Building up knowledge through passive WiFi probes

Alessandro E.C. Redondi, Matteo Cesana*

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano Milan, Italy

Inexpensive WiFi-capable hardware can be nowadays easily used to capture traffic from end users and extract knowledge. Such knowledge can be leveraged to support advanced services like user profiling, device classifi-cation. We review here the main building blocks to develop a system based on passive WiFi monitors, that is, cheap and viable sniffers which collect data from end devices even without an explicit association to any Wi-Fi network. We provide an overview of the services which can be enabled by such approach with three practical scenarios: user localization, user profiling and device classification. We evaluate the performance of each one of the three scenarios and highlight the challenges and threats for the aforementioned systems.

1. Introduction

Recent studies on the growth rate of wireless traffic have predicted that Wi-Fi traffic will account for more than half of total IP traffic by 2019, with the total public Wi-Fi hotspots growing sevenfold from 2015 to 2020, from 64.2 million in 2015 to 432.5 million by 2020 [1]. This means that a good deal of IP traffic generated by end users to post on social networks, interact with friends, get access to entertainment and other services will go through a wireless first-mile connection.

In this context, capturing and properly processing data from such networks does provide a goldmine to build up value-added services. As a matter of fact, WiFi Internet Service Providers and system integrators are already broadening their commercial offer beyond the simple provision of internet connectivity to include advanced services based on WiFi data analytics. In such a way, commercial WiFi deployments may be transformed into powerful tools for conducting market research and gauging insights from customers, as WiFi traffic can reveal information on first time vs. frequent visitors at shops, customer loyalty, dwell times, walking paths, real-time heat-maps, customer gender and age.

The aforementioned services are generally offered by leveraging either *active* or *passive* WiFi measurements. Active WiFi measurements capture and analyze the traffic of end users which are associated to the specific WiFi hot spots. Such measurements are generally very rich in terms of available information (uplink/downlink traffic exchanged, total connection time, etc.). Furthermore, they generally include the identity of the accessing user, since the vast majority of WiFi hot spots around the world require some type of authentication (e.g., through captive portals). Conversely, passive measurements occur when the data is collected from end user devices which are not associated to any WiFi hot spots. Clearly, such measurements are generally "less informative" than active ones since they are based only on WiFi management frames which are exchanged by WiFi devices regardless their association status (e.g., association request/response, probe request/response, etc.). Still, insightful information can be extracted from passive measurements with the clear advantage of being less intrusive (and less expensive) than active approaches.

In this work, we showcase the potentials of leveraging passive wifi measurements to extract value-added knowledge. Namely, we focus here on the analysis of WiFi *probe request* management frames, which are broadcast by end devices to probe for available WiFi hot spots. Starting from the availability of millions of probe request frames, we provide three different contributions: (i) we propose a thorough analysis of localization-based services built on top of probe request frames; (ii) we propose a method to identify groups of people having similar behaviors in the way they visit a particular area and (iii) we show how to leverage the information contained into probe request frames to automatically detect if the sending device is a smartphone or a laptop, an information that can be used to optimize the network configuration and/or implement services such as management of wide WiFi network or smart content caching approaches.

Referring to the first contribution, only very recently some attention has been given to the problem of exploiting probe request frames to localize users in a passive way. The majority of the works in this area focus on creating location-based heat maps or to track mobile users in a coarse way, rather than focusing on fine-grained localization. Differently, we try to push the localization accuracy of systems based on probe request frames to its limit and we evaluate two localization techniques based on probe request: in the first one, we resort to

Received 7 March 2017;
Received in revised form 23 October 2017;
Accepted 18 December 2017
Available online 19 December 2017

* Corresponding author.
E-mail addresses: alessandroenrico.redondi@polimi.it (A.E.C. Redondi), matteo.cesana@polimi.it, cesana@elet.polimi.it (M. Cesana).

Published Journal Article available at https://doi.org/10.1016/j.comcom.2017.12.012

parametric model-based triangulation approaches, whilst in the second one we use fingerprinting.

For the second contribution, we propose a set of features derived from the analysis of probe requests capture time that are later used to cluster the users in different groups. We show that with our approach groups of users with very different behaviours can be highlighted and separated. Moreover, we show that probe request messages can be used to infer the geographical features of users (provenience and attitude to travel).

As for smartphone/laptop classification, we show that it can be be performed by collecting (and parsing) only probe requests Wi-Fi management frames, in contrast with those systems that resort to invasive deep packet inspection techniques to read out application layer information in the exchanged packets. Our proposed classification framework first characterizes each device with a set of *features* extracted from the probe request frames; the reference set of feature captures information on the temporal process of probe request transmission (how frequently probe requests are transmitted) and the power levels used in the probe request transmission. Then, a supervised learning approach is used to train different classifiers able to predict the type of transmitting device just by looking at its corresponding features.

The manuscript is organized as follows: Section 2 provides a survey on the reference literature exploiting passive measurements within WiFi networks; Section 3 provides a quick background on WiFi active scanning procedures and further describes the reference system set up used to collect and analyzed the passive WiFi traces; in Section 4 we show how to perform localization, user profiling and smartphone/laptop classification based on passive WiFi measurements. Section 5 concludes the work.

2. Background and related work

The IEEE 802.11 standard defines three types of layer-2 frames which are exchanged among WiFi devices: *control frames, management frames*, and *data frames* [2]. Passive measurement systems generally leverage *management frames* which are exchanged by Wi-Fi enable devices. Note that such devices do not need to be associated to any WiFi access point in order to exchange management frames.

More specifically, we are interested in the management frames transmitted by end devices during the *Active Scanning* phase, that is, the phase in which they search for WiFi networks (access points) in the surroundings to connect to. In such phase, each end device broadcasts a *probe request* management frame to stimulate in-range APs to manifest themselves (replying with a probe reply management frame). Such probe requests are usually broadcasted in sequence on all the available WiFi channels (1–14). The set of information which is contained in (or can be easily extracted by) probe request frames include the Medium Access Control (MAC) address of the sending device, the Received Signal Strength Indicator (RSSI) out of the transmission of the frame and the Preferred Network List (PNL), that is a list of Service Set IDentifiers (SSID) of the WIFi networks which are already known by the sending device. Fig. 1 reports the standard format of probe request frames

Recent studies have shown that properly collecting, processing and possibly coupling such basic information with other external data allows for building up build up some knowledge on the reference population/scenario the probe requests are collected from. The proposed system to accomplish this process are generally composed of three main elements: (i) a collection front-end based on WiFi-enabled hardware to collect probe requests frames; such hardware can be composed of either commercial Access Points or by "home-made" solutions based on low-power embedded devices; (ii) a data processing engine which operates on the collected data to extract the target context knowledge, and optionally (iii) an external service providing side-information on the reference scenario which can be coupled in the data processing phase with the information extracted from the field.

Table 1
Knowledge extracted from passive WiFi sensor systems.

Context	References	
	Indoor	Outdoor
Localization/Tracking/Density	[3–8]	[9–12]
De-Anonymization	[13–17]	[13–17]
Users/Device Profiling	[16–25]	[16–17] [20–25]

The available systems based on passive WiFi probes can be classified according to the specific target information/knowledge which is extracted. Generally speaking, three broad classes can be identified in this respect: (i) systems targeting localization and tracking of end users; (ii) systems willing to associate a specific identity (real or cyber) to any given captured device; and (iii) systems targeting end user profiling with respect to technical and social parameters. Table 1 reports a classification of the reference literature with respect to the aforementioned guidelines, further distinguishing among contributions targeting indoor and outdoor environments.

2.1. Localization and tracking

Systems of this type generally exploit the information on the RSSI and the proximity to WiFi sensors to infer the geographical position of end users. The work in [3] targets pedestrian flow estimation across the security check in an airport. Several non-supervised learning approaches are proposed and qualitatively compared against a proxy measure for the flows and density, that is, the number of boarding pass scans performed at the security check at given time intervals.

Along the same lines, Fukuzaki et al. propose in [4] a pedestrian flow estimation service in shopping malls; the proposed system first estimates the position of a given device by leveraging the Received Signal Strength Indicator (RSSI) out of a probe request message captured by multiple probes, and then builds up a statistical characterization of the users flows. The proposed system is also used to count the number of people in the reference environment, namely, the authors use a simple linear model which returns the estimated number of people out of the total number of perceived MAC addresses; the model is trained against a secondary people counting system based on motion detection sensors at the entrances of the shopping mall.

Very recently, there have been some works targeting the problem of passive indoor localization using setups similar to the one proposed in this paper. In [5], a system composed of eight WiFi sniffers is deployed in an area of about 5000 m^2. A triangulation-based algorithm is used to localize coarsely the users in eight areas of the experimental area, however no details on the performance of the localization algorithm are reported. In [6], twelve WiFi sniffers are deployed in an area of about 340 m^2. Fingerprint localization through k-Nearest Neighbour classifier is performed, with a reported median error of about 4.5 m. The work in [7] presents *Probr*, an open source software solution to capture and process probe requests in order to support several on-line analysis tasks,

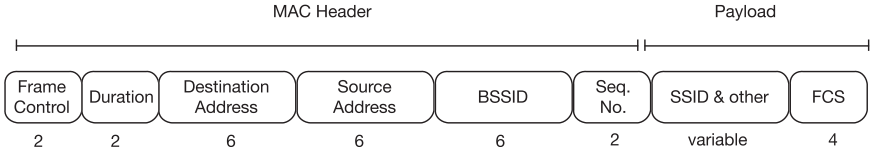


Fig. 1. Probe request frame format. Numbers represent the field size in bytes.

including localization, room utilization and people tracking. Localization is implemented with triangulation through a RSSI-based multilateration algorithm, but no results on its performance are reported. Finally, in [8] a system based on probe requests sniffing and fingerprint localization with Bayesian Classifier is presented. The system is tested in an indoor area of 30 m^2 with 4 sniffers and is reported to achieve a median localization error of about 2 m. Compared to these works, we provide a more detailed performance evaluation of both triangulation and fingerprinting-based localization on either a controlled or a realistic scenario.

