WiFi Probes sniffing: an Artificial Intelligence based approach for MAC addresses de-randomization

Marco Uras*, Raimondo Cossu*, Enrico Ferrara†, Ovidiu Bagdasar†, Antonio Liotta‡ and Luigi Atzori*
* Department of Electrical and Electronic Engineering (DIEE), UdR CNIT, University of Cagliari, IT
(e-mail: Marco.Uras@unica.it, Raimondo.Cossu@unica.it, L.Atzori@ieee.org)
† Data Science Research Centre (DSRC), University of Derby, Derby, UK
(e-mail: E.Ferrara@derby.ac.uk, O.Bagdasar@derby.ac.uk)
‡ Faculty of Computer Science - Free University of Bozen-Bolzano, Italy
(e-mail: liotta.antonio@gmail.com)

***Abstract*—To improve city services, local administrators need to have a deep understanding of how the citizens explore the city, use the relevant services, interact and move. This is a challenging task, which has triggered extensive research in the last decade, with major solutions that rely on analysing traces of network traffic generated by citizens WiFi devices. One major approach relies on catching the probe requests sent by devices during WiFi active scanning, which allows for counting the number of people in a given area and to analyse the permanence and return times. This approach has been a solid solution until some manufacturer introduced the *MAC address randomization* process to improve the user's privacy, even if in some circumstances this seems to deteriorate network performance as well as the user experience. In this work we present a novel techniques to tackle the limitations introduced by the randomization procedures and that allows for extracting data useful for smart cities development. The proposed algorithm extracts the most relevant *information elements* within probe requests and apply *clustering algorithms* (such as DBSCAN and OPTICS) to discover the exact number of devices which are generating probe requests. Experimental results showed encouraging results with an accuracy of 65.2% and 91.3% using the DBSCAN and the OPTICS algorithms, respectively.**

I. Introduction

In the last decade, understanding how people move around the city is becoming increasingly important for the local administrators for a better design of smart cities services. An approach that is often followed in this context is based on the analysis of traffic generated by our personal mobile device. For example, in the UK, the government started an experiment where some smartphone-monitoring bins can track people through the WiFi interfacing, thus obtaining key information on people's behaviour and adjusting the service accordingly [1]. Another solution has been implemented by Cloud4Wi, which tracks people when moving around in shops and malls and are then able to provide various information about the areas of greatest interest among the shoppers [2]. In [3], a system for Smart Cities scenarios is presented; on the basis of the WiFi signals, the authors have demonstrated to be able to distinguish walking pedestrians from those waiting in the sidewalks in the proximity of a pedestrian crossing. They are also able to estimate the exact position for people that are waiting to cross the street. All the previous solutions are based on the analysis of Wi-Fi Probe Requests analysis, which are configuration frames in the Wi-Fi communication setup. It is clear that the Probe Requests contain an a lot of information; an example is the Preferred Network List (PNL), i.e. the SSID of the Access Points known by the device and its MAC address, even if it is an information less and less used [4]. In spite of the lack of information related to PNL, there are papers that show how the Probe Requests information can be used to create traces of mobility in order to estimate density and flows within cities. In this context, a good example is provided by a feature of the system People Mobility Analytics built in [5], a real-world system for people monitoring based on WiFi probes. The information used to obtain this data is considered, according to the GDPR, as personal identification information (PII) since it can be used to identify a specific person's movements. This appear to be the reason why some manufacturer started to implement MAC address randomization. However, although this approach is commendable in the present work we demonstrate that is not enough to provide adequate users' privacy. In the next subsection, we provide a brief description of the Probe Request frame and the Information Elements (IEs) which are used to send important information from device to the Access Points, for instance the client hardware capabilities. In order to test our algorithm we have collected 83127 Probe Request packets in 3 different scenario as figured out in tab. II . Exploring better the dataset we could notice which a subset of them is composed of packets with randomly generated MAC address as explained in section IV-A. Based on a particular data cleaning pipeline, we show how it is possible and easily feasible extract good features which allow us to count and track people exploiting WiFi standard weaknesses. The major contributions of this paper are the followings: the design and implementation of a pipeline to WiFi probes data cleaning and analysis; the implementation of a clustering model in order to find the real number of devices which are generating Probe Requests in a not crowded environment; the definition of a system to dynamically extract features in order to create a device signature, which is the novelty with respect to the state of the art. In section II, we present the structure of MAC address and Probe Request with its Information Elements (IEs), subsequently a brief analysis of the state of art on the MAC address randomization process and the related works in de-ranomization techniques; in section III we describe how the proposed algorithm works with the major functionalities of data analysis; section IV presents the experimental results; section V draws final conclusions.

