POLITECNICO DI TORINO
Repository ISTITUZIONALE

IoT-based Mobility Tracking for Smart City Applications

Original
IoT-based Mobility Tracking for Smart City Applications / Gebru, K., Casetti, C.E., Chiasserini, C.F., Giaccone, P.. - STAMPA. - (2020), pp. 326-330. (EuCNC 2020 Dubrovnik (Croatia) June 2020) [10.1109/EuCNC48522.2020.9200941].

Availability:
This version is available at: 11583/2815294 since: 2021-03-15T12:13:26Z

Publisher:
IEEE

Published
DOI:10.1109/EuCNC48522.2020.9200941

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

08 October 2026

IoT-based Mobility Tracking for Smart City Applications

Kalkidan Gebru, Claudio Casetti, Carla Fabiana Chiasserini, Paolo Giaccone
Politecnico di Torino and CNIT, Italy

***Abstract*—The proliferation of IoT devices and the growing deployment of 5G networks combine to provide the perfect ecosystem for advanced smart city use cases. In this paper, we address the possibility of detecting and quantifying flows of people on city streets thanks to deployment of commercial sensors, connected to the 5G network, that capture WiFi probes transmitted by people's smartphones. We first outline the motivation and challenges of such a scenario. Then, we illustrate our approach and present results derived from live measurements in a testbed deployed in the city of Turin within the 5G-EVE project. We show that we can quite accurately estimate transit flows by simply collecting anonymized MAC addresses and timestamps from smartphones of passers-by.**

I. Introduction

It is widely believed that Internet-of-Things (IoT) systems will have a momentous impact on people's everyday lives, as testified by the development of specific uses cases for upcoming 5G networks. Nowhere will this impact be more tangible than in our cities.

One of the key smart city scenarios addressed by the European 5G-EVE project [1] requires the identification and quantification of people in sensitive areas (e.g., for safety and security purposes, such as during large crowd gatherings) or in areas of transit (e.g., for the purpose of dimensioning transportation networks or transit/parking/sheltering infrastructure, etc.). While the detection of presence and head count is important, more valuable information would stem from the identification of flows of people. Cameras can be used for this purpose, although they require a high upfront investment, resource-consuming detection software, expensive maintenance, not to mention the privacy concerns they usually raise.

Alternative solutions exist, such as sensors that scan the WiFi bands and passively capture probes transmitted by smartphones as they try to identify known nearby WiFi Access Points (APs). However, these sensors have some limitations: (i) they only detect people who carry a smartphone (although it can be argued that this is now the majority of passers-by); (ii) if used in a standalone mode, they only quantify the presence of people, not the path they are following; (iii) the information they expose is non-customizable and it is largely affected by implementation nuances in WiFi probe timing, hence a considerable amount of inference is required.

In this work, we address the above concerns, presenting a framework that uses data collected by commercial WiFi probe-detection sensors, henceforth referred to as "scanners", and infers flow densities and direction of transit of people on city streets. As mentioned above, inference techniques have to content with the implementation uncertainties and partiality of information exposed by commercial scanners. For this reason, we engaged in a measurement campaign in a real testbed scenario, realized within the 5G-EVE project that allowed us to establish a ground truth on which to test our framework.

The rest of the paper is organized as follows: in Section II we present the scenario and challenges of our testbed. Next, in Section III, we provide some insight on how data are collected and a preliminary assessment of mobility detection, while the algorithms used for a more in-depth inference process are detailed in Section IV, along with some sample results. A discussion of related work in Section V and of future work in Section VI concludes the paper.

II. IoT scenario and challenges

We consider an area characterized by a large inflow and outflow of people, namely the streets surrounding the campus of Politecnico di Torino, near one of the city's main railway stations and a subway station. Thousands of people transit through these streets on foot or by bike on their way into or out of the campus. In collaboration with TIM, the main Italian mobile operator, we have installed two WiFi probe-detection scanners, Meshliums by Libellium [2], on campus premises. Four more are installed in nearby streets, with the additional support of the City of Turin.