Rouveyrol et al. show how to leverage commercial WiFi routers to track end users; the authors collect the results WiFi scanning procedures performed by a single smartphone, which, coupled with the smartphone positioning information, are used to showcase the feasibility of a large-scale WiFi network to track end users.

Trajectory estimation and crowd-control are also addressed in [10] and [12] with similar approaches. In [10], the authors propose an Hidden Markov Model which returns the most probable device trajectory receiving as input the WiFi detections of end devices. Bonne et al. focus on mobility characterization at mass events.

Differently from the previous works where the WiFi sensor are static, Chon et al. show in [9] how to use portable devices to collect insightful information from the surrounding environment; the reference systems is composed of 25 users with smartphones and portable devices lingering around Seoul and capturing WiFi traffic; the authors then show how to use such information to design application in the field of users mobility, tracking and co-location detection (two users in the same space). In [26], a single Unmanned Aerial Vehicle (UAV) is equipped with a sniffer device and used to collect probe request frames from different positions. A classification baed localization algorithm is then applied to estimate the position of a user, with a reported accuracy of 7.5 m in 80% of the cases.

2.2. De-Anonymization

The works in this class aim at associating each device to a specific end user, either a physical person or a cyber-physical avatar. In [14], WiFi passive probes are used to showcase the vulnerability of current WiFi procedures with respect to the *stalker attack* attack where a stalker physically follows the person she wants to associate to a MAC address and keeps collecting probe request frames from the environment.

In [15], the authors propose a system that exploits a limitation of Wi-Fi-based Positioning Systems (WPSs) in order to spoof the geolocation of the user. This spoofed geolocation is then used as a side-channel information source to establish the link between a Wi-Fi MAC address and the user's profile on a geotagged service like FourSquare, Facebook and others.

In [13] a system of cameras is used together with passive WiFi probes to de-anonymize MAC address. Namely, the authors show how positioning information extracted from the RSSI of received probe request messages can be coupled with positioning information extracted from a system of cameras, such that a person which is captured on an image can be mapped to a MAC address which is collected through probe requests.

The authors of [17] propose a de-anonymizing system for probe request messages. The proposed solution is able to infer the geographical provenience of the users generating probe request messages by looking at the BSSID advertised in broadcast probe requests and querying an external geo-localization service for WiFi hotspots (Wigle[1]).

Chernyshev et al. [16] look at information that can be extracted from the SSID strings contained in probe request messages. The assumption is that the entities of potential interest, such as locations and personal names contained within SSIDs, can be recognized in an automated fashion; the authors show that the attributes which can be extracted from the SSIDs can be used as a basis for inference attacks.

2.3. Users/device profiling

The work belonging to this class generally aims at extracting features of the device generating the probe request and/or of the owners of such devices. Cunche et al. study in [22] and later in [23] the possibility of extracting social relationships between individuals by analyzing the similarity of their probe request packets. The authors first define a set of features related to the probe request messages which constitute a *fingerprint*, and then measure the similarity of fingerprint couples. Different similarity measures and feature compositions are considered. Along the same lines, the authors of [25] show how SSIDs captured in probe request packets can be used to infer known business venues.

Cheng et al. show in [21] how to detect relationships among users by parsing the PNL inside probe request messages and resorting to Wigle for geo-localizing APs.

After performing a collection campaign of probe requests lasted about 3 months and targeting events of International, National, and Citywide relevance, the authors of [24] focus on discovering social-related phenomena like: the distribution of languages of the people participating in the events, the vendors of the devices they use, and, based on the cost of the different brands, give insights on the wealth of the population.

In [18], a non-intrusive Wi-Fi sniffing system is proposed to profile the users in office environments. The authors assign to each MAC a set of features extracted from the generated WiFi traffic; such features include the dwell time of a given MAC in the system and the probing frequency. A non-supervised clustering technique is then applied to find patterns in the behavior of different individuals.

A similar approach is carried out in [19] which focuses on profiling end users in university campuses. The proposed system leverages a supervised learning approach which is able to classify users per usage groups (short stay, long stay, etc.)

Different than the previous work which targets the profiling of end users, the authors of [20] focus on the profiling of WiFi devices. The proposed study analyzes several packet capture configurations (e.g., antennas and network interfaces) and characterizes the behavior of different devices (hardware/software configuration) with respect to the emission process of probe requests.

3. System set up and data preprocessing

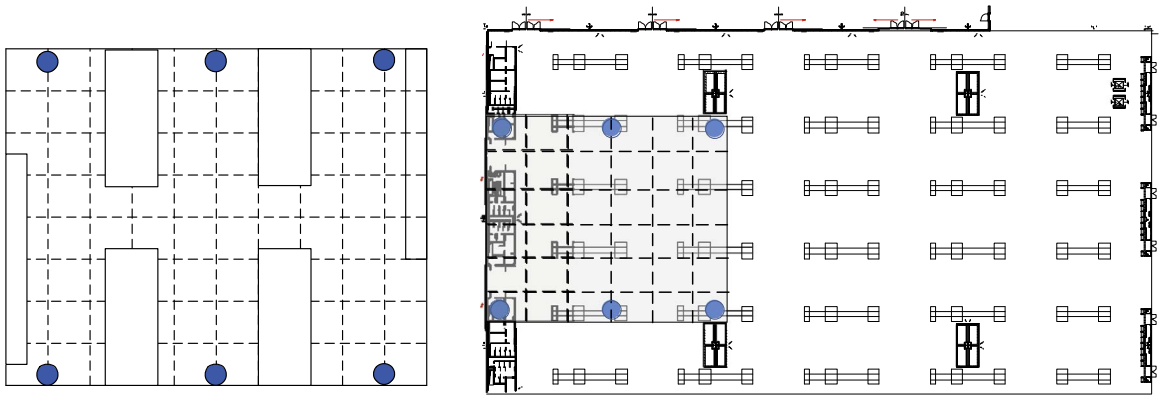
3.1. System set up

The data collection system to capture over the air probe request messages consists of Raspberry Pi 3 nodes running Jessy OS and further equipped with a TP-Link TL-WN725N WiFi dongle tuned to WiFi channel number 1; Each node runs a *t-shark* (the terminal version of WireShark) script to capture only probe request frames. The information which is collected by the probing system is polished (by eliminating corrupted packet captures and eliding all the packet fields which are useless) and finally stored to a MySQL database for further analysis.

Each probe request message that is retained for database insertion includes the following fields:

- the source MAC address (*src*),
- the Organizationally Unique Identifier - OUI which identifies the radio chip *vendor*;
- the Service Set Identifier (SSID) of the probe request which can be either "Broadcast" or a string containing the SSID of a Wi-Fi network known to the device.
- the *id* of the Raspberry receiving the probe request;

[1] https://wigle.net



(a) (b)

Fig. 2. Reference environments for (a) *Lablife* dataset and (b) *FieraMilano* dataset. For *Lablife*, the 80 m^2 area is divided into 80 1m × 1m cells. For *FieraMilano*, the selected area is divided in 36 6m × 6m cells. Solids represent furniture, desks and other obstacles, localization cells are reported in dashed lines and solid circles report the position of the six anchor nodes collecting probe requests.

- the received signal strength indicator (*rssi*) out of the received probe request;
- the timestamp of the reception event (*rs*);
- the sequence number (*sn*) of the probe request message.

The aforementioned probing hardware is used to build up three different datasets:

- *LabLife dataset*: six probing devices are deployed on the walls of our research lab which has a surface of 80 m^2 (see Fig. 2(a)); the six anchor nodes, which are time-synchronized through NTP protocol, store the traffic files locally and then send them to a central server periodically (every hour) leveraging wired Ethernet connectivity. This dataset includes about 11 millions entries referring to a collection period of four weeks, and it is used to perform localization and user profiling.
- *FieraMilano dataset*: the same setup of the *LabLife* dataset is deployed at FieraMilano, an area devoted to public exhibitions in the city of Milan. The experimental area consists of a 1296 m^2 indoor space and the six probing devices are arranged in the same configuration as the *LabLife* dataset ((see Fig. 2(b))). Wi-Fi probe requests from visitors of an exhibition were collected for five days from Monday to Friday. In total, more than 2 millions probes were collected.
- *Lectures dataset*: For what concerns device classification, a single probing devices is used to gather probe request frames generated during particular university classes (“tutorials” and “hands-on” lectures) where students have their own laptops and smartphones with them. At the beginning of the lecture, students are asked to (i) turn on the Wi-Fi interfaces of their devices and (ii) compile an anonymous form and insert the MAC addresses of their smartphones and laptops to serve as ground truth. In addition to those entries whose MAC addresses are labeled by students as belonging to either the “laptop” or “mobile”, we also add to the database all those probe request frames from device manufactured by a laptop-only or mobile-only producer. The manufacturer is identified from the first 3 octets of the MAC address (the so-called Organizationally Unique Identifier - OUI). In detail, probe request frames from Intel and Liteon devices are automatically marked as coming from laptops, while probe requests from Huawei, Nokia, Sony Mobile, Xiaomi and onePlus are labeled as “mobile”. In total, the database consists of more than 2×10^5 different probe request entries, spanning 10 different hours over 5 days and belonging to 279 different devices of known type.

