communications Engineering Department, University of Málaga, where he is currently a Full Professor. He has coauthored more than 130 publications in leading conferences and journals and holds eight patents owned by Nokia or Ericsson. His current research interests include self-organizing networks, radio resource management, and data analytics.

MARTA SOLERA received the M.Sc. and Ph.D. degrees in telecommunication engineering from the Univesitat Politècnica de Catalunya (UPC), in 1996 and 2006, respectively. She is currently an Associate Professor with the Department of Communication Engineering, University of Malaga (UMA). Since 1996, she has been lecturing in several universities such as UPC, Universidad Nacional Autonoma de Mexico (UNAM), and UMA. She has been involved in several public funded national research projects in the field of multimedia and mobile communications. Her research interests include design and performance evaluation of multimedia services over mobile networks.

SALVADOR LUNA-RAMÍREZ received the M.S. degree in telecommunication engineering and the Ph.D. degree from the University of Málaga, Spain, in 2000 and 2010, respectively. Since 2000, he has been with the Department of Communications Engineering, University of Málaga, where he is currently an Associate Professor. His research interests include self-optimization of mobile radio access networks and radio resource management.

LUIS ROBERTO JIMÉNEZ received the M.S. degree in electronics and communications engineering from the Santo Domingo Institute of Technology (INTEC), Santo Domingo, Dominican Republic, in 2013, and the M.S.E. degree in telematics and telecommunication networks from the University of Malaga, Málaga, Spain, in 2015, where he is currently pursuing the Ph.D. degree in telecommunications engineering. His current research interests include self-optimization networks and performance evaluation of multimedia services over mobile networks based on customer experience. He is a recipient of a Junta de Andalucía Scholarship (2017–2021) over methods planning and optimizing QoE in B4G networks.