978-1-7281-6339-0/20/$31.00 ©2020 IEEE

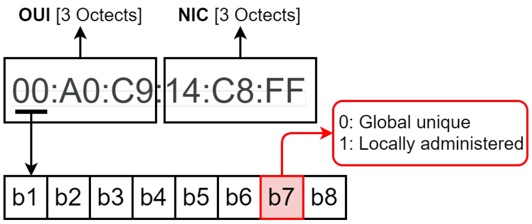

Figure 1: Global unique and Locally administered bit detail of a MAC address.

II. Background

A. MAC structure, Probe request and Information Element

Each WiFi device has a MAC address that uniquely identifies itself in the local network. Fig. 1 shows the structure of the MAC address, which is composed of six octets of bits. The first three octets are assigned by the IEEE to the device manufacturer and constitute the Organization Unique Identifier (OUI). The remaining three octets are called Network Interface Controller (NIC) and are assigned by the manufacturer. An interesting part concerns the second least bit in the first octet, highlighted in red in the fig. 1. In WiFi applications if this is set to 0, then the MAC address should be globally unique and it is kept constant over the time. Otherwise, when this bit is set to 1, the MAC address should be locally administered; as a consequence the MAC address is randomly generated and may change from one session to another, i.e. we can consider it such as a virtual MAC address. A device that wishes to know which WiFi networks there are in the its environs, sends a particular message which is called Probe Request. Specifically, it sends a burst of these messages with associated a time limit within which it must receive reply to connect to the network. Every Access Points (APs) that receive these frames within the established time replies to the device by sending, in its turn, a Probe Response Frame with the information necessary to establish the connection. The Probe Request and the Probe Response are two sub-types of a particular frame called Management Frame which is divided into MAC Header and Frame Body. Fields within the frame body could have fixed lengths, called fixed fields, or variable lengths called Information Elements (IEs). All IEs are labelled with an identification number and its size; the structure for each IE is defined by the standard[1]. The first octet of the Information Element is reserved for the Element ID, the second defines the whole information element's length and the remaining bits contain the information. Accordingly, each IE conveyed in the Probe Request Frame is identified by its ID and its length, which indicates the number of octets used by the IE content.

B. MAC Randomization

Randomization of MAC addresses is the process of generating virtual MAC addresses by end-devices during the phase of active scanning for Access Point in the WiFi context. Such activity is designed and performed to guarantee devices that their real MAC address remain unknown, preventing tracking issues. When the AP and the device find themselves, they set up the connection and only after that, the device uses its real MAC address due the fact which only starting from that moment the entire communication is encrypted. In detail, the MAC address randomization process is managed by the device Operating System (OS); there are no standard algorithms for this process, and each OS implements its custom randomization operations.
Linux OS introduced the MAC address randomization starting from version 3.18 of its kernel [6]. Most of WiFi drivers are configured to change the MAC address every 60 seconds [7]. However, the Linux OS has three methodologies for assigning a MAC address: use the real MAC address, use a completely virtual MAC address or use a partially virtual MAC address keeping the first three octets equal to the real OUI of the network card manufacturer.
Windows OS use another type of randomization, where support for randomization of the MAC address was indeed one of the most important innovations regarding wireless networking at the time it was announced in 2015[2]. The effort done in [8] explains how randomization works on Windows 10, which shows that random MAC addresses are used in the Probes Requests and also during the Authorisation and Association phases before the network connection.
Google introduced the randomization of the MAC addresses starting with version 6 of Android OS; in version 8 is enabled by default in every device[3]. Google's implementation uses a set of fake MAC addresses for network discovery.
Apple introduced the MAC address randomization in iOS OS since version 8. Based on laboratory experiments with various models, iOS devices randomize the MAC address for each burst of Probes Requests (generally composed by 2-5 probes). The level of privacy guaranteed in this case is very high. In general, as shown, each manufacturer implements proprietary algorithms; the consequence is a high variability of the MAC addresses randomization process.