3.2. Data pre-processing and lessons learned

a) *MAC randomization:* In response to the possibility of easily capturing probe request frames, most operating systems for smartphones and laptops have now implemented different variants of MAC address randomization in order to protect the privacy of the users. The incidence of such effect is considerable in the reference datasets: out of 231442 unique MAC addresses detected in the *LabLife* and *Lectures* dataset, only 90937 (approx 39%) of the MAC addresses has a valid, registered OUI. We deduct that the remaining MAC addresses, which do not have a valid OUI, are therefore randomized by the specific operative systems.

b) *Unicast probe request:* Unicast probe request messages are a minority: only 25% of probe request messages of the datasets do notify any SSID, and only 20% of the detected MAC addresses does advertise any SSID at all. Fig. 3 report the cumulative distribution of the number of SSIDs advertised by MAC addresses advertising at least one SSID. As clear from the figure, 60% of probe request which advertise at least one SSID does advertise only one SSID.

c) *Temporal analysis:* The probe request traffic stored in the *LabLife* dataset features the traditional periodic behavior of any telecommunication network traffic following classical daily and weekly periodicity. Fig. 4 shows the number of detected probes in a single day (Fig. 4a) and over a month (Fig. 4b) (note the drop in traffic due to the Italian national holiday on the 25th of April and the lack of data due to a power outage on May the first).

d) *Inter-probing time:* Other works in the literature observe that the

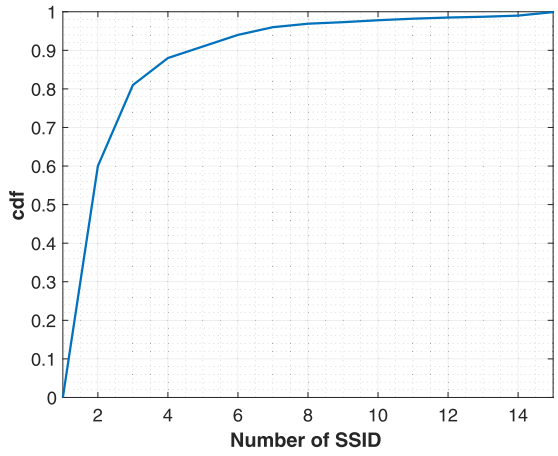
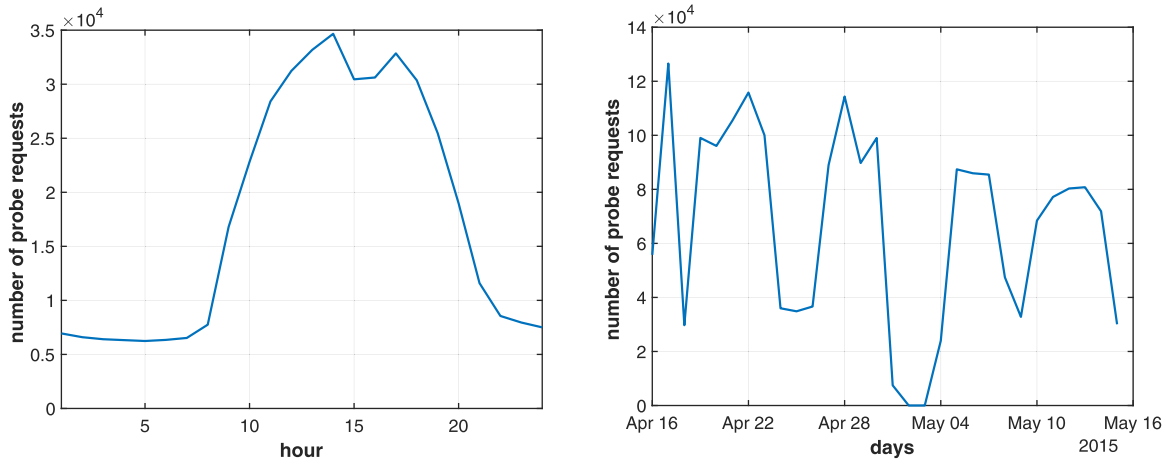


Fig. 3. Cumulative distribution function of the number of SSID by a specific MAC address.



(a) probe request per hour

(b) probe requests per day

Fig. 4. Number of detected probe requests in a reference day per hour (a) and in a reference month per day (b).

Table 2
Mean and standard deviation of the inter-probing time.

Device	Inter-probing Time [min]	
	Mean	Standard Deviation
Apple	4.71	4.83
Motorola	3.73	2.55
Samsung	4.25	1.63

average probe request emission frequency for idle terminals is in the range of one probe every 2–5 minutes. A similar behavior has been observed also in our datasets. Table 2 reports the mean and standard deviation of the emission period for three different devices (distinguished per vendor) in idle mode which have been monitored for a total period of 8 hours.

It is worth observing here that the specific characteristics of the probe request generation process are extremely device- and user-dependent. Each device (hardware/software combination) emits probe request according to its own internal rules and optimizations. We will show how this distinctive behaviour can be leveraged to classify the device type in Section 4.3.

4. Extracting information from passive traffic

We focus on three possible types of information which can be inferred by looking at probe request traffic; namely we report on the use and performance of probe-based localization systems (Section 4.1), probe-based users profiling algorithms (Section 4.2), and probe-based automatic classification of end devices (Section 4.3).

4.1. Device localization

We compare hereafter two well known localization approaches which leverage the received signal strength of wireless transmissions: *model-based parametric* localization and *fingerprinting* localization; the first, postulates the existence of a model function which relates the received power to the distance travelled by the electromagnetic signal; once the model is known/estimated, the terminal-anchor(s) distance can be calculated out of the received power, and the estimated position of the terminal is finally obtained through trilateration.

Differently, *fingerprinting* localization builds on the assumption that any position in the reference environment is characterized by a distinctive "fingerprint", defined as the vector of RSSIs values at the anchor nodes corresponding to a probe request transmission performed at that very position. The localization process is composed of two distinct phases: in the training phase, the reference environment is sampled at given positions and a database is populated with lines containing the sampled position (cell id), and the RSSIs values received at all the anchor nodes out of probe requests transmitted from the sampled position. Once the database is populated, localization is performed in the online phase: the fingerprint coming from an unknown position is matched to all the entries in the database to find the closest match, and hence the closest position. Different fingerprint matching algorithms can be used to perform the database search, from the simple nearest neighbour search to advanced machine learning algorithms such as decision tree and random forest.

For what concerns the *LabLife* dataset, the reference area is discretized in 1[m^2] cells, for a total of 80 cells. As for the *FieraMilano* dataset, the 1296 m^2 area is divided in a 6×6 grid with squared cells of 36m^2 For each cell, several probe request frames emitted by a WiFi-enabled smartphone are collected at the six capturing devices. Note that each probe request frame has a unique sequence number, making it possible to identify a single probe request frames (and the corresponding RSSI) on the different capturing devices. To stimulate the emission of probe request frames from the smartphone, we use the built-in Wi-Fi management app to continuously refresh the list of available networks. For each cell, one minute of probe request frames is captured and labeled with the center coordinates of the cell $\overline{x_i}$. Data from the six different capturing devices is then merged into a single dataset of 3200 entries for the *LabLife* dataset and 1400 entries for the *FieraMilano* (about 40 entries per cell), where each entry has the form ($\overline{x_i}$,RSSI$_1$, ..., RSSI$_6$) and RSSI$_j$ is the received signal strength at the j-th capturing device. Each labeled dataset is then split into a *training dataset*, composed of 80% of each database entries, and a *evaluation dataset*, composed of the remaining 20% entries.

4.1.1. Model-based localization

In *model-based parametric* localization, the following reference propagation model is used:

$$RSSI = p_0 - 10\alpha \log\left(\frac{d}{d_0}\right), \tag{1}$$

where RSSI is the received signal strength indicator, p_0 is the received power at a reference distance of d_0 from the transmitter, and α is the path loss index.

The model parameters are empirically estimated for the test environment through an exhaustive measurement campaign. Namely, for all the possible combinations of parameters p_0 and α, the position of the

reference user, that is the vector coordinates of the cell the user lives in, $\overline{x}$, is then estimated by solving the following regularized least-square problem.

$$min \sum_{i=1}^{N} \theta_i (\|\overline{x} - \overline{y_i}\|_2 - d_i)^2, \tag{2}$$

where $\overline{y_i}$ is the position vector for the i-th probe element, and θ_i is a normalization factor, defined as:

$$\theta_i = \frac{\frac{1}{d_i^\gamma}}{\sum_{i=1}^{N} \frac{1}{d_i^\gamma}}$$

The parameters of the propagation model defined in Eq. (1) and the normalization parameter γ are then set to P_0^{opt}, α^{opt} and γ_{opt} which minimize the average mean square localization error of Eq. (2) for all the positions in the *training dataset*.

The localization precision is then tested on the *evaluation dataset* by applying the propagation model Eq. (1) with the optimal parameters set and solving Eq. (2).

