User Trajectory Extraction Based on WiFi Scanning

Maroš Čavojský, Marek Uhlar, Marian Ivanis, Martin Molnar, Martin Drozda
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology,
Bratislava, Slovakia
Email: maros.cavojsky@stuba.sk, martin.drozda@stuba.sk

***Abstract*—We have proposed, implemented and compared several approaches for user movement (trajectory) extraction. Unlike other approaches, our approaches are purely based on WiFi sensing without the knowledge of user's physical location. This is a favorable approach in scenarios that aim at high energy efficiency. We only collect WiFi information passively, i.e. we only listen to broadcast beacons and do not transmit any probe requests. Our tests are based on 1,000+ Android mobile devices.**

I. Introduction

In order to track a (mobile device) user's location, several options can be considered:

- GPS based tracking, which is however not available in buildings or in areas with high building density. GPS is also known to be inherently energy inefficient and unsuitable for tracking that spans several days or even hours.
- Combined WiFi and GPS tracking, when GPS is applied only when location change is detected by sensing WiFi networks; see [1] for advantages and disadvantages of such a cascading approach.
- WiFi based tracking, where WiFi APs get mapped to physical (geographical) locations.
- WiFi based tracking, where the geographical location of an WiFi AP (Access Point) is unknown and its location is only evaluated with graph-theoretic measures such as number of neighboring WiFi APs or number of users that scanned this WiFi with their mobile devices (what can be mapped to node weight).

Herein we consider the last case, where only WiFi scanning is applied in order to infer a user's (relative) location. We only consider WiFi technology, however, our approach can be extended to other technologies such as Bluetooth. Tracking a user's location purely with respect to adjacency to a WiFi AP is an enabling approach for many social applications such as local broadcast (done at locations with a high user density) including location relevant media sharing or location prediction.

By collecting information on WiFi APs, organizing it and understanding its chronological continuity, it is possible to create a sequence of WiFi networks differentiated by their basic service set identifier (BSSID). Such a sequence can be used to estimate a profile of movement, or more precisely, trajectory of movement of a specific device user. There are many alternatives for how to formalize this trajectory, our approach relies on a weighted undirected graph, where the nodes represent places and the edges model users moving among places. Such a graph can be used to analyze a user's movement patterns between places with WiFi networks including periodic activities.

II. Related Work

Related research has been focused on techniques for how to predict the user's next location. A large part of related work uses trajectory extraction and prediction approaches based on position information provided by GPS [2], [3], [4], [5].

Other approaches are focused on how to recommend friends with respect to their individual historic locations [6]. Instead of using traditional trajectories in meaning of a collection of GPS points, there are also other approaches, where authors are constructing a semantic trajectory. In this way J. Ying et al. propose a novel framework by exploring semantic trajectories of mobile users, in order to predict the next location of a mobile user in support of various location-based services [7]. However, systems based on retrieving location data from GPS are only usable outside of buildings and for indoor situations they have to be used with layered architecture together with some other sort of positioning system [8].

Werner et al. proposed a multilevel architecture for indoor trajectory processing using WiFi signals as information source. They showed that it is possible to use WiFi signals in order to directly infer the trajectory out of a given collection of trajectories with high success rates [9].

Yet other research [4] is trying to predict the next location of a moving object based on the previous movements of all moving objects in a certain area without considering any information about a given user. Other authors are using WiFi logs instead of GPS and other coordinate system based predictions. Paul Y. Cao et al. analyzed WLAN logs for a human mobility predictability study across demographics (age, gender, and academic major). As a result they also verified that there is no significant difference between males and females on their long-term entropies. Their finding confirms the results by Song et al. [10]. However, when entropies of male and female students are compared on a

978-1-5386-7810-7/18/$31.00 ©2018 IEEE
DOI 10.1109/W-FiCloud.2018.00024

daily basis, females' entropies are slightly larger than males' in general [11].

Some approaches attempt to solve other problems such as semantic location using SSID of WiFi network by lexical analyzer [12]. Another view toward prediction is given by finding periodical patterns in location data [13]. Other research investigates the probability that a user moves from one location to another in various points in time [14].

III. Problem Formulation

In this section we introduce the data model used in this paper and the problem formulation.