C. Related Works

In the literature there are many examples of solutions that analyse WiFi traffic to people counting and device tracking, this work was inspired by what was done in [9] and [5]. Most of those solutions are out of date and do not take into account the effects of MAC address randomization or partially manage this aspect. Over time, the diffusion of MAC address randomization adopted by manufacturers has exponentially increased. Therefore, it is necessary to introduce techniques to counteract the phenomenon of randomization.

1) Fingerprinting

In [7] the author proposed a way to fingerprint the probe requests sent by a single device. The device's fingerprint is calculated using the Information Elements (IEs) contained within the probe requests. Inter-burst times were also added to this information. This allowed to obtain a more accurate analysis. Subsequently, the similarity between the various fingerprints is calculated and analyzed. In order to test the tracking tool through fingerprints, datasets of scans made at different times were used. To be precise, two datasets created by the authors and a public dataset of the Sapienza (last update 2013-09-10). Finally the clusters, which identify the single fingerprint, were calculated using the K-Nearby Neighbors (KNN) algorithm.

[1] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7786995

[2] https://channel9.msdn.com/Events/WinHEC/2015/WHT201

[3] https://source.android.com/devices/tech/connect/wifi-mac-randomization

Each fingerprint correspond at one device. The results showed that they were able to count the detected devices with an accuracy of 75%. The step forward for this work is to consider context-dependent IEs in a adaptive way as shown in this paper. In [10], a system has been developed in order to counting wireless devices. A fingerprint is created using the inter-frame time of the probe requests. The fingerprint is a unique identifier calculated using algorithms and uniquely represents the device. Franklin et al. [11] proposed a system that analyzes the inter-frame time of packets allowing the creation of a device driver fingerprint. Any of the approaches introduced above have the same main problem. Use timing as feature could add errors in clustering and classification, timing information is not reliable due to scattering problems and multi-path phenomena that occur in real-world environments because those phenomena introduce some random delays between probes and bursts. In our work we avoid to use timing as primary feature, discarding inter-burst and inter-frame times but taking into account the IEs content and how it changes accordingly to the context in which it belongs to.

2) *Active Sniffing Methods*

Several authors have proposed or carried out tests using active sniffing methodologies.
In [12], the authors have analyzed various actives sniffing techniques in order to track smartphone reducing the error introduced by the virtual MAC addresses on over 170 thousand MAC address, generating over 8 million probe request. However, the approach that has given the best results uses the parameters related to WiFi Protected Setup (WPS) for the devices that support it. A parameter called Universally Unique Identifier-Enrollee (UUID-E) has been found to derive directly from the device's real MAC address, which is the MAC address assigned by the manufacturer. However, the same work shown that a device's WPS UUID-E can often be used to derive its real MAC from the randomized MAC address. Martin et al. [13] analyze various techniques that can be used on a large scale to be able to trace random MAC addresses to a single device. In particular, active sniffing methods exploit various vulnerability and made attacks such as KARMA attack [9][12] (creation of fake Access Point from the list of probe BSSIDs) and RTS/CTS attack [12] in order to obtain the true MAC address during the negotiation of the connection with an AP, this approach requires the device's known SSID as knowledge of the attacker.

3) *Physical Layer Analysis*

Other research has applied concepts of fingerprinting also to the ISO/OSI physical layer. In [14] Brik et al. they propose a technique that is able to identify the origin interface of an 802.11 frame by performing a passive analysis of radio frequencies. In particular, machine learning tools are used in order to have an accuracy of 99% for devices counting, but this approach is good only in a laboratory environment and it is useless if applied to a real world environment because of the high radio interference and the needing of a complicated setup to collect data.

III. Proposed Algorithm

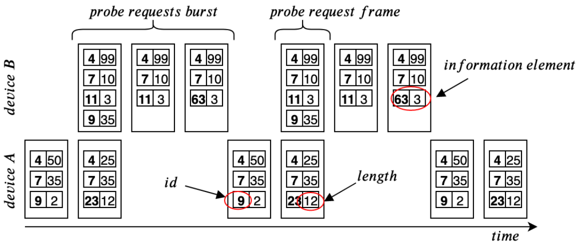

Figure 2: Probe requests burst and their information elements sent by two devices over time.

