Passive classification of Wi-Fi enabled devices

Alessandro E. C. Redondi, Davide Sanvito, Matteo Cesana
Dipartimento di Elettronica, Bioingegneria e Informazione (DEIB)
Politecnico di Milano
Piazza Leonardo da Vinci, 32
Milano, Italy
name.surname@polimi.it

ABSTRACT

We propose a method for classifying Wi-Fi enabled mobile handheld devices (smartphones) and non-handheld devices (laptops) in a completely passive way, that is resorting neither to traffic probes on network edge devices nor to deep packet inspection techniques to read application layer information. Instead, classification is performed starting from *probe requests* Wi-Fi frames, which can be sniffed with inexpensive commercial hardware. We extract distinctive features from probe request frames (how many probe requests are transmitted by each device, how frequently, etc.) and take a machine learning approach, training four different classifiers to recognize the two types of devices. We compare the performance of the different classifiers and identify a solution based on a Random Decision Forest that correctly classify devices 95% of the times. The classification method is then used as a pre-processing stage to analyze network traffic traces from the wireless network of a university building, with interesting considerations on the way different types of devices uses the network (amount of data exchanged, duration of connections, etc.). The proposed methodology finds application in many scenarios related to Wi-Fi network management/optimization and Wi-Fi based services.

CCS Concepts

•**Networks** → **Network measurement;** Network monitoring;

Keywords

Device classification; probe requests analysis; traffic analysis

1. INTRODUCTION

Network traffic from wireless devices will exceed traffic from wired devices by 2019, accounting for 66% of the total IP traffic [1]. That is almost double with respect to 2013, when non-PC devices generated 33 percent of the total IP traffic.

For this reason, in the last few years there has been a constantly increasing attention towards the analysis and the profiling of traffic generated by WiFi-enabled devices, with particular focus on the so called Bring Your Own Devices (BYOD), that is smartphones, laptops and tablets more and more frequently brought by people on their workplaces or study places. Such a class of devices is substantially different compared to traditional wired PCs for what concerns the users behavior and the traffic pattern they produce. Indeed, BYODs join and leave the network frequently and their shape and size makes it possible to use them almost everywhere. In addition, the vast range of networked applications running on top of such devices (instant messaging, social networking, video streaming, online videogames, etc...) makes them a concern for a series of reasons, including network management and security [7].

At the same time, WiFi-enabled BYOD devices can be separated in two different classes: mobile handheld devices (MHD), composed of smartphones and tablets, and non handheld devices (NHD) or laptops. The two classes differ in a series of physical and technical features (size, weight, battery capacity, type of wireless connectivity, operating system, etc.) and are generally used for different purposes, with direct implication on the network traffic pattern they produce. In this context, the knowledge (or prediction) of what type of traffic (and devices) is actually using the network can be leveraged to optimize the network configuration and/or implement and support several services (e.g., management of wide WiFi networks, smart content caching approaches, etc.) [11, 8, 3, 15, 9, 10, 19, 18].

Clearly, any analysis on the traffic differences between MHD and NHD builds on the capability to classify traffic flows as belonging to MHD or NHD devices. The available work in the field can be broadly grouped in two classes: (i) approaches that exploit only the Medium Access Control (MAC) addresses contained in WiFi frames generated by the devices, (ii) approaches that resort to some type of *active* traffic/packet inspection tool available in the reference network (e.g., direct access to DHCP logs, inspection of the *User-Agent* field of HTTP headers, etc.). Both classes though have drawbacks. Namely, the approaches based on MAC addresses perform device classification just by looking at the vendor information contained in the Organizationally Unique Identifiers (OUIs) of the MAC address; however, since some of the most popular vendors (e.g. Apple, Samsung) produce both handheld and non-handheld

MSWIM '16 Malta
© 2016 ACM. ISBN 123-4567-24-567/08/06...$15.00
DOI: 10.475/123_4

devices, many devices are excluded from the classification due to the impossibility to assign them to a specific class only by looking at their vendor.