A. Data Model

When a mobile device is turned on with WiFi adapter enabled then this device is collecting time-stamped $wifiScan$ defined as follows:

Definition 1: A *WifiScan* is a set of pairs $S = (t, W_1), (t, W_2), ..., (t, W_n)$,
where t is a time variable and W_i denotes a WiFi network W_i. WifiScan is thus a set of WiFi networks in range of device radio at time given by t. WiFi network W_i is in turn defined as follows:

Definition 2: A *WiFi network* is a tuple $W = (BSSID, SSID, frequency, level, tsf, capabilities)$,
where $SSID$ is service set identifier (also known as network name), $frequency$ is the frequency in MHz over which device is communicating with access point, $level$ is signal strength in dBm, tsf is time in microseconds for Timing Synchronization Function (TSF) specified in IEEE 802.11 and $capabilities$ records the authentication, key management, and encryption schemes supported by the access point.

B. Problem Characterization

Our two goals in this paper are: (i) extraction of trajectories of mobile devices in form of WiFi networks sequences, (ii) finding out how many WiFi networks and links between them we can drop using denoising and filtering while the information about user movement still remains preserved. We had acquired 1000+ mobile devices, that we distributed among our students, thus we have a representative data set to test our aim against real data. Our data set spans 10 months of academic year from September 2016 to July 2017.

We can divide our work in this paper into four major steps:

- *Data collection* – Overview of methods used for collection of WiFi logs recorded by Android OS devices.
- *Graph construction* – As each $WifiScan$ may contain more than one WiFi network we introduce different approaches to select WiFi network from each $WiFiScan$ to represent a node in graph.
- *Segmentation* – There are places with no WiFi coverage, user can disable WiFi adapter at any given time and/or device can be in deep sleep when not doing any scanning. This may lead to significant problems in reconstruction of user trajectory.
- *Comparison* – At the end we will compare and discuss different approaches for user trajectory extraction.

1) Data collection: In general, WiFi network scanning in Android OS can be done in the following two ways:

- *Passive* – default for Android. Mobile device listens for broadcast beacons, what can be considered energy efficient as mobile device does not need to transmit.
- *Active* – mobile device needs to tune in its radio to a particular channel and then it transmits probe request (for which it waits for about 50 milliseconds).

Due to energy efficiency considerations, we applied the passive approach. Collecting information on WiFi networks was done by scanning for WiFi networks in the range of an Android device. This is slower to perform than the active approach as mobile device needs to listen to every channel for some time period in order to detect broadcast beacons. WiFi networks periodically transmit beacons to announce the presence of a WiFi AP.

The active approach requires that probe requests get sent and this has to be repeated for each existing radio channel. After transmitting a probe request, mobile device waits about 50 milliseconds. Since the device must repeat this transmit and receive procedure for all accessible channels, it results in *higher power consumption.*

Another advantage of the passive approach is, mobile device running with Android OS can be in deep sleep. As high power usage of mobile application is a top reason why users stop using an application, we decided to rely on the passive approach when collecting information on WiFi networks. Notice also that when Android device connects to a WiFi network, the frequency of WiFi scans can dramatically decrease or stop completely.

Long pauses between two successive WiFi scans can also be caused by: (i) disabled WiFi radio and (ii) area without WiFi coverage (iii) device in deep sleep. These three situations need to be addressed when computing user movement graph as it impacts the topology of such a graph.

Yet another challenge is the varying received signal strength information (RSSI) caused by basic signal propagation phenomena such as reflection, scattering, fading or multi-path propagation. High signal strength variance can falsely imply user's movement.

2) Graph construction: Having discussed the challenges and phenomena connected with WiFi network scanning, let us further develop a formal graph model, where nodes represent WiFi APs (places) and edges represent user transitions between WiFi APs. We define *WiFiPlace* as follows:

Definition 3: A *WifiPlace* is $P = (W_n)$,
where W_n is a single WiFi network, which was chosen from $WifiScan$ by defined criteria. It can represent a place such as school, home, shopping mall or class room. Thus $wifiPlace$ is a representation of node when building

graph of user movement. The methods applied to choosing WifiPlaces are described later.

Definition 4: A *wifiTrace* is a sequence of *wifiPlaces* $T = (t_1, P_1), (t_2, P_2), ..., (t_n, P_n)$,
where for $i = 1...n$, t_i is a non-decreasing time variable that corresponds to WiFi network scan time. In our case, we interpret T as a time sequence of WiFi places identifying users movements.