Each scanner is connected to the 5G-EVE platform, specifically to the Turin site, and, through it, data are made available on the OneM2M platform [3], as shown in Fig. 1. It is indeed our long-term goal to devise a flow detection methodology that can be later deployed as a Network Function on a suitable part of the 5G-EVE architecture, e.g., on an edge cloud or MEC.

Scanning the WiFi bands at 2.4 and 5 GHz, the scanners receive WiFi probes transmitted by nearby smartphones and store information that can be extracted from such probes, namely the timestamp and the (hashed) MAC address of the sender. Probes are nominally transmitted by all mobile devices

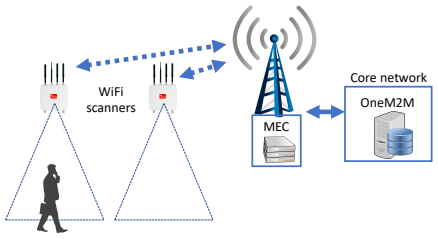

Fig. 1. 5G EVE architecture supporting the testbed

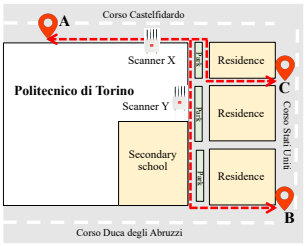

Fig. 2. Map of the roads covered by the testbed showing the position of the WiFi scanners (X and Y) and of the 3 starting/ending places (A, B, C) for the 4 paths under consideration

with a WiFi interface and they serve the purpose of quickly identifying the presence of a nearby AP that they may have previously already associated to.

The *challenges* that our methodology has to face are manifold.

- First of all, as explained below, we have to contend with the likelihood that MAC addresses in WiFi probes are randomized by the source, which complicates the counting process since the same device is seen by either scanner as having different identities. As discussed in Section V, past works have addressed the problem of identifying a device despite the randomization, but all of them require to collect the complete MAC headers, whose information is typically not available in off-the-shelf WiFi scanners, as the ones considered in our testbed.
- Additionally, the process of probe request generation is not specified by the IEEE 802.11 standard, and thus the inter-generation time of probe requests depends on the vendor [4]. Past works listed in Section V assume to receive frequent samples from mobility sensors (e.g., one sample every second), but they cannot be often applied in practical cases since off-the-shelf WiFi scanners like the ones considered in our experiments provide samples at a very coarse timescale (i.e., one sample every minute).
- Lastly, the deployment of scanners is not always functional to optimizing the coverage of an area or to facilitating the operation of our methodology. Driven by the need to adapt to the existing infrastructure (power outlets, posts, walls that may hinder the reception...), the resulting coverage may be either spotty, with uncovered portions of sidewalks or streets, or overlapping. In the latter case, especially if the timescale of probe collection is coarse, it results in one or more probes being detected by multiple scanners at the same time.

III. Testbed and Data Collection

In this section, we describe the testbed and some preliminary evaluation we ran on the probes we collected. We used just 2 WiFi scanners, leaving as future work the extension of our study to larger values of scanners. The location of the installed WiFi scanners, labeled X and Y, and the layout of the streets near the testbed area is shown in Fig. 2. The sampled probe requests are logged in JSON format by each scanner every 50 seconds. An extract of the log file is shown in Fig. 3. Each sample comprises four fields:


```
{"data":[
   {"RSSI":"-68","Vendor":"Samsung",
               "TimeStamp":"2020-02-15 12:32:45","MAC":"B7...BA"},
   {"RSSI":"-64","Vendor":"Apple",
               "TimeStamp":"2020-02-15 12:32:45","MAC":"9E...01"},
   {"RSSI":"-86","Vendor":"Unknown",
               "TimeStamp":"2020-02-15 12:32:45","MAC":"3F...FA"}
   ]}
```