On the other hand, the approaches based on active traffic/packet inspection do have two major drawbacks: deploying traffic/packet inspection probes in the network might not be always possible, and, even when this 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 [2] 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 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 to analyze.

For these reasons, we propose here a less invasive but still effective way to perform device classification. Our proposal is entirely *passive*, in that it does resort neither to traffic probes on network edge devices nor to deep packet inspection techniques to read out application layer information. Instead, we claim that device classification can be performed by collecting (and parsing) only *probe requests* Wi-Fi management frames. Such frames are transmitted in-the-clear by any Wi-Fi enabled device to requests information from in-range access points, can be captured with almost any low-cost commercially available Wi-Fi interface and carry enough information to perform device classification accurately. Our proposed classification framework first labels each device with a set of *features* extracted from the probe request frames the device itself is generating; the reference set of feature capture 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 the transmitting device just by looking at its corresponding *features*.

The rest of this paper is structured in the following way: Section 2 and Section 3 describe how we collected the data used to train the classifiers and which are the features extracted from the captured probe request frames. Section 4 describes the supervised classification approaches and reports on their performance evaluation. A selected classification method is then used to perform the analysis of Wi-Fi network traffic in a university campus: results of such an analysis are reported in Section 5. Section 6 summarizes recent works related to MHD / NHD traffic analysis, focusing in particular on the device classification methods used, as well as works related to probe request frames analysis. Finally, Section 7 concludes the paper.

2. DATA COLLECTION

Our dataset consists of network data traces lasting several hours and containing only sniffed probe request frames collected during particular university classes ("tutorials" or "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 for our classification methods. In addition to those entries whose MAC addresses are labeled by students as belonging to either the "laptop" or "mobile" class, 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". The data is collected using a standard laptop running Linux and equipped with a Wi-Fi card set in *monitor* mode on 802.11 channel 1. We used *tshark* (the terminal version of WireShark) to capture only probe request frames, which are stored in a local MySQL database for further analysis. Each database entry thus contains the following fields: source MAC address, OUI, timestamp, probe request sequence number, received signal strength (RSS) and the Service Set Identifier (SSID) of the probe request. Note that the latter can be either "Broadcast" or a string containing the SSID of a Wi-Fi network known to the device. In total, our database consists of more than $2{\times}10^5$ different probe request entries, spanning 10 different hours over 5 days and belonging to 279 different devices of known type (groundtruth). For simplicity, let N_s be the number of entries in the database having s as source MAC address.

3. FEATURE EXTRACTION

For each MAC address contained in the database the following features are extracted:

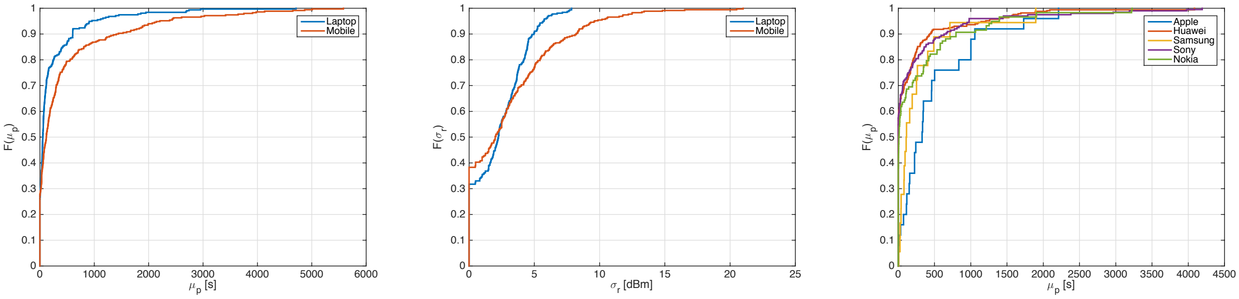
- *Inter-Probe Period (IPP):* Many works related to probe requests analysis have highlighted that different devices transmit probe requests with different temporal frequencies. Moreover, mobile devices vary a lot their probing pattern depending on their status. As an example, the probing frequency is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