Definition 5: A *wifiPath* is a sequence of pairs $H = (t_1, P_1), (t_2, P_2), ..., (t_n, P_n)$,
where for $i = 1...n$, t_i is a non-decreasing time variable that corresponds to WiFi network scan time for $WiFiPlace$ P_i, such that $\forall P_i, P_i \neq P_{i+1}$. $WifiPath$ is thus a time sequence of WiFi Places identifying users movements. Transition between subsequent pairs in $wifiPath$ corresponds to edge when building graph of user movement.

People tend to go to the same place repeatedly. Often, people's habits can be identified as a possible cause. We can loosely divide these cases into three categories:

- *Regular*: These are places visited almost every day such as home or work place, in some cases this can also be temporary accommodation such as a hotel.
- *Irregular*: These are places visited often, however, the visits are not periodic, for example bars, restaurants, shopping mall or barber.
- *Non-repeating*: When the visit period is too long, we do not have enough data to identify habits. For example, it can be mapped to visiting your cousin twice a year.

Our aim is to find out which places get visited regularly or irregularly by a user and infer a movement graph of this user. We assume that such a graph can be used for prediction of user movement.

3) Segmentation: Time difference between successive WiFi scans can be large. The main reasons and consequences of these situations could be as follow:

- *Disabled WiFi radio* – a user can disable WiFi radio at any time and enable it again whenever desired. The worst case is that he/she can disable and enabled it at the same place, but in that time period, he can go and return from another place. This behavior of user results in situation which is analyzed as remaining at the same place for the time of disabled WiFi radio.
- *No WiFi coverage*: Even nowadays there are places without coverage of any WiFi network. As a user moves to a place without WiFi coverage, this situation is similar to that with disabling WiFi radio. The user ends up with empty WiFi scans despite moving away from the current location. As in the previous case this is identified as staying at the same place.

We formalize the above two cases as *gaps* defined as follows:

Definition 6: If for any subsequent WiFi places in *wifiTrace*, (t_i, P_i) and (t_{i+1}, P_{i+1}), it holds $t_{i+1} - t_i > \beta$, then we say that between (t_i, P_i) and (t_{i+1}, P_{i+1}) is a *gap* of size β.

Experimentally we set a value of β to 60 minutes. This time was among other reasons based on Android Behavior Changes [15] that require that at least one WiFi scan should be done each hour. By adding gaps we can create graph containing several components which are not connected with each other. These components, for example, can represent places to which we traveled by plane. We think that adding $gaps$ in WiFi Trace and WiFi Path can give us better insight on relationship between user movement and static (no movement) patterns.

4) Comparison: Given the discussion and definitions above, we can formulate the problem statement investigated herein as follows:

A *user movement graph* is represented by nodes $WiFiPlaces$ and edges, which are pairs of $WifiPath$. It should have minimum number of nodes and edges while information about user movements remains preserved.

We will now introduce several approaches that generate graphs that preserve a different degree of user movement information.

IV. Experimental Results

A. Implementation Details

To achieve our objective, we created an Android application for data collection and a Java server application, which can continuously analyze incoming data from users. Results of server-side analysis are used for updating user movement graph. In this section, we describe methods and algorithms we are applying in more detail.

B. Data overview

Within a research project, we had acquired 1000+ mobile devices, that were distributed among our students. The results shown herein were collected by our students during a 10-month period from September 2016 - July 2017. Over 120 million WiFi scan records consisting of over 635,000 unique WiFi networks collected by 455 devices (a high volume subset of 1000+ devices) can provide insights about our students' behavior patterns (bars, restaurants, clubs etc.).

C. Identifying traces from WiFi scans

To start identifying traces of users we propose an approach in which we create a chronological sequence of WiFi networks by using all the $WifiScans$ collected by a specific user's device. Considering that $WifiTrace$ is a sequence of $wifiScans$, then the result should be represented as a sequence of time-stamped WiFi networks. Therefore the implemented algorithm must select a single WiFi network from each wifiScan. As choosing the right WiFi network from Wifi scan influences graph topology, we need to address several challenges in our logs:

- *Identical SSID*: Several WiFi networks can share a single SSID.

- *Repeating routes*: People tend to go to the same place repeatedly and usually follow the same route. High signal strength variance can falsely imply that user takes a different route and it result in different $WifiTrace$.
- *Repeating places*: The same place can be identified by different $WifiPlaces$ due to signal strength variance. This situations should be also considered when extracting a $WifiPlace$ from each $WifiScan$.
- *Gaps*: As already discussed, delays between successive Wifi scans can be large.