4.1.2. Fingerprint-based localization

As explained before, the set of possible locations to be returned as an estimate is discrete (the 80 or 36 cells of the reference area for the *LabLife* or *FieraMilano* dataset, respectively). Therefore, it is possible to tackle the localization problem as a classification problem, where each location corresponds to a different class. A classification algorithm takes as input the set of RSSIs from the unknown position and returns the most probable location (class) for that input. Three different classifier are compared in this work:

- k-Nearest Neighbour (kNN): when $k = 1$ this simple classification algorithm assigns as output the cell corresponding to the nearest fingerprint in the training dataset. When $k > 1$, the algorithms return the most occurring cell among the k nearest neighbour fingerprints. Euclidean distance is used to compute distance measurements between the test and training fingerprints.
- Naïve Bayes (NB): this algorithm assigns to the fingerprint s a probability value $P(T|s)$, computed using the Bayes Theorem and assuming that features are independent, that is:

$$P(T|\mathbf{s}) = \frac{P(\mathbf{s}|T)P(T)}{P(\mathbf{s})} = P(T) \prod_i P(s_i|T), \tag{3}$$

  where s_i denotes the i-th RSSI component of s and the denominator $P(s)$ can be ignored as it is the same for all classes. In the training phase, the Naïve Bayes classifier learns $P(s_i|T)$ by fitting Gaussian probability distributions to each individual feature. In the test phase, given a newly observed fingerprint s, the NB classifier returns the most probable class, that is the class T for which $P(T|\mathbf{s})$ is maximized.
- Random forest (RF): this is an ensemble algorithm that has been shown to perform very well in several machine learning and data analysis tasks. A random forest classifier constructs several decision trees at training time, and outputs as a prediction the mode of the classes predicted by the individual trees (majority voting). The individual trees are obtained selecting each time a random training sample in order to decrease model variance (i.e. overfitting) and a random subset of the input features to produce weakly correlated trees.

4.1.3. Comparing localization systems

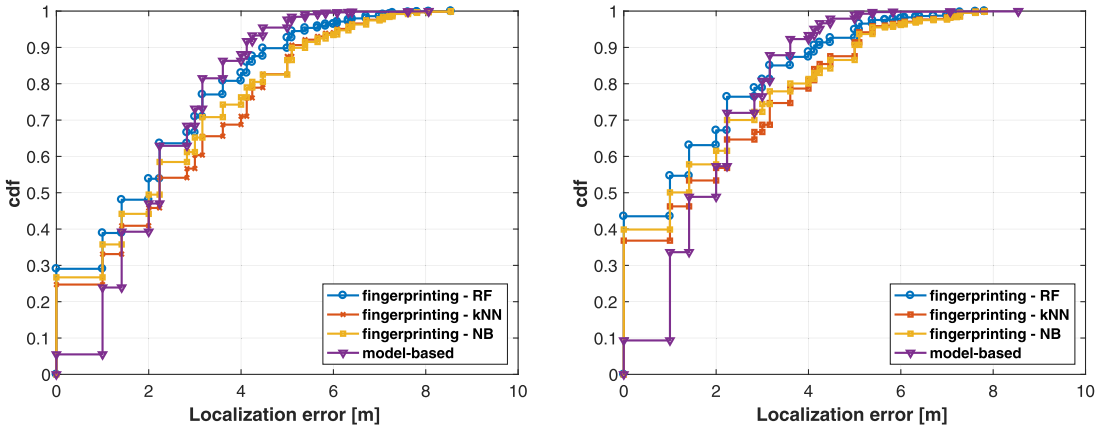
Figs. 5 and 6 report the cumulative distribution function of the localization error calculated on the *LabLife* and *FieraMilano* test dataset, respectively. Both figures report the performance obtained with (a) four anchor nodes and (b) six anchor nodes. In case of four anchors, only the data received from the capturing devices placed at the four corners of the reference environment (see e.g., Fig. 2) is used. Table 3 reports the mean localization errors for the different configurations tested. Moreover, we included in Table 3 the mean localization errors reported by the works in [6] and [8], both using fingerprinting localization, as a benchmark. Three main considerations can be made from the inspection of such results:

1) Overall, the accuracy of fingerprinting methods is higher than that achieved with model-based localization. This is visible comparing the value of the offset of the different CDFs (that is the percentage of times the system retrieved the correct cell in the grid). For the *LabLife* dataset with 4 anchor nodes, fingerprinting allows to retrieve the correct cell 30% of the times, while the accuracy of model-based localization is only 5%. When using 6 anchor nodes, the accuracy increases to 43% for fingerprinting and to 10% for model-based localization. The same consideration can be made when considering the mean localization errors shown in Table 3. When considering the 80th percentile, however, the two localization methods have similar performance.
2) Among the different fingerprint algorithms tested, the Random Forest algorithm always achieves the best results. This is clearly visible from the inspection of the CDFs, but also considering the results in Table 3. In particular, when comparing our results for the *LabLife* dataset with the ones reported in [6] or [8], the Random Forest algorithm allows to obtain a comparable or smaller error even with a smaller density of probing devices.
3) Overall, regardless of the localization algorithm used, the localization error depends on how many probing devices are installed in the reference area. Considering that the hardware setup used in this work is extremely simple and cheap (less than 50 USD per probing device), accurate passive localization is practically feasible in many scenarios.

4.2. User profiling

Besides being used for localization, probe requests can be leveraged to profile the users within the reference environment. We start from the observation that the population mix of the area under analysis is composed of different kinds of people. For what concerns the *LabLife* dataset, the research lab is visited by faculty, PhD and master students, visitors and passers-by. As for the *FieraMilano* dataset, the area is chracterized by the presence of visitors of the exhibition, staff and security personnel. In addition, visitors had the possibility of purchasing different tickets (1-day, 3-days and 5-days) for accessing the exhibition All these classes of users are in general different by return and sojourn time in the areas under analysis, which, in turn, can be captured by a specific "footprint" in the probe request emission. In particular, the return time of a user is captured using the *inter-probe period distribution* as basic feature. The construction of such a feature follows these steps:

- All timestamps t_i of the N_s captured probe request frames belonging to a single MAC address are sorted in chronological increasing order in an array $\mathbf{T} = [t_1, t_2, ..., t_{N_s}]$.
- The differential vector $\mathbf{P} = [t_2 - t_1, t_3 - t_2, ..., t_{N_s} - t_{N_s-1}]$ is constructed and thresholded so that all differences lower than 15 minutes are removed. This enables to capture only those time differences caused by an actual return of the user, filtering out those entries caused by the natural behaviour of the probe request emission process controlled by the operating system of the specific device.
- A histogram is created grouping all time differences in 5 different bins, whose limits are created empirically considering the natural behavior of a the user under consideration. For the *LabLife* dataset, the first bin captures all returns between 15 minutes and 3 hours, which correspond to users leaving from and coming back to the lab for a short time (e.g., lunch break). The second bin includes those



(a) Four anchor nodes (b) Six anchor nodes

Fig. 5. Cumulative distribution function of the localization error for the *LabLife* dataset when using four (a) and six (b) anchor nodes.

returns between 10 and 19 hours, which correspond to the daily return of users during the working week. The third bin contains time differences between 34 and 50 hours, which corresponds to a day-off from work. The fourth bin captures returns between 58 and 65 hours, which roughly correspond to users coming back after a 2-day break (weekend). Finally, the last bin captures all those returns greater than 65 hours. As for the *FieraMilano* dataset, considering that data acquisition was performed for five days (from Monday to Friday) we removed the last bin of the histogram.

- The resulting histogram is then normalized to unity and thresholded so that all entries below 0.01 are set to zero and the remaining one are set to 1. For the *FieraMilano* dataset, five additional binary features b_i are added, where $b_i = 1$ if the MAC address is present in the database for i days and 0 otherwise. The resulting 5-dimensional vector (for the *LabLife* dataset) or 9-dimensional binary vectors (for the *FieraMilano* dataset), are used as fingerprint for characterizing a user. The entire process is summarized in Fig. 8 for the *LabLife* dataset.

4.2.1. *Clustering the LabLife dataset*

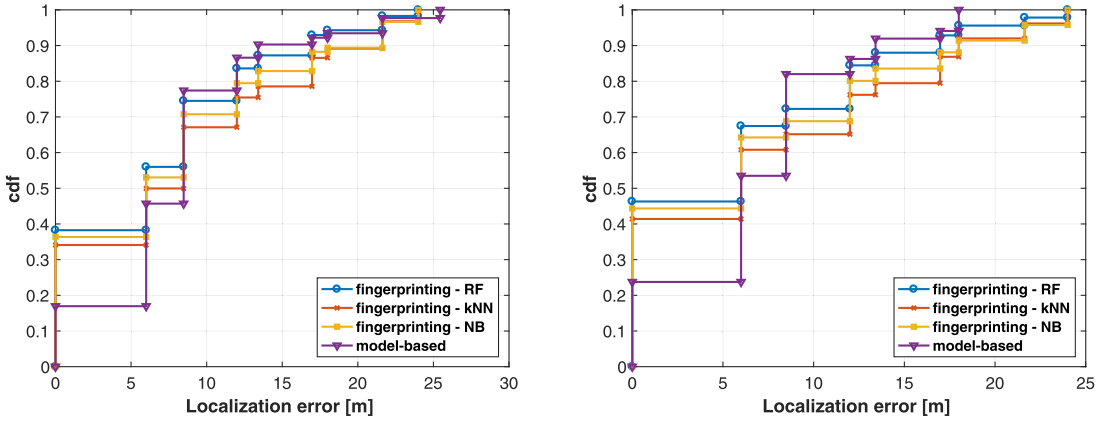
It is reasonable to expect different inter-probe period distributions for different users of the lab. As an example, the binary feature vector of a PhD students working regularly in the lab (shown in Fig. 8(c)) is characterized by the first four entries equal to one. Conversely, the feature vector of a user which visited the lab once for a few hours and never returned has only the first entry different from zero. At the same time, users having similar behaviours in visiting the lab will have similar feature vectors.