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

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

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

Figure 1(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



(a) (b) (c)

Figure 1: (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)

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 (sniffer) 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:

$$\mu_{r,s} = \frac{1}{N_s - 1} \sum_{i=1}^{N_s - 1} r_i, \tag{3}$$

and

$$\sigma_{r,s} = \sqrt{\frac{1}{N_s - 1} \sum_{i=1}^{N_s - 1} (r_i - \mu_{r,s})^2} \tag{4}$$

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 Figure 1(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

$$c_{p,s} = \frac{\sigma_{p,s}}{\mu_{p,s}}, \tag{5}$$

and

$$c_{r,s} = \frac{\sigma_{r,s}}{\mu_{r,s}}. \tag{6}$$

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 ([9, 10]). 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, Figure 1(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 $d_{1,s}, d_{2,s}, \ldots, d_{V-1,s}$, such that:

$$d_{i,s} = \begin{cases} 1 & \text{if } s \text{ is from the } i\text{-th vendor} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Note that the V-*th* vendor is identified by having all $d_{i,s}$ equal to zero.

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

$$\mathbf{f} = \{\mu_p, \sigma_p, \mu_r, \sigma_r, c_p, c_r, N_b, N_k, \frac{N_b}{N}, \frac{N_k}{N}, d_1, \ldots, d_{V-1}\}, \tag{8}$$

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.

4. CLASSIFICATION ALGORITHMS

We aim at solving the following problem: given a feature vector $\mathbf{f}$ belonging to a device of unknown type T (and computed through processing of sniffed probe request frames as explained in Section 3), predict wether 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): this simple algorithm assigns to the feature vector $\mathbf{f}$ a probability value $P(T|\mathbf{f})$, computed using the Bayes Theorem and assuming that features are independent, that is:

  $$P(T|\mathbf{f}) = \frac{P(\mathbf{f}|T)P(T)}{P(\mathbf{f})} = P(T)\prod_i P(f_i|T), \qquad (9)$$

  where f_i denotes the i-th component of $\mathbf{f}$ and the denominator $P(\mathbf{f})$ can be ignored as it is the same for all classes. In the training phase, the Naïve Bayes classifier 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. In the test phase, given a newly observed feature vector $\mathbf{f}$, the NB classifier returns the most probable class, that is the class T for which $P(T|\mathbf{f})$ 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): 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 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.

Table 1: 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

The performance of such classifiers are obtained resorting to k-fold cross validation: first, the original set of 279 observations 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:

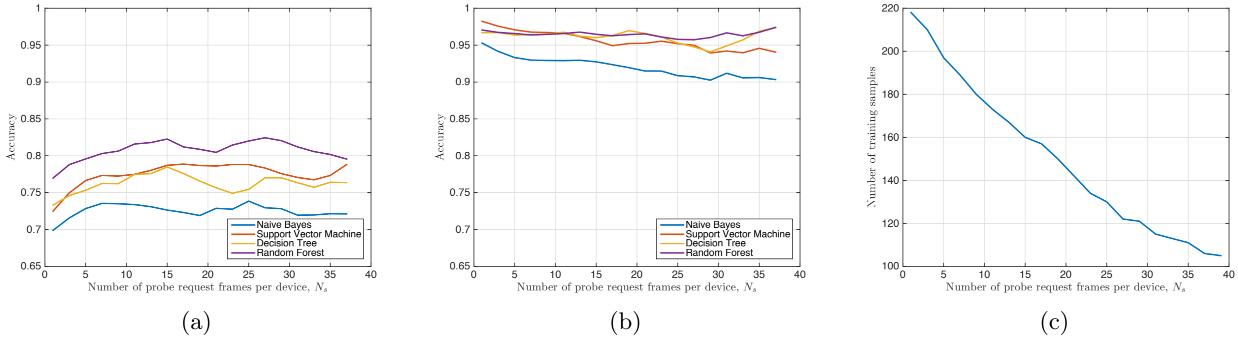
- Quantitative features only (QF): we consider only the numerical features extracted from the database of probe requests, that is $\{\mu_p, \sigma_p, \mu_r, \sigma_r, c_p, c_r, N_b, N_k, \frac{N_b}{N}, \frac{N_k}{N}\}$, for training and testing the classifiers. This scenario reflects the case in which 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 [9, 10].