Fig. 3. Sample data message in JSON format during one sampling event

- the RSSI at the receiver: this field has not been considered in our methodology since the scanner documentation does not explain how the actual value of RSSI is evaluated (e.g., it is unclear if it is the average or the maximum) and how the RSSI of multiple probe requests received during the same sampling period is computed. Furthermore, as better explained in Section V, the RSSI has not been considered a reliable metric for mobility tracking.
- the interface vendor: this field has not been considered since it does not allow to identify uniquely the interface and in many cases is equal to "Unknown". However, it could be useful to understand if the MAC is randomized, since this practice is only implemented by some vendors.
- the sampling time: although given with the precision of one second, the *same* sampling time is reported for all probe requests observed during the same 50-second sampling period. Thus it is not possible to have a detailed timing sequence of the probe requests, making the tracking extremely challenging. Furthermore, multiple probe requests from the same device during the same sampling period are collapsed into a single sample. Finally, even if the WiFi scanners are synchronized through NTP, when the sampling time difference is smaller than 50 seconds it is not possible to be sure about the temporal sequence of events, making it harder to detect the actual direction of movement.
- the device MAC address: this value has been obtained by digesting the device MAC address through a SHA-224 function. Notably, the default hash function available in the Libellium WiFi scanner digests the MAC address together with the current time, not allowing the identification of the same MAC address at different times. To circumvent this problem, the hashing mechanism was modified in order to digest just the MAC address.

Finally, every two minutes, the scanners upload the log of the WiFi probe requests transmitted from nearby devices to the OneM2M server, using their cellular link. The OneM2M server from where we have downloaded all the sample used in our study is available through the 5G-EVE infrastructure.

In a preliminary assessment of the mobility patterns of the devices captured by the two WiFi scanners, we tried to address the following two questions: (i) What are the most common mobility patterns in the testbed area? (ii) What is the effect of MAC randomization in collecting such patterns?

We downloaded a full data trace from the 5G EVE OneM2M server corresponding to a week in October 2019. This trace comprises 195,762 distinct MAC addresses. As discussed in Section II, a single device may appear with multiple MAC addresses in the trace due to MAC randomization, thus

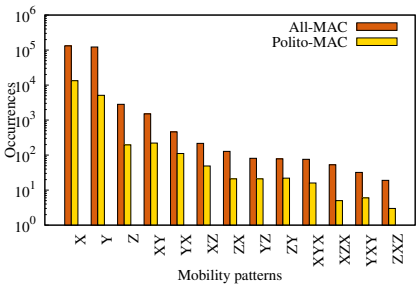

Fig. 4. Popularity of mobility patterns captured in October 2019 and collected through 5G EVE infrastructure

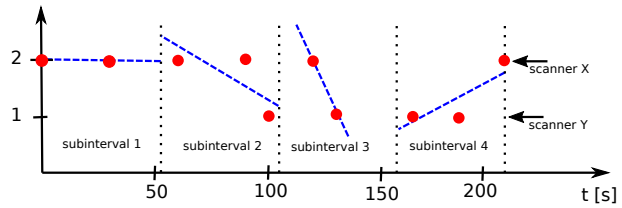

Fig. 5. The points of the path map $\gamma(t)$ in the considered toy case scenario

the number of distinct MAC addresses provides only an upper bound on the number of devices passing in the testbed area shown in Fig. 2. In order to understand the effect of MAC randomization, thanks to the collaboration of the IT Area of Politecnico di Torino, we collected a list of 34,927 MAC addresses of devices used by community members to connect to the campus WiFi network. They comprise students, professors and administrative employees. For privacy reasons, these MAC addresses were anonymized through the same SHA-224 hash function used by the WiFi scanners. This allowed us to identify the subset of MAC addresses corresponding to Politecnico users in the original trace. We remark that these MAC addresses *are not randomized*, since they are collected *after* the device has associated to one of the Politecnico APs.