Table 3
Average localization errors.

Dataset	WiFi probes	Area [m^2]	Probe Density [probe / m^2]	Localization Method	Mean Error [m]
[6]	12	340	0.035	Fingerprint (kNN)	4.5
[8]	4	30	0.133	Fingerprint (NB)	2
LabLife	4	80	0.05	Fingerprint (RF)	2.06
LabLife	4	80	0.05	Model-based	2.3
LabLife	6	80	0.075	Fingerprint (RF)	1.5
LabLife	6	80	0.075	Model-based	1.9
FieraMilano	4	1296	0.003	Fingerprint (RF)	6.9
FieraMilano	4	1296	0.003	Model-based	7.8
FieraMilano	6	1296	0.005	Fingerprint (RF)	6
FieraMilano	6	1296	0.005	Model-based	6.7

To identify such groups of users, we apply k-medoids clustering on the feature data using Hamming distance as similarity metric.

Since the number of cluster k is not known a-priori, we run k-medoids with k from 1 to 10 and we study the corresponding total within-cluster sum of squares distances. The result is illustrated in Fig. 7: one should locate a kink in the sum of squares curve to identify the optimal number of clusters (the so called *elbow rule*). According to Fig. 7 we set k equal to 2 in our case: the resulting cluster centers are the two feature vectors [1 0 0 0 0] and [1 1 0 1 0]. As one can see, the first cluster is



(a) Four anchor nodes (b) Six anchor nodes

Fig. 6. Cumulative distribution function of the localization error for the *FieraMilano* dataset when using four (a) and six (b) anchor nodes.

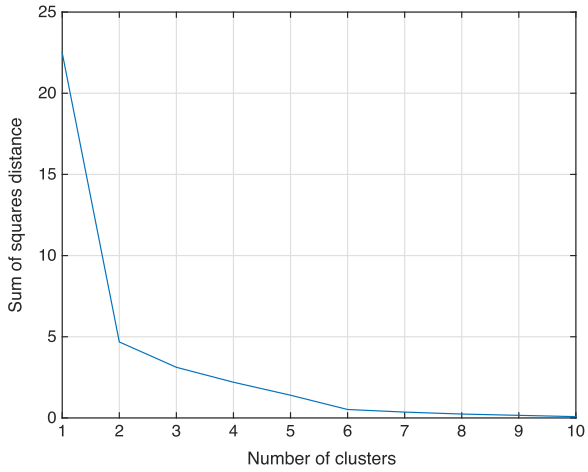


Fig. 7. Total within-cluster sum of squares distances vs. number of clusters. Applying the *elbow rule* results in select $k=2$ as the optimal number of clusters.

identified by users which are detected for a short period of time (the first bin is present), but never return. We denote this first cluster as the group of *passers-by* users. Conversely, the second cluster is represented by users characterized by the peaks corresponding to daily and weekends returns, a behavior typical of people working in the lab. Therefore, we denote the second cluster as the group of *workers*. Interestingly, 73% of the MAC address in the dataset are labeled as passers-by, while only 27% are labeled as workers. This could be explained considering that the lab is closely located to a public park and a pedestrian footway: therefore, many of the captured probe requests may easily be coming from devices belonging to people not related to the lab whatsoever. To further analyze the results of the clustering operation, we conducted a survey asking the MAC addresses of the devices belonging to people working in the lab and in the closest rooms. In the survey, each participant could also indicate a measure of attendance to the lab in the set "mostly every day", "mostly once per week" or "rarely". The survey resulted in the collection of 53 MAC addresses: 41 participants indicated a daily presence, 10 a weekly presence and 2 indicated a rare presence. Such data is used as ground-truth for evaluating the performance of clustering. As a first result, all 53 MAC addresses classified as workers by the *k*-medoid algorithm. To further analyze the performance of clustering, we run again *k*-medoids only on the subset of MAC addresses that participated to the survey. This time we fixed the number of clusters to 3: the resulting cluster centers are [1 1 0 1 0], [1 0 1 1 0] and [1 0 0 1 1]. Table 4 shows how the different MAC addresses in the survey where labeled by the clustering algorithm: the first cluster is again capturing *workers* of the lab, characterized by daily and weekly returns. In fact, 93% of the MAC addresses belonging to users that answered "mostly every day" are belonging to this cluster. The second

Table 4
Clustering results on the *LabLife* dataset.

	Every Day (41)	Once per Week (10)	Rarely (2)
Cluster 1: [1 1 0 1 0]	38 (93%)	2 (20%)	0
Cluster 2: [1 0 1 1 0]	3 (7%)	8 (80%)	0
Cluster 3: [1 0 0 0 1]	0	0	2 (100%)

cluster is characterised by returns corresponding to a day-off or a weekend break, and 80% of the users that answered "mostly once per week" are captured in this cluster. Finally, the users which answered "rarely" are clustered together in the third cluster, which lacks short-time returns completely.

4.2.2. *Clustering the FieraMilano dataset*

We repeated the clustering process also for the 40,223 MAC addresses with valid OUI (i.e., non randomized) contained in he *FieraMilano* dataset, each one represented as a 9-dimensional binary feature vectors. We run *k*-medoids with different *k*-values and analyse the within-cluster sum of squares curve: in this case the kink is observed for $k = 3$, and the corresponding clusters centers are [1 0 0 0 1 0 0 0 0], [1 1 1 1 1 1 1 0 0] and [1 1 0 0 1 1 1 1 1]. The first cluster corresponds to people who visited the exhibition for only one day and never returned. The second cluster corresponds to people who visited the exhibitions for three days and the last cluster corresponds to people whose device were recorded for all five days. Given the particular scenario under consideration, it was not possible to perform any survey asking for MAC addresses. Therefore, to quantify the performance of the method, we rely on the statistics provided by the administration after the exhibition. According to such statistics, 65,237 visitors were admitted to the exhibition. Among them, 24% owned a 1-day ticket, 73% a 3-day ticket and 3% a 5-day ticket. In addition about 1000 staff and security personnel were present in the area during the five days of the exhibition (whose pattern is assumed to be similar to the one belonging to 5-day tickets owners). Table 5 reports the distribution of MAC addresses among the three clusters identified by the proposed algorithm compared to the ground truth data obtained from the statistics. As one can see, the two distributions are very close one to another, confirming the validity of the proposed approach. The discrepancy in the absolute numbers can be attributed to the fact that not all visitors had the Wi-Fi interface of their devices turned on or that their devices transmitted probe requests with a randomized MAC address. Note also that the proposed approach allows to tune the period of times chosen to build the histogram bins, therefore opening up to applications in several other scenarios where users are characterized by different time-dependent dynamics.

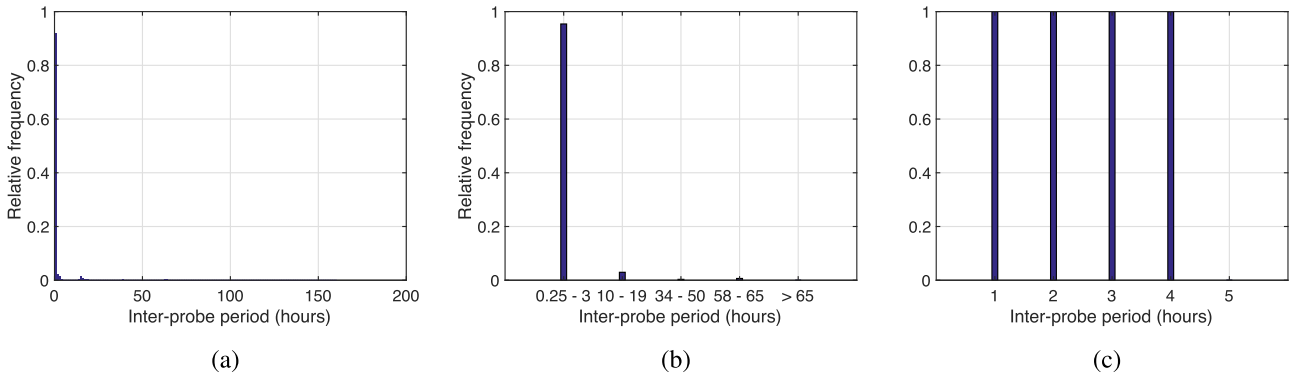


Fig. 8. Creation of the binary feature vector for a specific MAC address. (a) The inter-probe period distribution is created (b) The distribution is quantized in 5 bins corresponding to the natural activity of the users (c) The bins are thresholded and binarized.

Table 5
Clustering results on the *FieraMilano* dataset.