Based on these challenges we propose 6 approaches for selecting a WiFi network from each WiFi scan.

1) Maximum RSSI: In this naïve method we pick WiFi network with the highest RSSI from each WiFi Scan. This WiFi network is then used as $WifiPlace$ in $WifiTrace$. This method is shown in Algorithm 1.

```
Data: $UserData$: list of collected $wifiScan$
Result: $resultList$: user $wifiTrace$
$resultList$ := empty;
foreach $wifiScan \in userData$ do
    add maxRSSI($wifiScan$) to $resultList$;
end
return $resultList$;
```

Algorithm 1: Max RSSI algorithm

A disadvantage of this approach is that each time a user visits a place, a different WiFi network can get chosen due to varying signal strength. However, this approach is easy to interpret and a suitable choice for a base case.

2) Sticky WiFi: In this approach we pick first WiFi network with the highest RSSI from each WiFi Scan as before. The next time, a user visits the same place, this WiFi network gets chosen; see Algorithm 2.

```
Data: $UserData$: list of collected $wifiScan$
Result: $resultList$: user $wifiTrace$
$resultList$ := empty;
$currentWifi$ := null;
foreach $wifiScan \in userData$ do
    if $currentWifi$ not in $wifiScan$ then
        $currentWifi$ := null;
    end
    if $currentWifi$ is null then
        $currentWifi$ := maxRSSI($wifiScan$);
    end
    add $currentWifi$ to $resultList$;
end
return $resultList$;
```

Algorithm 2: Sticky WiFi algorithm

3) History WiFi: In this approach, if the currently chosen WiFi network disappears from WiFi scan, we first use a WiFi network from history. If this is not possible, we use the WiFi network with highest RSSI. This approach is shown in Algorithm 3.

```
Data: $UserData$: list of collected $wifiScan$
Result: $resultList$: user $wifiTrace$
$resultList$ := empty;
$currentWifi$ := null;
$historyList$ := empty;
foreach $wifiScan \in userData$ do
    if $currentWifi$ not in $wifiScan$ then
        $currentWifi$ := null;
    end
    if $currentWifi$ is null then
        $tmp$ := $wifiScan \cap historyList$;
        if $tmp$ is empty then
            $currentWifi$ := maxRSSI($wifiScan$);
            add $currentWifi$ to $historyList$;
        else
            $currentWifi$ := $tmp$;
        end
    end
    add $currentWifi$ to $resultList$;
end
return $resultList$;
```

Algorithm 3: History WiFi algorithm

4) Gaps approach: This approach takes into consideration gaps as defined in Def. 6. If two subsequent $wifiPlaces$ have time scan difference larger than β then we do not add an edge between these two $wifiPlaces$. This approach is shown in Algorithm 4.

```
Data: user $wifiTrace$
Result: $resultList$: user $wifiTrace$
$resultList$ := empty;
$previous$ := null;
foreach $actual \in wifiTrace$ do
    if $previous$ is null then
        add $actual$ to $resultList$;
    else
        $diff$ = timeDiff($previous$, $actual$);
        if $diff \geq \beta$ minutes then
            add $gap$ to $resultList$;
        end
        add $actual$ to $resultList$;
    end
    $previous$:=$actual$;
end
return $resultList$;
```

Algorithm 4: Adding Gaps algorithm

D. Basic Statistics

Tables I and II give a statistic on the collected WiFi scans, unique SSID and BSSID. We can see that about 4% WiFi

Table I
COLLECTED DATA PER WEEK

Day	WiFi scans	Unique SSID	BSSID
Mon	1,745,504	34,522	65,082
Tue	1,847,378	35,635	67,576
Wed	1,969,656	38,472	72,044
Thu	1,842,963	38,685	72,286
Fri	1,863,711	42,356	78,518
Sat	1,577,403	32,925	58,906
Sun	1,436,169	30,236	52,036

Table II
COLLECTED DATA PER MONTH

Month	WiFi scans	Unique SSID	BSSID
January	1,374,104	23,300	40,959
February	1,045,974	21,229	38,666
May	913,805	15,996	27,163
June	1,645,319	27,537	51,052
July	1591558	38,520	70,932
August	1,581,652	36,651	65,530
September	1,453,050	31,183	57,089
October	1,813,962	26,625	49,823
November	1,671,224	23,027	42,686
December	1,538,188	23,545	44,673

networks were unique, which implies that user trajectories are stable, i.e. people often move around the same APs. We can also see that SSIDs get often reused.