- All features (AF): finally, in this scenario, we train and test the classifiers using both quantitative and qualitative features.

Table 1 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



(a) (b) (c)

Figure 2: (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)

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 Figure 2(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.

Figure 2(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 Figure 2(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. TRAFFIC ANALYSIS

This section shows the results of the analysis of network data traces extracted from the wireless network of a university campus building, performed after applying the proposed classification method to identify which traces belong to MHD or NHD. The wireless network under study is composed of 28 different wireless access points (AP) located on four different floors of the building. The access points run the AirWave Management Platform system, which allows to observe every device connected to the wireless network. For each access point, the uplink/downlink bandwidth usage and the number of connected clients are available with a sampling period of 5 minutes. Additionally, for each connected client, the following information are available: MAC address of the device, timestamp of the association with the AP, duration of the session (time elapsed from the association with the AP), average and variance of the bandwidth usage during the session [kbps] as well as average and variance of the signal quality during the session [dB]. We focus our analysis on a period of two weeks, from the 20th of May, 2016 to the 3rd of June, 2016. A single Raspberry PI 3 coupled with a Netgear WNA1100 Wi-Fi dongle in monitor mode is used to capture probe request frames. Such device is placed in an open space of the building that students use to study, work on their projects or simply pass time between two lectures. Such a place is therefore characterized with a good mix of MHD and NHD devices, whose emitted probe request frames are captured by the Raspberry PI. We analyze the frames with our classification algorithm (using the Random Forest classifier fed with both quantitative and qualitative features), labeling each observed MAC address as "Laptop" or "Mobile". Note that, differently from what proposed in Section 4, no groundtruth is available to assess the accuracy of our classification. We restrict the analysis of the data traces from the AirWave Management Platform only to those devices seen and classified by our method. Over the two weeks object of our analysis, a total of 2519 unique devices were observed, generating a total of 10287 different sessions. Figure 3 shows the distribution of different vendors in our dataset: note that Apple and Samsung, vendors who produce both laptops and smartphones, together sum up to almost 40% of the total devices seen. This confirms that approaches only based on the OUI for device classification may exclude a lot of data from the analysis. Table 2 reports the result of our classification on the available data. As one can see, over 75% percent of the observed devices and sessions are classified as non handheld devices. Considering that the university wireless network bandwidth is limited to 2 Mbps per user, the results in Table 2 can be due to a growing tendency of mobile users to use their cellular connections (e.g., LTE, 3G) instead of Wi-Fi in the university campus to experience better quality of service.

5.1 Session start time

First, we look at the distribution of starting time of sessions. We identify the minute of the day (from 1 to 1440) at which each connection starts and plot in Figure 4(a) its probability distribution for MHD and NHD device. Several observations can be made from the inspection of such dis-

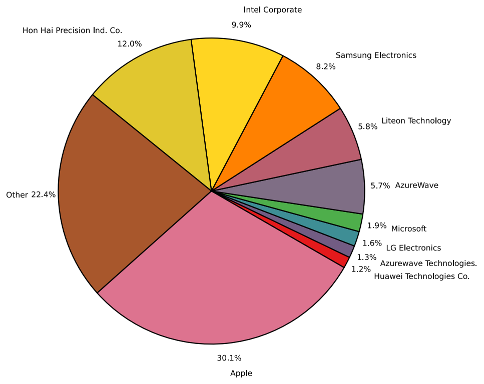


Figure 3: Distribution of the different device vendors (best viewed in color)

tribution:

1. The global trend follows the human daily pattern, with very few sessions started during night. Two sharp peaks are visible in the morning and in the afternoon, divided by a valley corresponding to the lunch break at minute 800 (1.30 PM).