The proposed algorithm is based on the analysis of the probe requests sent by the WiFi devices, with particular attention to the information elements (IEs) and the lengths of the information (LENs) which are taken in account as algorithm's inputs. IEs within the probe request are not mandatory but they are sent because they are necessary to explain which functionalities are supported by the device itself [12]. This work has inspired our paper and in our experiments we notice which each device could send them all or only some. These differences seem to depend on the choices of the manufacturer of the wireless network card and the logic of implementation of the operating system. Therefore, our experiments have shown that the IEs are generated in an pseudo-univocal way for each device, with little variations in time. Variations may still happen according to the context in which the devices are located. In order to clarify these concepts, Fig. 2 shows how the length of information elements changes over time. Taking into account those inputs was already done in other works [7, 12], however they do not manged the IEs in a dynamic way so they may not consider the reserved Information Element IDs which are generally manufacturer-dependent. Instead, our proposed algorithm works in adaptive way, so it can recognise which Information Elements ID are most frequent and with "enough variability", this concept will be more clear in the following. However, in order to obtain the right output from the algorithm, the the raw data must be conditioned. In the following subsection, we firstly describe the procedures which we perform to clear and pre-process the data to make it ready for the algorithm; secondly, we present the proposed clustering based on the Information Element IDs and Lengths to estimate the real number of devices generating all the data collected. This pipeline flow is shown in Fig. 3 and detailed description is provided in III-A.

Table I: Most frequent Element IDs

Element ID	# Packets	Meaning
50	1436	Extended Support Rates and BSS Membership
45	933	HT Capabilities
127	723	Extended Capabilities
3	590	DSSS Parameter Set
191	89	VHT Capabilities
238	24	Reserved
128	15	Reserved

A. *Data cleaning and preparation*

The data cleaning and preparation is the most important step of all Data Analysis processes, therefore this step is done as first. Indeed, the collected raw data still needs a preliminary checks for errors, deletion of unnecessary information, identification and eventually imputation of missing values. Initially, all the corrupted packets are discarded to avoid the introduction of errors in the following operations. Then, the first important filter is applied to divide the flow of packets into those that contain random MAC addresses and the remaining ones. This filter returns two data subsets: one contains all the packets sent by devices that are sending they real MAC address and the one that is composed by all the packets sent by devices that are implementing the MAC address randomization. As explained in section II-A, by checking the second least significant bit in the first octet of the MAC address allow us to start performing this separation. At this point, it is already easy to count how many devices with real MAC address are present. The challenge is now to extract the number of devices that are generating all the MAC address of the packets belonging to the second subset. To obtain the needed insights and perform the clustering operations, it is necessary to extract and to order the mentioned features, i.e., the IE IDs and the LENs values for each packet in the random MAC addresses subset. Each packet is then deeply examined, by extracting the MAC, time, IDs and LENs, generating a Data Frame. The resulting table give us a first overview on the data regarding all the packets sent by the random MAC addresses, but it is necessary to have a better view to continue the analysis. The resulting sparse matrix is then converted into a dense matrix: here for each MAC address, the values of the LENs for each IDs are grouped. This view allows us to have a kind of signature for each MAC address and this could be useful to cluster the different addresses that come from the same device because the base theory is that sign is the same or similar.

$$E_{ies} = e_{i,j} \in R^{m \times n} \tag{1}$$

Where m is the number of different MAC addresses present in the whole dataset and n is the number of different IDs founded. The matrix that come from this step, it is huge enough to create confusion and not all the features are obviously significant to discriminate the different devices. For this reason, the next step is to choose the features to take in account. A previous analysis has been done, it concerns the features variability, in detail all the columns of the 1 are analyzed and the unique values for each ID are counted.

Let is defined IDS as the vector of Information Elements IDs founded in the previous step:

$$IDS \in R^{n} \tag{2}$$

In order to reduce the dimension of the vector IDS, we have introduced the threshold parameter α. Its value represents the minimum unique values that the column j of matrix E_{ies} needs to have to be taken in account. Changing α we can change the dimension of IDS vector simply applying this rule:

$$IDS = \{unique(E_{:,j}) > \alpha\} \;\; \forall j \in E_{ies} \tag{3}$$

After this filter is applied to E_{ies} matrix we have into the IDS vector only the columns of E_{ies} which have at least α different values. The best α is computed by an iterative method where 1 is considered as starting value and the stop condition is set to:

$$dim(IDS_k) - dim(IDS_{k+1}) \leq 2 \tag{4}$$

Where k is the current value of α. To make it more concrete, Fig. 4 shows the characteristic behaviour of the Information Element IDS dimension, which we have considered as features, on changing the threshold α. To check the validity of our solution we compared the IEs resulting by our test with the IEs identified by Vanhoef et al.[12] Consider that some differences due to the time of the study are present; an example is the preferred network list that is an information that is less and less available in probe requests, in fact the SSID field is now almost always empty[4]. However, if IDS_k dimension is less than the group of IEs mentioned above, the algorithm takes in account the IEs identified in tab. I.