We analyzed the mobility patterns shown in the whole trace. For each MAC, we computed the temporal sequence $\mathcal{T}$ of detection events that can be represented as follows: $\mathcal{T} = [(t_i, s_i)]_i$, for increasing values of t_i ($i = 0, 1, 2...$). A generic pair in $\mathcal{T}$ represents the events according to which scanner s_i detected the device at time t_i, where $s_i \in \{X, Y\}$. We partitioned $\mathcal{T}$ into subsequences by gathering all the consecutive coverage events occurring with a time difference less than 4 minutes. Each subsequence models a different mobility pattern and has been associated to a representative string to summarize the sequence of scanner identifiers. E.g., a string "X" means that the device was detected only by scanner X, instead "XYX" means that the device was under the coverage of X then under the coverage of Y and then again under the coverage of X. We used "Z" in the mobility string to denote the case in which the device was under the coverage of both scanners at the same time.

Figure 4 shows the number of occurrences in the trace of each mobility pattern string, for all the MAC addresses in the trace ("All-MAC") and just for Politecnico addresses ("Polito-MAC"). The most common mobility patterns are clearly the ones corresponding to the coverage of a single scanner (either "X" or "Y"). This result is affected by the randomization process that might change the MAC address between two detection events at disjoint scanners. Focusing just on the results for "X" and "Y" patterns for Polito-MAC, we can observe that they are still the most common patterns, suggesting that these two are the actual most popular mobility patterns for all the devices in the area. Indeed, both "X" and "Y" correspond to paths compatible with the expected main flows of people walking in the area and entering and leaving the campus area. It is also possible to notice that the popularity profiles for All-MAC and for Polito-MAC are almost identical, except for a scaling factor due to randomization and to the larger population of devices captured in All-MAC. We can conclude that the popularity profile of mobility patterns is not affected by the randomization, thus we can study the mobility patterns directly on All-MAC without considering the effect of MAC randomization.

IV. Mobility Flow Tracking Methodology

Our next step is the use of the probe patterns as they are detected by each scanner in order to pinpoint a temporal sequence $\mathcal{T}$ to actual walking paths on the streets near the scanners. The aim of our mobility tracking system is thus to associate the probes transmitted by a mobile device and detected by the WiFi scanners to the most likely path, across a given set of predefined paths that are monitored in the area. The classification is based on some preliminary experiments to build the "ground-truth" information, which yields a catalog of fingerprint vectors for each possible path. Thanks to this catalog, the sequence of probes sent by a new mobile device and detected by the scanners is compared with all known fingerprints and the path with the most similar fingerprint is associated as output of the mobility tracking, as detailed more formally in the following.

Let $\mathcal{P}$ be the set of predefined paths in the considered area to monitor and let $p \in \mathcal{P}$ be a generic path. In order to compute the fingerprint f_p of a path p, we let the scanners collect probe samples by having a person take k walks along p, carrying a device. In the following, we will refer to such a device as "ground-truth device" and to each walk along p as a "run". Consider the following toy example (assuming all times expressed in seconds):

$$\mathcal{T} = [(0, X), (30, X), (60, X), (90, X), (100, Y), \\ (120, X), (130, Y), (160, Y), (190, Y), (220, X)].$$

where $\mathcal{T}$ is as defined in Sec. III. The above expression can be interpreted in the following way: the ground-truth device was detected by scanner X at times $0, 30, 60, 90, 120, 210$ and by scanner Y at times $100, 130, 160, 190$. Note that detection events occur at multiples of 30s, i.e., periodically as in the considered off-the-shelf scanners, and the sampling events have a 10s offset between scanners. Now from $\mathcal{T}$ we compute a *path map* $\gamma(t_i) = 2$ if $s_i = X$ and $\gamma(t_i) = 1$ is $s_i = Y$. These two values have been arbitrarily chosen and do not affect at all the final classification result. Fig. 5 shows the path map for the considered toy case scenario. Let δ be the observation period, i.e., the total duration during which the ground-truth device has been detected, $\delta = \max_i\{t_i\} - \min_i\{t_i\}$. Let us now

partition the observation period into N temporal subintervals, each of duration δ/N. Notably, N is the only parameter that should be tuned according to the proposed scheme and later we will show that already $N = 4$ yields good results. In the toy example, $\delta = 210$s and each subinterval lasts 52.5s when $N = 4$.