2. From minutes 1 to 180 and from 1200 to 1440, corresponding to the period 8 PM - 3 AM, the probability of a MHD starting a session is higher than that of a NHD. This can be explained taking into account that online activities at the end of a working day (chatting, email checking) are carried out more frequently with a MHD.

3. Conversely, during the day, there is no clear difference between MHD and NHD devices: in the morning (8:30 - 10:30 AM) we observe a higher probability for MHD device, while in the afternoon NHD sessions are more frequent.

5.2 Network usage

Then, we analyze how MHD and NHD devices use the network. Figure 4(b) shows the cumulative distribution function of the network usage (total volume of data exchanged with the AP divided by the duration of the session) in kbps for the two different class of devices, together with the CDF obtained considering all devices together. As one can see there is a dramatic difference between MHD and NHD usage: handheld devices have an average usage of about 50 kbps, while non-handheld devices have an average usage which is more than 4 times higher (203.9 kbps). Also, 95% of MHD have an usage below 200 kbps, while for NHD this value raises to 1 Mbps. This can be explained considering the different applications typically run on MHD (e.g., email, messaging services, quick browsing, etc...) compared to NHD (file download, heavy browsing, etc.). Indirectly, such a result confirms the goodness of our classification method in segmenting MHD and NHD devices.

Table 2: Classification results on the period of two weeks

Observed Devices	**MHD**	**NHD**
2519	658 (26.12%)	1861 (73.88%)
Observed Sessions	**MHD**	**NHD**
10287	2429 (23.61%)	7858 (76.39%)

5.3 Session duration

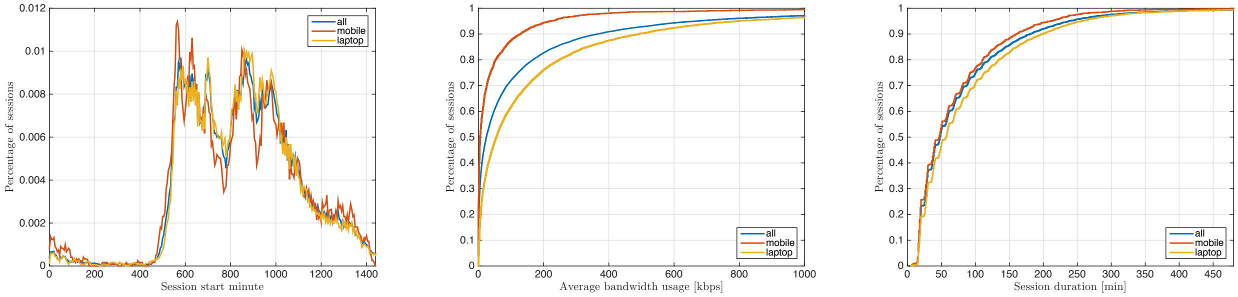
Finally, we also look at the distribution of the duration of the sessions in Figure 4(c). Differently from previous studies ([9, 10]), in which the duration of sessions of MHD was observed as notably lower than NHD, here we do not find such a great difference. The average session duration for MHD is 70.4 minutes, while for NHD is 86.5 minutes. Coupling this result with what explained previously in Sections 5.1 and 5.2, it seems that both MHD and NHD users tend to remain connected to the network for a long period of time and their behaviour differ just in the amount of data transferred over the network and partially in when they start connections with the network.

6. RELATED WORK

To the best of our knowledge, the first work analyzing differences in the traffic behavior of MHD and NHD devices is the one from Maier et al. [11], where network data from residential DSL lines spanning a period of 11 months is analyzed. To identify which DSL lines hosted MHD and hence to identify the corresponding traffic traces, the authors rely on the user-agent strings contained in HTTP headers, which are generally precise indicators of a device and its operating system. For non-HTTP traffic traces, the authors take advantage of the IP TTL field, which turns out to be different in MHD operating systems compared to the most commonly used PC OSs. As a result of such an analysis, the authors show that MHD traffic is dominated by multimedia content and downloads of mobile applications, and that MHD HTTP objects are larger on average than NHD ones.