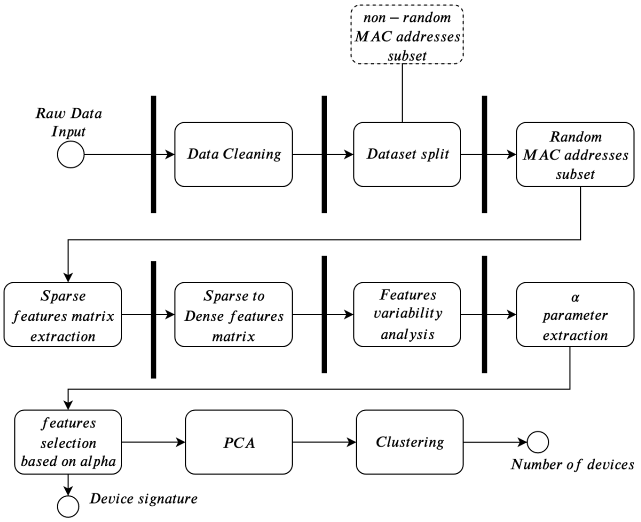
non − random MAC addresses subset
Raw Data Input
Data Cleaning
Dataset split
Random MAC addresses subset
Sparse features matrix extraction
Sparse to Dense features matrix
Features variability analysis
α parameter extraction
features selection based on alpha
PCA
Clustering
Number of devices
Device signature

Figure 3: Data Analysis pipeline

In all the tests which we have done the number of features IDS_k reach the stop condition, written in the inequality 4, using a α value between 5 and 10, therefore in this way it is possible reduce the features number to the first value below 15. This reduction, allow to continue the procedure with an higher knowledge on the behaviour of the different devices. Indeed reducing the features number in this way and taking a look at the remaining, it appears correct and intuitive because they have a physical meaning. Table I shown the most frequent IDs present in the whole dataset, the related information are specific from the device that is sending the probe. In the end, all the data prepared and conditioned are passed to the clustering algorithms.

B. Density-based Clustering Modelling

There are several types of clustering algorithms in literature, density-based approaches rely on the amount of points which are within a predefined radius in the features space. They have the advantage of being able to create arbitrary clusters and if properly configured they enjoy good scalability. Among the density-based algorithms we find, there is a well-known algorithm that is called Density Based Spatial Clustering of Application with Noise (DBSCAN) [15]. However, there are also other alternatives such as Fuzzy Joint Points (FJP) [16], and Noise-Robust Fuzzy Joint Points (NRFJP), finally we mention the successor of DBSCAN, also known as OPTICS: Ordering Points To Identify the Clustering Structure [17]. Density-based clusters are defined in the features space as variable density areas separated each other by more rarefied areas. The idea could be explained introducing the definition of *core-points*, *density-reachable points*, *density-connected* and *outliers* or *noise*. In order to define a core point, its neighbourhood of radius ε has to contain at least $MinPts$ points, i.e. we can say which the points density in the neighbourhood of p has to cross a threshold so that a point p can be defined a *core-point*. Furthermore, a point u is defined a *directly-density-reachable point* if it is in the neighbourhood of p. However, a point could be only *density-reachable* if there is a transitive closure of direct density-reachability. Finally, outliers are defined as the set of points in the dataset which not belonging to any cluster.

Once the previous definitions are clarified we can define the concept of cluster in both DBSCAN and OPTICS algorithms highlighting the main differences, because they are the algorithm that we have used in this study.

DBSCAN: once ε and $MinPts$ are given the clusters' density is defined and is not possible to change it during the clustering process.

OPTICS: it is based on the principles of DBSCAN and follows all its definitions. However, as first step the patterns are ordered such that spatially closest points become neighbours in the final ordering, subsequently the additional definitions are applied in order to find clusters with different densities [17], for simplicity in the following we have defined the parameter χ_i to identify the determines the minimum value of steepness on the OPTICS reachability plot which allows the algorithm to constitutes a cluster boundary.