With the above data we can now compute the fingerprint f_p. We remark that this is just one of the possible fingerprints that can be designed for path identification. The fingerprint we use is represented by a vector of $2N$ real numbers, formally $f_p \in \mathbb{R}^{2N}$. We divide such a vector in two parts:

- *coverage part*: the first N values of the fingerprint ($f_p(i)$ for $i \in \{1, \ldots, N\}$) are computed as the average of $\gamma(t)$ for each subinterval. This weighs the detection of the device from multiple scanners during the same intervals.
- *direction part*: the last N values of the fingerprint ($f_p(i)$ for $i \in \{N+1, \ldots, 2N\}$) model the mobility direction between the two scanners for each for the subintervals. Formally, it is computed as the slope of the best fitting linear interpolating function of the samples within the considered subinterval.

In the considered toy example, the subintervals would be $[0, 52.5), [52.5, 105), [105, 157.5), [157.5, 210]$ and the corresponding fingerprint would be computed as:

$$f_p = [\underbrace{2, 1.67, 1.5, 1.33}_{\text{coverage}}, \underbrace{0, -0.019, -0.1, 0.018}_{\text{direction}}]$$

Indeed, during the first subinterval the ground-truth device was detected by only by scanner X (i.e., 2) and the corresponding slope is 0. During the second subinterval, it was detected twice by X (i.e., 2) and once by Y (i.e., 1), thus the average is 1.67 and the corresponding slope is negative, suggesting that the device moved mainly from X to Y. A similar reasoning applies to the following two subintervals.

By performing many runs with the ground-truth device, a set of fingerprints is attached to each path. Thus, in order to find a match for a new device, the mobility tracking system computes its fingerprint and looks up the most similar fingerprint, using a simple Euclidean norm to evaluate the distance between vectors. In case many paths show fingerprints at a minimum distance, the path with the maximum number of minimum distance fingerprints is chosen. If still more than one path is found, the device is marked as untraceable.

A. Ground-truth experiment

We selected 4 paths in our testbed, denoting them as AB, BA, AC, CA, with the starting/ending points shown in Fig. 2. These paths have been chosen as meaningful for the typical people walking in the area and very challenging to be detected, since all the paths are partially covered by both scanners and the path direction is hard to be detected, as discussed before. We performed 17 runs for each path, walking at a slow pace, and recorded the actual time at which we started and ended each run and path. As a ground-truth device, we used a Samsung A6 smartphone with Android 9.0. We forced the smartphone to provide the list of the nearby WiFi APs every two seconds in order to force sending the probe requests. After performing all the runs, we downloaded from the OneM2M server the trace with all the logs referring to the period of interest and extracted all the data corresponding to the ground-truth device. Thanks to the temporal information of each run, we could identify the data retrieved by the WiFi scanners for each run and compute the corresponding fingerprint according to the procedure described in Sec. IV. At the end, we had the collection of 17 fingerprints for each one of the 4 paths.

TABLE I. ACCURACY OF THE MOBILITY TRACKING ALGORITHM

Path	Correctly classified test runs (%)
AB	87.5
BA	100
AC	100
CA	100

B. Experimental results for mobility tracking

We tested the mobility tracking trained with the ground-truth experiment above on 8 test runs for each of the 4 paths. Table I shows, for each path, the fraction of test runs that were correctly classified. We only experienced one incorrect classification for a test run on AB path which was classified as AC. Note that this error is due to the similarity of the two paths from the point of view of scanner Y, since in both cases the test device appears as approaching the scanner and then moving away.