E. Visualization

Our goal was also to create a clear and simple visualization of user movement. In our case this task is equivalent to $wifiPath$ identification.

Our approach to visualization is quite straightforward. First, we compute a $wifiPath$ by means of the algorithms already discussed. Then we create an edge between consecutive wifiPlaces, except when there is a gap as defined in Def. 6. The initial edge weight is set to 1. If there is an edge, which already exists in graph, then edge weight is increased by 1. Node weight is calculated as the weight of incoming edges. Finally, we order nodes by their weight in non-decreasing order and keep removing nodes until the resulting graph becomes planar. To check whether the graph is planar, we used the algorithm by Di Battista and Tamassia [16].

F. Experimental results

Each algorithm introduced in our paper was implemented and data was collected with user's mobile devices. This data represents a set of chronologically recorded $WifiScans$. The three key approaches for WiFi Place selection (i) Maximum RSSI, (ii) Sticky WiFi and (iii) History WiFi were all enhanced by adding gaps to $wifiTrace$ and $wifiPath$. This gives us in total six approaches, which we were testing and comparing against our defined criteria.

When analyzing our approaches, we have chosen several representative users (devices) with the highest amount of collected data. The results for several devices including 95% confidence intervals are shown in Figures 1, 2 and 3. These figures show the number of nodes and edges that remained in the graph after the above mentioned approaches were applied.

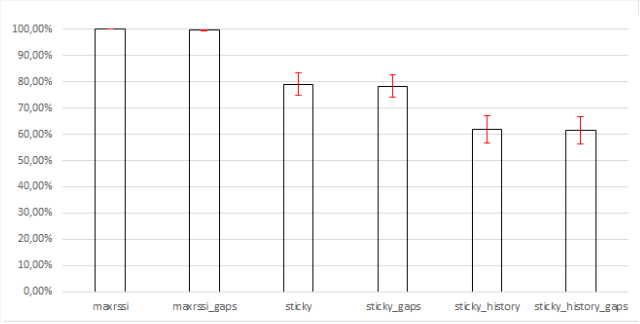

Figure 1. AVG % of remaining nodes according to each method (21 devices)

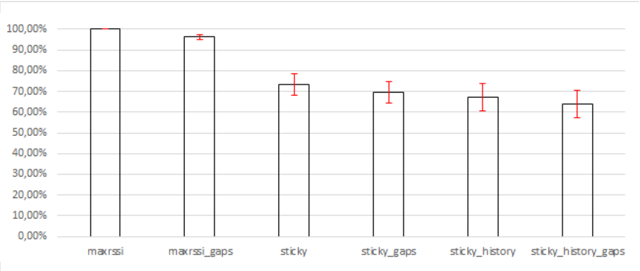

Figure 2. AVG % of remaining edges according to each method (21 devices)

Our aim was to minimize the number of nodes and edges in graph, while information about user movement remains preserved. This was done by eliminating $WifiPlace$s at the same physical place. Grouping different WifiPlaces to a single WifiPlace, represented by a physical place, helped decrease the number of edges; see Figure 2. Our results, in general, do not support that considering gaps is useful.

We have also identified $WifiPlaces$, which users visited more often and in some cases we could also calculate periodicity of these visits. In Table III are shown several $WifiPlace$s, which were periodically visited by a chosen user. The majority of users in our data set are students, so this table is dominated by eduroam, which provides a

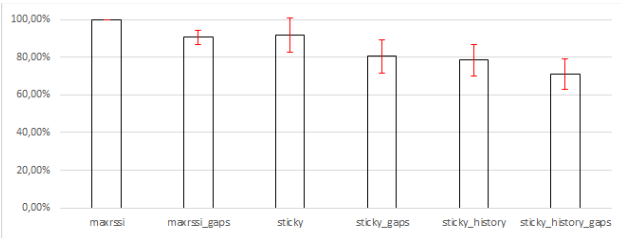

Figure 3. AVG % of maximum node degree according to each method (21 devices)

Table III
REPEATED PLACES OF USER

SSID	Num. of BSSIDs	Period
eduroam	80	≈ daily
Ynet	11	≈ daily
Aupark	6	≈ weekly
Subway	1	≈ monthly
wifiSiet	1	≈ 3 weeks

unified access to university WiFi networks across Europe and several Asian countries.