IV. Results

A. Data collection and dataset characterisation

In this section we provide an overview of the datasets used and how the collections have been done. To test and validate the proposed algorithms was necessary to collect enough data to have different brands and operating systems. The acquisitions have been done using a laptop with Ubuntu OS, Wireshark and an external WiFi antenna set in monitor mode.

The data acquisition has been done in three different kind of scenarios, obtaining data saved in three different dataset.

Laboratory scenario: the dataset has been done inside a semianechoic chamber considering 23 devices;

Controlled scenario: were acquired during a workshop at university of Cagliari, into a room with 30 people with smartphone turned on and without notebook or smartwatch;

Real-world scenario: were acquired inside the university campus where the number of present people was counted by human and was 45. However, in this situation is not possible to take into account if all the people whom was present had only one WiFi device turned on.

Scenario	Probe Pkts	Virtual MAC add.es	Real MAC add.es
Laboratory	15151	181	8
Controlled	23665	179	28
Real-world	44311	1508	162

Table II: Details of data-sets

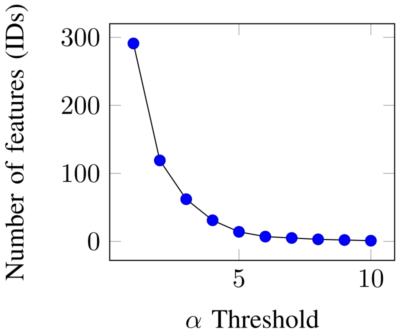

Figure 4: Number of total features taken into account changing the threshold α

The collections has a mean duration of 30 minutes. In the laboratory dataset we gathered data from 23 devices as shown in tab. III, in this scenario we were able to provide a detailed description of each device.

The controlled scenario dataset, in the first analysis provided 28 real MAC addresses, however only 12 of them have been taken into account and the other 16 have been eliminated. The discarded MAC addresses had a received signal strength indicator (RSSI) lower than -54 dBm. That value has been chosen as threshold based on the mode of RSSI distribution. It highlights how those MAC addresses were associated to device outside the area of interest. Actually this is the very first data filter in the Data Cleaning module.

A similar cleaning approach has been done to the real-world dataset. In this scenario is also possible to collect data provided by people whom were walking or driving outside the acquisition area. For this reason it is fundamental to make a good data cleaning in order to avoid over-counting situation due to the near road where some probe request could be collected. Even if we didn't have the mobile details for each scenario presented, the acquired data has been really useful to check and compare the IEs behaviour inside the probe request.

Table III: Devices under study

BRAND	# of devices	OSs
Huawei	5	Android 5.1(1)/6(1)/8(2)/9(1)
Samsung	6	Android 4.2(1)/7(1)/8(1)/9(3)
ZTE	1	Android 8
Xiaomi	4	Android 6.1(1)/9(3)
Honor	1	Android 9.1
Apple	5	iOS 12.4.4(1)/13.3(4)
Motorola	1	Android 7

B. Clustering results

As explained in section III-A after the splitting in two subset, the algorithm takes in account only the packets where the address was generated randomly. After the transformation from sparse to dense matrix, the algorithm evaluates the feature variability (shown in fig.4) and select the α value to obtain less than 10 features to take in account. After that, the resulting values are sent to a PCA algorithm to reduce the feature space to three dimensions and then the features were sent to the clustering part of the algorithm. Following the clustering theory to obtain the most affordable number

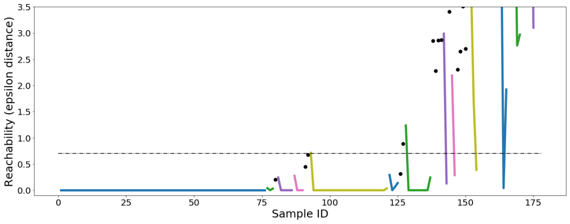

Figure 5: Reachability plot

of cluster, it is primary to choose the correct χ_i parameter. The proposed algorithm iterates χ_i between 0 and 1 with a step of 0.01 calculating the resulting number of clusters. Let is define $N(\chi_i)$ as the number of cluster given χ_i, the algorithm chooses the max value of frequency in the histogram of $N(\chi_i)$ distribution. Once χ_i is computed the output of OPTICS clustering is 13 as shown in fig. 5, where the line with different colours represents the clusters. In the same figure it is represented also the threshold ε used for DBSCAN, obtained with the same procedure used for χ_i and which value is 0.7; with these conditions the algorithm returns 8 clusters. To evaluate the accuracy of the algorithm developed using the two different clustering methodologies (DBSCAN and OPTICS) we calculated the number of devices identified by our algorithm over the real number of devices present in the testing chamber. Using DBSCAN the recognized devices are 7 that added to the same 8 non random devices returns a total of 15 over 23 real devices, obtaining an accuracy of 65.2%. Instead, using OPTICS the recognized devices are 13 that added to 8 non random devices returns a total of 21 over 23 real devices, obtaining an accuracy of 91.3%.