	Estimation	Official Statistics
Cluster 1: [1 0 0 0 1 0 0 0 0]	8203 (21%)	15,627 (23.5%)
Cluster 2: [1 1 1 1 1 1 1 0 0]	30,442 (76%)	47,623 (72%)
Cluster 3: [1 1 0 0 1 1 1 1 1]	1578 (3%)	2957 (4.5%)
TOTAL	40,223	66,237

Table 6
Classification accuracy using only dummy features.

Algorithm	Accuracy
Naive Bayes	0.8029
Support Vector Machine	0.7957
Decision Tree	0.778
Random Forest	0.8129

4.3. Smartphone/Laptop classification

In this section we show how to use probe requests to classify whether a user device is a mobile handheld device (e.g., smartphone) or a non handheld device (laptop). It is already well known that these two device classes produce very different traffic patterns, which impact on the network performance in different ways. Therefore, understanding whether a device belongs to one or the other class is of key importance for at least two reasons:

- *Network traffic analysis:* although the main characteristics of the traffic patterns produced by handheld and non handheld devices have already been studied in the past, the advent of new technologies, trends and applications may change such characteristics significantly. Therefore such analysis must be repeated from time to time with the objective of identifying potential important changes in how the two classes of devices differ [27–30]. Clearly, this operation calls for automatic methods able to classify traffic flows as belonging to one or the other class.
- *Network management and optimization:* related to the previous point, the findings stemming from traffic analysis reveal important information that can be used to implement managing and optimization tools for the entire network. Examples includes customized traffic shaping policies [31], smart content caching approaches [32], management of wide WiFi networks [33,34] and server provisioning [35].

Different approaches to perform device classification are available in the literature. A trivial and inexpensive method is to perform device classification by relying solely on the first 3 octets of a MAC address, known as the Organizationally Unique Identifier (OUI), which identifies uniquely the manufacturer of a device [28,29]. The method works for those manufacturers who make only smartphones (e.g., HTC, Rim, Nokia) or laptops (e.g., Asus, Dell). Unfortunately, most of the devices used nowadays are produced by manufacturers which produce both types of devices (e.g., Samsung, Apple): therefore such a method can be used only on a very small percentage of devices. To overcome this issue, two alternative methods are generally used in the literature, both resorting to traffic inspection tools which "read inside" application-layer traffic: DHCP fingerprinting [34] and inspection of the *User Agent* field of HTTP headers [27,30,33,35,36]. Unfortunately, both methods have two major drawbacks: first, they require the use of dedicated hardware (e.g., Deep Packet Inspectors) or software licenses (e.g., Cisco Identity Services Engine) on the existing network architecture, which might not be always possible due to high costs and administrative, management or privacy issues. Moreover, even when the installation of packet inspection devices is possible, the increase in encrypted traffic makes it hard to extract useful information out of such tools. It's a matter of fact that web giants (Google, Amazon, Facebook, etc.) protect the traffic through their servers with HTTPS: as an example, a recent transparency report from Google [37] stated that 77% of the requests to its servers used encrypted connections, with such percentage destined to increase dramatically in the next few years. Such a trend imposes tight limits on the use of those methods based on the inspection of application layer information such as the *User-Agent* header field, which is encrypted in HTTPS and thus hard or even impossible to analyze. In contrast to such approaches, we show that the very same task of device classification can be performed in a less intrusive and expensive way by passively analyzing probe requests traffic.

The key tenet of this approach is that the probe traffic generated by different classes of devices is homogeneous/similar in the same class. For the classification task at hand, we focus on the following set of features:

- *Inter-Probe Period (IPP):* different devices do transmit probe requests with different temporal frequencies; moreover, the probing pattern of mobile devices is extremely status-dependant; as an example, the probing frequency is generally decreased when the screen is turned off, and each time a user presses a button or unblocks the phone a new probe request is transmitted. We attempt to capture those behaviours with two specific features. In particular, all timestamps t_i of the probe request frames belonging to a single MAC address are extracted and sorted in chronological increasing order in an array $\mathbf{T} = [t_1, t_2, ..., t_{N_s}]$. Let $p_i = t_{i+1} - t_i$ be the i-th inter-probe period. We define the average inter-probe period as:

$$\mu_{p,s} = \frac{1}{N_s - 1} \sum_{i=1}^{N_s - 1} p_i. \tag{4}$$

Similarly, we define the standard deviation of the inter-probe period as:

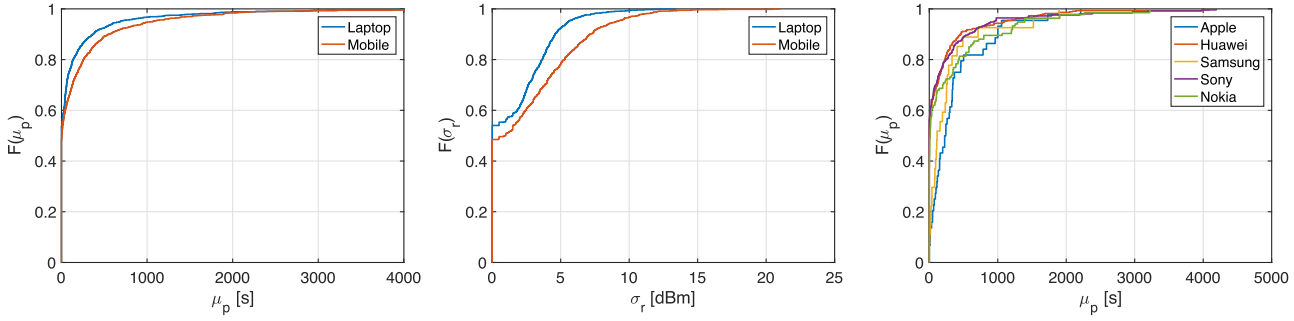


Fig. 9. (a) CDF of the inter-probe period for laptops and mobile devices; (b) CDF of the standard deviation of the RSS for laptops and mobile devices; (c) CDF of the inter-probe period for mobile devices of different vendors (best viewed in color).

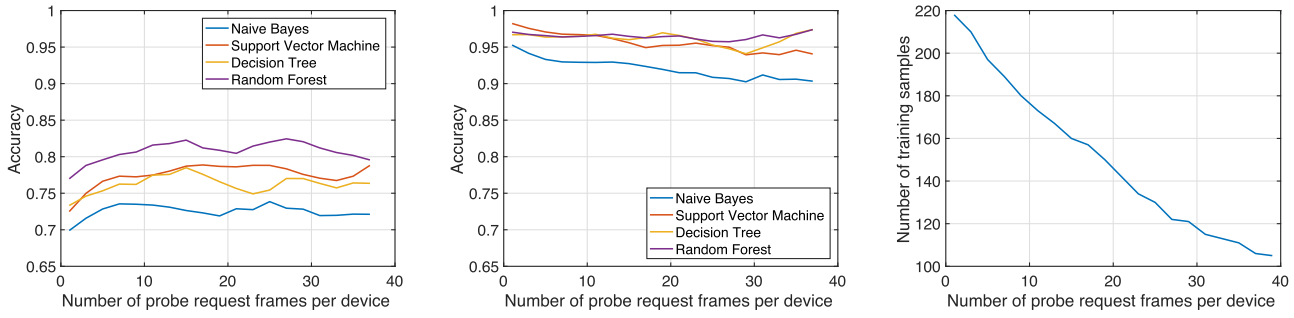


Fig. 10. (a) Classification accuracy when using only quantitative features; (b) Classification accuracy when using both quantitative and dummy features. (c) Number of training samples at different values of N_s. (best viewed in color).

$$\sigma_{p,s} = \sqrt{\frac{1}{N_s - 1} \sum_{i=1}^{N_s-1} (p_i - \mu_{p,s})^2}. \tag{5}$$

Fig. 9(a) shows the Cumulative Distribution Function of the average inter-probe period for laptops and mobile devices. We can observe that laptop devices probe more frequently than smartphones: 50% of all laptops have an inter-probe period of less than 60 seconds, and 95% of them have an IPP of less than 1000 seconds. The IPP for the same percentages of smartphones are considerably higher, 120 seconds and 2300 seconds, respectively.

- *Received Signal Strength (RSS):* The received signal strength measures the power of a probe request as seen from the receiver and depends on the distance between the transmitter and the receiver as well as on other effects characterizing the radio environment (presence of obstacles, mutual antenna orientations, etc.). Similarly to the IPP, we capture the RSS using two features, namely:

$$\mu_{r,s} = \frac{1}{N_s - 1} \sum_{i=1}^{N_s-1} r_i, \tag{6}$$

and

$$\sigma_{r,s} = \sqrt{\frac{1}{N_s - 1} \sum_{i=1}^{N_s-1} (r_i - \mu_{r,s})^2} \tag{7}$$

that is, the average and standard deviation of the RSS of the captured probe request frames. The reason behind the use of such two features is the following: we posit that handheld devices exhibit a higher variance in the RSS compared to non-handheld devices. Indeed, smartphones and tablets are more frequently handled and moved than laptops, creating fluctuations in the RSS measurements captured by $\sigma_{r,\,s}$. Such difference is clearly illustrated in Fig. 9(b): as one can see, 95% of the laptops in our dataset have standard deviation of the RSS lower than 5 dBm, while for mobile devices this value is almost double.

- *Coefficients of variation:* For both IPP and RSS features we also compute the coefficients of variation

$$c_{p,s} = \frac{\sigma_{p,s}}{\mu_{p,s}}, \tag{8}$$

and

$$c_{r,s} = \frac{\sigma_{r,s}}{\mu_{r,s}}. \tag{9}$$

Such coefficients are useful to provide a dimensionless feature and to compare the degree of variation of measurements from different devices regardless of their mean value.