V. CONCLUSIONS AND FUTURE WORK

The aim of our approaches presented herein was to create a user movement profile based on WiFi networks represented by graph and remove any WiFi places that can be mapped to the same physical location. To achieve this aim we have considered, implemented and compared several approaches. Our experimental results show that we were able to construct a graph, representing user movement, where the number of Wifi places representing the same location is decreased. We have also considered the notion of "gaps" that represent time period in which a given device was off or unavailable (for various reasons). We could not confirm with statistical significance that considering gaps is necessary when constructing user movement graphs.

As ideas for future work we consider introduction of semantic information about Wifi Places. This will allow for removing of WiFi Places of lesser interest to users and instead including into movement graph WiFi places with a stronger historical relevance. This can be a more favorable approach than evaluating WiFi Place relevance only with respect to its membership to a physical location.

REFERENCES

[1] M. Čavojský and M. Drozda, "Energy efficient trajectory recording of mobile devices using wifi scanning," in *Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences*. IEEE, 2016, pp. 1079–1085.

[2] R. Montoliu, J. Blom, and D. Gatica-Perez, "Discovering places of interest in everyday life from smartphone data," *Multimedia tools and applications*, vol. 62, no. 1, pp. 179–207, 2013.

[3] K. Michael, A. McNamee, M. Michael, and H. Tootell, "Location-Based Intelligence – Modeling Behavior in Humans using GPS Location-Based Intelligence – Modeling Behavior in Humans using GPS Location-Based Intelligence – Modeling Behavior in Humans using GPS," in *2006 IEEE International Symposium on Technology and Society (ISTAS 2006)*. IEEE, 2006, pp. 1–8. [Online]. Available: http://ro.uow.edu.au/infopapers/386

[4] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti, "WhereNext: a Location Predictor on Trajectory Pattern Mining," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 637–646.

[5] C.-C. Hung, C.-W. Chang, and W.-C. Peng, "Mining trajectory profiles for discovering user communities," in *Proceedings of the 2009 International Workshop on Location Based Social Networks - LBSN '09*. ACM, 2009, pp. 1–8.

[6] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma, "Recommending friends and locations based on individual location history," *ACM Transactions on the Web*, vol. 5, no. 1, p. 5, 2011.

[7] J. J.-C. Ying, W.-C. Lee, T.-C. Weng, and V. S. Tseng, "Semantic trajectory mining for location prediction," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2011, pp. 34–43.

[8] T. Choudhury, M. Philipose, D. Wyatt, and J. Lester, "Towards Activity Databases: Using Sensors and Statistical Models to Summarize People's Lives," *IEEE Data Eng. Bull.*, vol. 29, no. 1, pp. 49–58, 2006.

[9] M. Werner, L. Schauer, and A. Scharf, "Reliable trajectory classification using Wi-Fi signal strength in indoor scenarios," in *Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION*. IEEE, 2014, pp. 663–670.

[10] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and POIs," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*. ACM, 2012, pp. 186–194.

[11] P. Y. Cao, G. Li, A. C. Champion, D. Xuan, S. Romig, and W. Zhao, "On Human Mobility Predictability Via WLAN Logs," 2017.

[12] S. Sobana and M. Selvi, "User Preference Profiling Through Wi-Fi Logs," *International Journal of Engineering Science and Computing*, vol. 2897, 2016. [Online]. Available: http://ijesc.org/

[13] J. Wang and B. Prabhala, "Periodicity Based Next Place Prediction. the Procedings of Mobile Data Challenge by Nokia Workshop at the Tenth International Conference on Pervasive Computing," *University of Illinois at Urbana*, 2012.

[14] S. Lee, J. Lim, J. Park, and K. Kim, "Next Place Prediction Based on Spatiotemporal Pattern Mining of Mobile Device Logs," *Sensors*, vol. 16, no. 2, p. 145, 2016.

[15] Google, "Android 8.0 Behavior Changes — Android Developers," 2017, Accessed: 25-Oct-2017. [Online]. Available: https://developer.android.com/about/versions/oreo/android-8.0-changes.html

[16] G. Di Battista and R. Tamassia, "On-line planarity testing," *SIAM Journal on Computing*, vol. 25, no. 5, pp. 956–997, 1996.