V. RELATED WORK

Several works in the literature have proposed solutions to detect flows of people using WiFi probes or other techniques. The work in [5] examines the movement of passengers near a train station using the data collected from WiFi probes. The method is based on a survey to determine the fraction of people enabling WiFi interface, which is used to convert WiFi counts into an estimated number of people. The work in [6] estimates the footfall on a street covered by a WiFi sensor by clustering the number of request probes based on two thresholds: the maximum time difference between probes and the maximum difference between probe sequence numbers. In our scenario the same approach cannot be applied since the commercial scanners we use provide neither the detailed timing of the probes nor detailed information regarding the MAC header. It should be noted that the correlation between sequence numbers of the probe request in which the MAC address has been randomized can be exploited to identify a single device. The authors of [7] propose a way to distinguish the visited locations and the social behaviors for a large crowd using GPS-based methodology that, differently from our scenarios, provides detailed information of stops and movements. The classification method is based on the maximum movement duration and on the minimum stop duration. The work in [8] aims to uncover social relationships in a university by defining "semantic trajectories", which are spacial and temporal based trajectory patterns. Social closeness of users is decided according to their affinity towards some particular places (e.g., dormitories, classrooms, libraries) and the similarity of mobility patterns. Likewise, within a campus, the paper [9] investigates the flows of students between different buildings. Students inside a building are grouped based on their holding time through a clustering algorithm and the flows between buildings are characterized for each group. The work in [10]

shows the possibility to predict nationwide voting results from the SSIDs of collected probes and Wigle database [11]. To the best of our knowledge, none of the existing works has tried to track the mobility in terms of detailed trajectory as in our work with off-the-shelf sensors.

In terms of methodology for WiFi mobility detection, the WiFi probe message provides many information. For instance, the MAC address serves to uniquely identify a device in any state, i.e., stationary or mobile. For privacy reasons this field, anonymized with hash function before storing, is preferred even for monitoring vehicular mobility [12]. However, the hash based anonymization cannot ensure privacy, as reported in [13]. The MAC address is also used for counting the number of users, such as the number of arrivals and departures to/from a given area [14], and estimating the density of users [15]. Furthermore, the first three bytes of the MAC address provide information about the device vendor. As a countermeasure for user tracking, some vendors have implemented a randomization technique where a device is able to generate a random local MAC address when sending probe requests to discover access points. The papers [16], [17] provide some approaches to de-randomize the MAC address, but they are based on the assumption of knowing the randomized version of a MAC address. Thus, they cannot be applied in contexts in which the addresses are anonymized. Depending on the vendors and status of the smartphones, randomized addresses can be generated every few seconds [18]. In general, randomization tends to invalidate methods for mobility tracking especially when the mobile user is very slow and/or the path is very long with respect to how frequently the random address is generated.

When the WiFi probes are transmitted from the user devices, the WiFi scanners will save the associated timestamp. The timestamp can be leveraged to classify between stationary and mobile devices by considering the consecutive probes and the received signal strength of the device [19]. Notably, works as [6] remove stationary users (e.g., attending classes, sitting, chatting or having lunch close to the scanners) from the mobility tracking.

The reception of the WiFi probes can be used to evaluate the RSSI (Received Signal Strength Indicator) and infer the distance from the WiFi sensors, as investigated in [20]. The method is shown to be effective for indoor scenarios but not reliable for outdoor scenarios. Indeed, signal propagation and attenuation is strongly affected by the environment [21]. Multipath effects and the way how users hold the smartphone also affects the received signals and can lead to noisy signals [22]. Moreover, RSSI values show high variance already for stationary devices [6]. For all these reasons, we are not considering the RSSI information to detect the mobility in our scenarios.

VI. Conclusions

We addressed the problem of mobility tracking in an operational testbed provided by the 5G-EVE European project. We installed some off-the-shelf WiFi scanners in our campus area, capable of capturing the MAC addresses of smartphones as they send probe request messages. In the paper, we proposed a scheme to track the mobility of the smartphones carried by passers-by using data reported by these scanners, and showed that its accuracy is experimentally very high. We also documented a limited effect of the MAC randomization on such patterns.

These preliminary results are encouraging and show the feasibility of detailed mobility tracking using passive methods, despite the MAC randomization occurring at the smartphones and the coarse information made available by WiFi scanners.

Acknowledgments

The work has been supported by European Horizon 2020 Programme through the project 5G-EVE on "European 5G validation platform for extensive trials" (grant agreement n. 815074).