V. Conclusion

In this study we have addressed the fundamental topic of MAC address randomization weakness due to the IEEE 802.11 standard vulnerabilities. We designed, implemented ed evaluated an adaptive algorithm which creates a signature based on the Information Elements (IEs) contained in WiFi Probe Requests collected in not crowded environment. The signature is used to count the real number of devices which are present near the data collection station, furthermore it is possible to use the same signature to track a device among different station in order to extract mobility pattern in smart cities context. In our vision MAC address randomization is still problematic because is not enough in order to completely protect users' privacy. It also a problem for networks efficiency by worsening the user experience of non-AP WiFi stations, as highlighted by IEEE which created the study group on Random and Changing MAC addresses (RCM) [4] in order to modify the standard. An idea for the future works is to implement the algorithm in a very crowded context and evaluate the performance of both data analysis pipeline and clustering process.

References

[1] D. Goodin. "No, this isn'ta scene from Minority Report. This trash can is stalking you". In: *Ars Technica* (2013).

[2] F. Potortì et al. "Wi-Fi probes as digital crumbs for crowd localisation". In: *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 2016, pp. 1–8.

[3] A. Guillén-Pérez and M. D. Cano Baños. "A WiFi-based method to count and locate pedestrians in urban traffic scenarios". In: *2018 14th WiMob)*. 2018, pp. 123–130.

[4] A. Dagelić, T. Perković, and M. Čagalj. "Location Privacy and Changes in WiFi Probe Request Based Connection Protocols Usage Through Years". In: *2019 4th International Conference on Smart and Sustainable Technologies (SpliTech)*. 2019, pp. 1–5.

[5] M. Uras et al. "PmA: A real-world system for people mobility monitoring and analysis based on Wi-Fi probes". In: *Journal of Cleaner Production* (2020), p. 122084.

[6] E. Grumbach. *iwlwi: mvm: support random MAC address for scanning*. Linux commit effd05ac479b.

[7] C. Matte et al. "Defeating MAC Address Randomization Through Timing Attacks". In: 2016, pp. 15–20.

[8] *van Hoef, M.: How MAC Address Randomization Works on Windows 10 (2016).* http://www.mathyvanhoef.com/2016/. Accessed: 27 Jan 2020.

[9] M. Uras, R. Cossu, and L. Atzori. "PmA: a solution for people mobility monitoring and analysis based on WiFi probes". In: *2019 4th International Conference on Smart and Sustainable Technologies (SpliTech)*. 2019, pp. 1–6.

[10] D. Choong et al. "Identifying unique devices through wireless fingerprinting". In: 2008, pp. 46–55.

[11] J. Franklin et al. "Passive Data Link Layer 802.11 Wireless Device Driver Fingerprinting". In: (2006).

[12] M. Vanhoef et al. "Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms". In: *ASIA CCS '16*. 2016.

[13] J. Martin et al. "A Study of MAC Address Randomization in Mobile Devices and When it Fails". In: *Proceedings on Privacy Enhancing Technologies* 2017 (2017).

[14] V. Brik et al. "Wireless device identification with radiometric signatures". In: 2008, pp. 116–127.

[15] M. Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *Kdd*. Vol. 96. 34. 1996, pp. 226–231.

[16] E. Nasibov. "An alternative fuzzy-hierarchical approach to cluster analysis". In: *Proc. 7th Int. Conf. on Application of Fuzzy Systems and Soft Computing, Germany*. 2006, pp. 113–123.

[17] M. Ankerst et al. "OPTICS: ordering points to identify the clustering structure". In: *ACM Sigmod record* 28.2 (1999), pp. 49–60.

[4] https://mentor.ieee.org/802.11/documents?is_dcn=DCN\%2C\%20Title\%2C\%20Author\%20or\%20Affiliation&is_group=0rcm