- *Number of probe requests with broadcast/known SSID:* For each source MAC address s we store the number of probe request frames with a "Broadcast" destination SSID $N_{b,\,s}$ and the number of probe request frames with a textual SSID (that is, the SSID of a Wi-Fi network to which the device associated at least once) $N_{k,\,s}$. Note that $N_s = N_{b,s} + N_{k,s}$. We also compute the proportion of broadcast and known probe request frames, that is $\frac{N_{b,s}}{N_s}$ and $\frac{N_{k,s}}{N_s}$. Finally, we also store the number of unique SSIDs contained in the probe request frames, that is $N_{u,\,s}$.
- *Device manufacturer:* Several works in the past have exploited the vendor information contained in the MAC address to infer the class of a device ([28,29]). Given that some vendors produce only mobile or laptop devices, it is reasonable to include the vendor as a feature for classification. We observe that the set of OUIs contained in the database is limited to V different vendors. At the same time, we observe that devices from different vendors have very different probing behaviors. As an example, Fig. 9(c) illustrates the CDF of the inter-probe period for 5 different vendors of mobile devices, with Huawei and Sony devices having the smallest inter-probe period while Apple devices have the largest one. To capture such differences, we create $V - 1$ dummy binary variables $d_{1,s}$, $d_{2,s}$, ..., $d_{V-1,s}$, such that $d_{i,s} = 1$ if s is from the i-th vendor, and $d_{i,s} = 0$, otherwise. Note that the V-*th* vendor is identified by having all $d_{i,\,s}$ equal to zero.

In summary, each device in the *Lectures* database is represented with the following feature feature vector:

$$\mathbf{f} = \left\{\mu_p, \sigma_p, \mu_r, \sigma_r, c_p, c_r, N_b, N_k, \frac{N_b}{N}, \frac{N_k}{N}, d_1, \ldots, d_{V-1}\right\}, \tag{10}$$

where we have suppressed the subscript s for simplicity. Finally, we label each entry in the dataset with its ground truth class "Laptop" or "Mobile". After the feature extraction step, our dataset consists of 279 labeled entries belonging to 150 laptops and 129 mobile devices.

We aim at solving the following problem: given a feature vector f belonging to a device of unknown type T (and computed through processing of sniffed probe request frames, predict whether the device is a laptop or a mobile device. We solve such a problem taking a supervised learning approach: we use different classifier algorithms that are trained with a set of labeled observations and are then evaluated on a set of completely new observations. In particular, we test the following classification algorithms:

- Naïve Bayes (NB): as previously mentioned, this algorithm learns $P(f_i|T)$ by fitting probability distributions to each individual feature: for real valued features, normal (Gaussian) distributions are used, while for binary features (e.g. d_1 to d_{V-1}) binomial distributions are used to model the data. The classifier then returns the most probable class, that is the class T for which $P(T|\mathrm{f})$, computed using Bayes Theorem, is maximized.
- Support Vector Machine (SVM): SVM classifiers are very popular supervised algorithms that construct a hyperplane in the subspace of features so that observation belonging to different classes are

separated by a margin as wide as possible. In addition, when the different classes are not linearly separable, SVMs allows to perform non-linear classification efficiently by first transforming the feature space with a non-linear kernel function, and then constructing a separating hyperplane in the transformed space.

- Decision tree (DT): a decision tree is a classification algorithm that returns the predicted class by iteratively making decisions on the value of the input features. Decisions are learned with a training process, starting with the most discriminative feature at the top (root) of tree and iteratively aggregating decisions in branches, finally arriving to the tree leaves (predicted classes). As a result, the learned tree can be more easily interpreted than a SVM classifier (e.g., it can be displayed graphically). As a drawback, decision trees generally do not have the same level of predictive accuracy as SVM, due to their tendency to overfit the training data.
- Random Forest (RF): as explained before, this ensemble algorithm is generally used to prevent overfitting when using decision trees, and has been shown to perform very well in several machine learning tasks. A random forest classifier constructs several decision trees at training time, and outputs as a prediction the mode of the classes predicted by the individual trees (majority voting).

The performance of such classifiers are obtained resorting to k-fold cross validation: first, the original set of 279 observations in the *Lectures dataset* is divided in k complementary subsets; then, $k-1$ subsets are used for training each classifier, while one is used for testing. The process is repeated k times, averaging the results. Here, we used $k=5$. The performance metric used throughout the tests is the *classification accuracy*, that is the fraction of correctly classified observations over the total number of tests.

We test the performance of the different classifiers in three different scenarios:

- Quantitative features only (QF): we consider only the numerical features extracted from the database of probe requests, that is $\{\mu_p, \sigma_p, \mu_r, \sigma_r, c_p, c_r, N_b, N_k, \frac{N_b}{N}, \frac{N_k}{N}\}$, for training and testing the classifiers. This scenario reflects the case in which the OUI information of a device cannot be read. This can happen if the MAC addresses of the devices are encrypted through randomization, a solution that several vendors are gradually implementing in the operating systems of their devices (e.g., iOS8, Android 6.0).
- Dummy features only (DF): conversely, we consider only the dummy features obtained with the OUI information available from the MAC address to perform classification. This approach applies machine learning techniques to the same information available to other approaches available in the literature [28,29].
- All features (AF): finally, in this scenario, we train and test the classifiers using both quantitative and qualitative features.

Table 6 shows the classification accuracy for the dummy features scenario. As one can see, the different classifiers have similar values of accuracy, around 80%. Note, however, that this value strongly depends on the distribution of device vendors in the dataset. As an example, if the majority of the devices in the dataset is from a vendor that produces both handheld and non handheld devices (e.g., Apple, Samsung), the accuracy of such method is expected to decrease dramatically due to the impossibility to link a vendor with a particular device class.

For the quantitative features and the all features scenarios, the tests are performed considering only those samples belonging to devices whose features are extracted starting from at least N_s probe request frames, each time increasing the value of N_s. Such value as a twofold effect on the performance of the classifiers: on one hand, increasing N_s allows to train the classifiers with more "stable" features, as those features involving mean and standard deviation operation are computed with an increasing set of samples. On the other hand, increasing N_s makes the number of samples available for training the classifiers decrease, as shown in Fig. 10(c). Note also that N_s is related to the amount of time one should spend to capture probe request frames, which increases with N_s.

Fig. 10(a) shows the classification accuracy of the different classifiers when using only quantitative features. As one can see, the accuracy of all classifier tends to increase for small values of N_s and decreases for high values of N_s. The first effect is due to the increasing stability in the computed features, while the latter effect is due to the decreasing number of training observations, as explained above. Overall, the Random Forest classifier exhibits the best performance, with a peak accuracy value of 83% for $N_s = 15$. The performance of the different classifiers in the scenario where both quantitative and qualitative features are used is illustrated in Fig. 10(b). First, it is possible to appreciate the great performance increase given by using both kind of features. In this case, all methods but the Naive Bayes classifier exhibit similar performance, with the Random Forest classifier correctly classifying more than 95% of the test samples. In this case, the positive effect of increasing N_s seems shadowed by the use of dummy features. On the contrary, increasing N_s too much hurts the performance of all classifiers, due to the decrease in the number of training samples.

5. Concluding remarks

We have addressed the issue of extracting as much information as possible out of inexpensive hardware/software systems to passively collect WiFi traffic. To this extent, we have analyzed real-life traffic datasets and we have reported on the realization and performances of three possible use cases/application of probe request-based data analytics: end user localization, end-user profiling and device classification. For the first application, an average localization error of 1.5 m is obtained using fingerprint-based localization and 6 capturing devices deployed in a 80[m^2] area. For what concerns user profiling, we proposed a set of features extracted from the capture time of probe request frames to cluster users in groups based on their dwell time and presence. A comparison with a manually obtained ground truth demonstrated the goodness of our approach. Finally, we showed that probe request frames can be used to distinguish between smartphones and laptop with very good accuracy.

It is worth to point out that the analysis carried out in this work builds on *clear* probe request messages, that is, probe request messages containing *real* MAC addresses; on the other hand, an increasing number of devices is nowadays adopting MAC randomization techniques to obfuscate the sender's identity; even if these techniques in principle make "less informative" the passive sniffing of WiFi control frames, still probe request packets constitute a considerable source of interesting information. In fact, recent studies [38] demonstrate that randomized probe request still retain enough information *entropy* which allows to extract similar knowledge as the one addressed in this work.

References

[1] C. Systems, Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20152020, Technical Report, CISCO, 2016.

[2] IEEE Standard for Information Technology - Telecommunications and Information Exchange between Systems - Local and Metropolitan Networks - Specific Requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications: Higher speed physical layer (phy) extension in the 2.4 ghz band, IEEE Std 802.11b-1999 (2000) 1–96, http://dx.doi.org/10.1109/IEEESTD.2000.90914.

[3] L. Schauer, M. Werner, P. Marcus, Estimating crowd densities and pedestrian flows using Wi-Fi and Bluetooth, Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, MOBIQUITOUS '14, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 2014, pp. 171–177, http://dx.doi.org/10.4108/icst.mobiquitous.2014.257870.