References

[1] European 5G validation platform for extensive trials. [Online]. Available: https://www.5g-eve.eu/

[2] Libelium Meshlium. [Online]. Available: http://www.libelium.com/products/meshlium/

[3] OneM2M. [Online]. Available: http://www.onem2m.org

[4] Y. Durmus and K. Langendoen, "WiFi authentication through social networks: A decentralized and context-aware approach," in *IEEE PERCOM*, 2014, pp. 532–538.

[5] P. Reichl, B. Oh, R. Ravitharan, and M. Stafford, "Using WiFi technologies to count passengers in real-time around rail infrastructure," in *IEEE ICIRT*, 2018, pp. 1–5.

[6] B. Soundararaj, J. Cheshire, and P. Longley, "Estimating real-time high-street footfall from Wi-Fi probe requests," *International Journal of Geographical Information Science*, vol. 34, no. 2, pp. 325–343, 2020.

[7] C. Chilipirea, C. Dobre, M. Baratchi, and M. van Steen, "Identifying movements in noisy crowd analytics data," in *IEEE MDM*, 2018, pp. 161–166.

[8] F. Wang, X. Zhu, and J. Miao, "Semantic trajectories-based social relationships discovery using WiFi monitors," *Personal and Ubiquitous Computing*, vol. 21, no. 1, pp. 85–96, 2017.

[9] E. Kalogianni, R. Sileryte, M. Lam, K. Zhou, M. Van der Ham, S. Van der Spek, and E. Verbree, "Passive WiFi monitoring of the rhythm of the campus," in *AGILE International Conference on Geographic Information Science*, 2015, pp. 1–4.

[10] A. Di Luzio, A. Mei, and J. Stefa, "Mind your probes: De-anonymization of large crowds through smartphone WiFi probe requests," in *IEEE INFOCOM*, 2016, pp. 1–9.

[11] WiGLE: Wireless network mapping. [Online]. Available: https://wigle.net/

[12] A. J. Fernández-Ares, A. M. Mora-Garcia, M. I. García-Arenas, P. García-Sánchez, G. Romero, S. M. Odeh, and P. A. Castillo, "A novel wireless mobility monitoring and tracking system: Applications for smart traffic," *IGI IJCSSA*, vol. 4, no. 2, pp. 55–71, 2016.

[13] L. Demir, M. Cunche, and C. Lauradoux, "Analysing the privacy policies of Wi-Fi trackers," in *Workshop on Physical Analytics*, 2014, pp. 39–44.

[14] A. Basalamah, "Crowd mobility analysis using WiFi sniffers," *IGI IJCSSA*, vol. 7, no. 12, pp. 374–378, 2016.

[15] U. G. Acer, G. Vanderhulst, A. Masshadi, A. Boran, C. Forlivesi, P. M. Scholl, and F. Kawsar, "Capturing personal and crowd behavior with Wi-Fi analytics," in *International Workshop on Physical Analytics*, 2016, pp. 43–48.

[16] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown, "A study of MAC address randomization in mobile devices and when it fails," *Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 365–383, 2017.

[17] J. Freudiger, "How talkative is your mobile device? an experimental study of Wi-Fi probe requests," in *ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2015, pp. 1–6.

[18] C. Matte, "Wi-Fi tracking: Fingerprinting attacks and countermeasures," Ph.D. dissertation, Université de Lyon, 2017.

[19] A. E. Redondi and M. Cesana, "Building up knowledge through passive WiFi probes," *Computer Communications*, vol. 117, pp. 1–12, 2018.

[20] G. Pipelidis, N. Tsiamitros, M. Kessner, and C. Prehofer, "HuMAn: Human movement analytics via WiFi probes," in *IEEE PERCOM*, 2019, pp. 370–372.

[21] L. Zhu, H. Tong, L. Lou, and Y. Xiong, "A passenger flow monitoring method in Hongqiao hub area based on gridded Wi-Fi sniffing," in *IEEE IMCEC*. IEEE, 2018, pp. 1052–1057.

[22] J. Weppner, B. Bischke, and P. Lukowicz, "Monitoring crowd condition in public spaces by tracking mobile consumer devices with WiFi interface," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 1363–1371.