In [8] network traffic traces from a campus wireless networks are analyzed by examining their content and flow properties (transport and application protocols used, flow length and durations, etc..). Handheld devices are filtered looking at HTTP user-agents as primary method, followed by a confirmation step using the Organizationally Unique Identifiers (OUIs) contained in the MAC addresses: 14% of the devices remain uncategorized and are excluded from the analysis. The key findings of such an analysis are that (i) the majority of handheld traffic is HTTP, (ii) the top content type for MHD is video and (iii) MHD tend to have smaller TCP flows and narrower range of flow durations, compared to NHD. A similar study with comparable results is performed by Zhu et al. in [19], where tcpdump traces from the Dalian university of technology are analyzed. Once again, device classification is performed by relying on the User-Agent field in HTTP headers.

Chen et al. in [3] analyze 3 days of WiFi network data collected by a monitor located at a gateway router fo the network. Again, MHD identification is performed by looking at keywords in the HTTP user-agent. The authors report a small increase in the percentage of HTTPS flows (4.3%), which are impossible to classify as belonging to MHD or NHD. The main findings are that MHD have longer local RTTs, and that the number of concurrent flows has negative effect on performance (and this effect is more significant on



(a) (b) (c)

Figure 4: (a) Probability distribution of the starting minute of Wi-Fi sessions; (b) Cumulative distribution function of the average bandwidth usage for MHD and NHD devices; (c) Cumulative density function of the average session duration for MHD and NHD devices (best viewed in color)

MHD than on NHD).

Papapanagiotou et al. in [15] study the web browsing behaviour of MHD and NHD devices on a 3 weeks long full-packet trace in a wireless enterprise environment. Classification is performed analyzing DHCP request header fields (Host-name, Vendor-name) and the OUIs, allowing to classify 97% of the devices (although no ground truth is provided [14]). The main finding of the analysis is that (i) NHD devices have more intelligent browser caching capabilities and (ii) a 10 MB browser cache in smatphones would be enough to provide 10% - 20% bandwidth savings.

In [9] and [10] Kumar et al. analyze wireless association traces connected at hte University of Florida. Classification of MHD and NHD is done only via inspection of the OUI, since no other information is available. Since some manufacturers (e.g. Apple and Samsung) produce both MHD and NHD, the authors conduct a user survey where users of smartphones were asked to give the first 3 octets of their MAC address (i.e., the OUI). The authors performed then classification assuming that a manufacturer does not use the same MAC address range for both smart-phones and laptops. The analysis include several spatio-temporal features (average session duration, number of session per days, mobility etc.), and shows dramatic differences between MHD and NHD behaviors.

Finally, Wei et al. in [18] propose Brofiler, an approach for studying how MHD behaves along different dimensions (protocol and control plane, data plane, temporal behavior). MHD and NHD are classified with the method proposed in [14] (i.e., looking at DHCP logs). Interestingly, the authors find that 24% of MHD have 50% of their traffic encrypted. This confirms the increasing trend in the use of HTTPS, with serious implication for those classification methods that use the HTTP User-Agent field.

In the last few years, many works have focused on the analysis of Wi-Fi probe request frames sniffed with off-the-shelf hardware. Such data traces have been used for several purposes, including estimating crowd densities and pedestrian flows [17, 6], user tracking and trajectory estimation [13, 16], privacy-related issues and device-to-identity linking [12, 5, 4].