[4] Y. Fukuzaki, M. Mochizuki, K. Murao, N. Nishio, Statistical analysis of actual number of pedestrians for Wi-Fi packet-based pedestrian flow sensing, Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers, UbiComp/ISWC'15 Adjunct, ACM, New York, NY, USA,

2015, pp. 1519–1526, http://dx.doi.org/10.1145/2800835.2801623.
[5] E. Vattapparamban, B.S. Çiftler, İ. Güvenç, K. Akkaya, A. Kadri, Indoor occupancy tracking in smart buildings using passive sniffing of probe requests, Communications Workshops (ICC), 2016 IEEE International Conference on, IEEE, 2016, pp. 38–44.
[6] F. Potortì, A. Crivello, M. Girolami, E. Traficante, P. Barsocchi, Wi-Fi probes as digital crumbs for crowd localisation, Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on, IEEE, 2016, pp. 1–8.
[7] J. Scheuner, G. Mazlami, D. Schöni, S. Stephan, A. De Carli, T. Bocek, B. Stiller, Probr-a generic and passive WiFi tracking system, Local Computer Networks (LCN), 2016 IEEE 41st Conference on, IEEE, 2016, pp. 495–502.
[8] H. Chen, Y. Zhang, W. Li, P. Zhang, Non-cooperative Wi-Fi localization via monitoring probe request frames, Vehicular Technology Conference (VTC-Fall), 2016 IEEE 84th, IEEE, 2016, pp. 1–5.
[9] Y. Chon, S. Kim, S. Lee, D. Kim, Y. Kim, H. Cha, Sensing WiFi packets in the air: Practicality and implications in urban mobility monitoring, Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '14, ACM, New York, NY, USA, 2014, pp. 189–200, http://dx.doi.org/10.1145/2632048.2636066.
[10] A.B.M. Musa, J. Eriksson, Tracking unmodified smartphones using Wi-Fi monitors, Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, SenSys '12, ACM, New York, NY, USA, 2012, pp. 281–294, http://dx.doi.org/10.1145/2426656.2426685.
[11] P. Rouveyrol, P. Raveneau, M. Cunche, Large Scale Wi-Fi tracking using a Botnet of Wireless Routers, Workshop on Surveillance & Technology, Philadelphia, United States, (2015).
[12] B. Bonn, A. Barzan, P. Quax, W. Lamotte, Wifipi: Involuntary tracking of visitors at mass events, World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a, (2013), pp. 1–6, http://dx.doi.org/10.1109/WoWMoM.2013.6583443.
[13] S. Du, J. Hua, Y. Gao, S. Zhong, Ev-linker: Mapping eavesdropped Wi-Fi packets to individuals via electronic and visual signal matching, J. Comput. Syst. Sci. 82 (1, Part B) (2016) 156–172. Mobile Social Networking and computing in Proximity (MSNP): A Multi-disciplinary Inspired Approach https://doi.org/10.1016/j.jcss.2015.06.005 .
[14] M. Cunche, I know your mac address: targeted tracking of individual using Wi-Fi, J. Comput. Virol. Hacking Techniques 10 (4) (2014) 219–227, http://dx.doi.org/10.1007/s11416-013-0196-1.
[15] C. Matte, J.P. Achara, M. Cunche, Device-to-identity linking attack using targeted Wi-Fi geolocation spoofing, Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks, WiSec '15, ACM, New York, NY, USA, 2015, pp. 20:1–20:6, http://dx.doi.org/10.1145/2766498.2766521.
[16] M. Chernyshev, C. Valli, P. Hannay, On 802.11 access point locatability and named entity recognition in service set identifiers, IEEE Trans. Inf. Forensics Secur. 11 (3) (2016) 584–593, http://dx.doi.org/10.1109/TIFS.2015.2507542.
[17] A. Di Luzio, A. Mei, J. Stefa, Mind your probes: De-anonymization of large crowds through smartphone wifi probe requests, Computer Communications (INFOCOM), 2016 IEEE Conference on, (2016).
[18] W. Qin, J. Zhang, B. Li, L. Sun, Discovering human presence activities with smartphones using nonintrusive Wi-Fi sniffer sensors: the big data prospective, Int. J. Distrib. Sens. Netw. (2013), http://dx.doi.org/10.1155/2013/927940.
[19] R. Kalogianni, E. Sileryte, M. Lam, K. Zhou, M. Van der Ham, S. Van der Spek, E. Verbree, Passive wifi monitoring of the rhythm of the campus, Proceedings of The 18th AGILE International Conference on Geographic Information Science, (2015), pp. 1–4.
[20] J. Freudiger, How talkative is your mobile device?: An experimental study of wi-fi probe requests, Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks, WiSec '15, ACM, New York, NY, USA, 2015, pp. 8:1–8:6, http://dx.doi.org/10.1145/2766498.2766517.
[21] N. Cheng, P. Mohapatra, M. Cunche, M.A. Kaafar, R. Boreli, S. Krishnamurthy, Inferring user relationship from hidden information in wlans, MILCOM 2012 - 2012 IEEE Military Communications Conference, (2012), pp. 1–6, http://dx.doi.org/10.1109/MILCOM.2012.6415713.
[22] M. Cunche, M.A. Kaafar, R. Boreli, I know who you will meet this evening! linking wireless devices using wi-fi probe requests, World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a, (2012), pp. 1–9, http://dx.doi.org/10.1109/WoWMoM.2012.6263700.
[23] M. Cunche, M.-A. Kaafar, R. Boreli, Linking wireless devices using information contained in wi-fi probe requests, Pervasive Mob. Comput. 11 (2014) 56–69 https://doi.org/10.1016/j.pmcj.2013.04.001.
[24] M.V. Barbera, A. Epasto, A. Mei, V.C. Perta, J. Stefa, Signals from the crowd: Uncovering social relationships through smartphone probes, Proceedings of the 2013 Conference on Internet Measurement Conference, IMC '13, ACM, New York, NY, USA, 2013, pp. 265–276, http://dx.doi.org/10.1145/2504730.2504742.
[25] S. Seneviratne, F. Jiang, M. Cunche, A. Seneviratne, Ssids in the wild: Extracting semantic information from wifi ssids, Local Computer Networks (LCN), 2015 IEEE 40th Conference on, (2015), pp. 494–497, http://dx.doi.org/10.1109/LCN.2015.7366361.
[26] V. Acuna, A. Kumbhar, E. Vattapparamban, F. Rajabli, I. Guvenc, Localization of wifi devices using probe requests captured at unmanned aerial vehicles, Wireless Communications and Networking Conference (WCNC), 2017 IEEE, IEEE, 2017, pp. 1–6.
[27] G. Maier, F. Schneider, A. Feldmann, A first look at mobile hand-held device traffic, Passive and Active Measurement, Springer, 2010, pp. 161–170.
[28] U. Kumar, J. Kim, A. Helmy, Changing patterns of mobile network (wlan) usage: Smart-phones vs. laptops, Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International, (2013), pp. 1584–1589, http://dx.doi.org/10.1109/IWCMC.2013.6583792.
[29] U. Kumar, J. Kim, A. Helmy, Comparing wireless network usage: Laptop vs Smart–phones, Proceedings of the 19th annual international conference on Mobile computing & networking, ACM, 2013, pp. 243–246.
[30] M. Zhu, Z. Zeng, L. Wang, Z. Qin, A. Pan, Y. Zhang, L. Shu, A measurement study of a campus wi-fi network with mixed handheld and non-handheld traffic, Electrical and Computer Engineering (CCECE), 2015 IEEE 28th Canadian Conference on, IEEE, 2015, pp. 848–853.
[31] S. Gebert, R. Pries, D. Schlosser, K. Heck, Internet access traffic measurement and analysis, Traffic Monitoring Anal. (2012) 29–42.
[32] I. Papapanagiotou, E.M. Nahum, V. Pappas, Smartphones vs. laptops: comparing web browsing behavior and the implications for caching, ACM SIGMETRICS Performance Evaluation Review, 40 ACM, 2012, pp. 423–424.
[33] A. Gember, A. Anand, A. Akella, A comparative study of handheld and non-handheld traffic in campus Wi-Fi networks, Proceedings of the 12th International Conference on Passive and Active Measurement, PAM'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 173–183.
[34] X. Wei, N. Valler, H. Madhyastha, I. Neamtiu, M. Faloutsos, A behavior-aware profiling of handheld devices, Computer Communications (INFOCOM), 2015 IEEE Conference on, (2015), pp. 846–854, http://dx.doi.org/10.1109/INFOCOM.2015.7218455.
[35] X. Chen, R. Jin, K. Suh, B. Wang, W. Wei, Network performance of smart mobile handhelds in a university campus Wi-Fi network, Proceedings of the 2012 ACM conference on Internet measurement conference, ACM, 2012, pp. 315–328.
[36] A. Ruiz-Ruiz, H. Blunck, T. Prentow, A. Stisen, M. Kjaergaard, Analysis methods for extracting knowledge from large-scale Wi-Fi monitoring to inform building facility planning, Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on, (2014), pp. 130–138, http://dx.doi.org/10.1109/PerCom.2014.6813953.
[37] Google transparency report: Https at Google(2016).
[38] M. Vanhoef, C. Matte, M. Cunche, L.S. Cardoso, F. Piessens, Why Mac address randomization is not enough: An analysis of Wi-Fi network discovery mechanisms, Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, ASIA CCS '16, ACM, New York, NY, USA, 2016, pp. 413–424, http://dx.doi.org/10.1145/2897845.2897883.