7. CONCLUSIONS

We proposed a method for classifying handheld (smartphones) and non-handheld (laptops) Wi-Fi enabled devices in a passive way, relying only on probe request frames captured with low-cost, commercially available hardware. We compared different algorithms to perform such a classification and identified a solution which correctly classifies the devices more than 95% of the times. Finally, we have used the proposed method to classify devices and performed an analysis of the network traffic traces in a university building, identifying interesting differences in the behavior of handheld and non-handheld devices. We believe that the proposed classification method can be used as a pre-processing stage in many scenarios related to Wi-Fi network management and optimization and Wi-Fi based services. To cite an example, we plan to apply the proposed methodology to improve the performance of localization systems based on radio-map fingerprinting by constructing different radio maps for MHD or NHD devices.

8. REFERENCES

[1] Cisco visual networking index: Forecast and methodology, 2014-2019 white paper. May 2015.

[2] Google transparency report: Https at google. Jan 2016.

[3] X. Chen, R. Jin, K. Suh, B. Wang, and W. Wei. Network performance of smart mobile handhelds in a university campus wifi network. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 315–328. ACM, 2012.

[4] M. Cunche. I know your mac address: Targeted tracking of individual using wi-fi. *Journal of Computer Virology and Hacking Techniques*, 10(4):219–227, 2014.

[5] S. Du, J. Hua, Y. Gao, and S. Zhong. Ev-linker: Mapping eavesdropped wi-fi packets to individuals via electronic and visual signal matching. *Journal of Computer and System Sciences*, 82(1):156–172, 2016.

[6] Y. Fukuzaki, M. Mochizuki, K. Murao, and N. Nishio. Statistical analysis of actual number of pedestrians for wi-fi packet-based pedestrian flow sensing. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pages 1519–1526. ACM, 2015.

[7] B. M. Gaff. Byod? omg! *Computer*, 48(2):10–11, 2015.

[8] A. Gember, A. Anand, and A. Akella. A comparative study of handheld and non-handheld traffic in campus wi-fi networks. In *Passive and Active Measurement*, pages 173–183. Springer, 2011.

[9] U. Kumar, J. Kim, and A. Helmy. Changing patterns of mobile network (wlan) usage: smart-phones vs. laptops. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pages 1584–1589. IEEE, 2013.

[10] U. Kumar, J. Kim, and A. Helmy. Comparing wireless network usage: laptop vs smart-phones. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 243–246. ACM, 2013.

[11] G. Maier, F. Schneider, and A. Feldmann. A first look at mobile hand-held device traffic. In *Passive and Active Measurement*, pages 161–170. Springer, 2010.

[12] C. Matte, J. P. Achara, and M. Cunche. Device-to-identity linking attack using targeted wi-fi geolocation spoofing. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, page 20. ACM, 2015.

[13] A. Musa and J. Eriksson. Tracking unmodified smartphones using wi-fi monitors. In *Proceedings of the 10th ACM conference on embedded network sensor systems*, pages 281–294. ACM, 2012.

[14] I. Papapanagiotou, E. M. Nahum, and V. Pappas. Configuring dhcp leases in the smartphone era. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 365–370. ACM, 2012.

[15] I. Papapanagiotou, E. M. Nahum, and V. Pappas. Smartphones vs. laptops: comparing web browsing behavior and the implications for caching. In *ACM SIGMETRICS Performance Evaluation Review*, volume 40, pages 423–424. ACM, 2012.

[16] P. Rouveyrol, P. Raveneau, and M. Cunche. Large scale wi-fi tracking using a botnet of wireless routers. In *Workshop on Surveillance & Technology*, 2015.

[17] L. Schauer, M. Werner, and P. Marcus. Estimating crowd densities and pedestrian flows using wi-fi and bluetooth. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 171–177. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[18] X. Wei, N. C. Valler, H. V. Madhyastha, I. Neamtiu, and M. Faloutsos. A behavior-aware profiling of handheld devices. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 846–854. IEEE, 2015.

[19] M. Zhu, Z. Zeng, L. Wang, Z. Qin, A. Pan, Y. Zhang, and L. Shu. A measurement study of a campus wi-fi network with mixed handheld and non-handheld traffic. In *Electrical and Computer Engineering (CCECE), 2015 IEEE 28th Canadian Conference on*, pages 848–853. IEEE, 2015.