

# WiVisit: POI Visit Identification Based on Auto-Generated Wi-Fi Fingerprint

Qiang Huang<sup>1</sup>, Xiang Li<sup>1</sup>, Xin Li<sup>1</sup>, Jiazhi Ni<sup>1</sup>, Xin Zhang<sup>1</sup>, Ning Xiao<sup>1</sup>, Hongyi Liu<sup>1</sup>, Chang Liu<sup>1</sup> and Youchen Wang<sup>1,2</sup>

<sup>1</sup>Tencent Inc. Beijing, China

<sup>2</sup>School of Transportation Science and Engineering, Beihang University, Beijing, China

## Abstract

Point of Interest (POI) visit is critical information for many location-based services, such as POI recommendation and advertising push. However, most POI visit information is obtained from user's check-in data on social networks, which inevitably contains false visits and misses parts of real visits. Some work has been done to attempt mining POI visits from users' GPS trajectories, but they could not cover indoor POIs. In this paper, we proposed a Wi-Fi fingerprint-based POI visit identification system, WiVisit, in order to get accurate POI visit info, including indoor POIs. Different from traditional Wi-Fi fingerprint-based localization system, WiVisit can generate Wi-Fi fingerprints automatically with Wi-Fi and POI binding info for different POIs without any human effort. Therefore, WiVisit system could be easily and widely deployed in the real world. Moreover, a multi-model fusion based POI visit identification method was used in WiVisit to handle multiple POI types. Finally, extensive real POI visits were collected and used to assess the performance of WiVisit system. WiVisit achieved a 90% recall rate and 83% accuracy from the visited POIs, which already outperformed the state-of-the-art.

## Keywords

Wi-Fi Fingerprint, POI, Indoor Localization

## 1. Introduction

The term "Point of Interest" (POI) refers to a geographical location that someone may find interesting, useful, or visit frequently. The POI visit information of a user is very important for a lot of location-based services (LBS), such as push advertisements, next POI recommendation [1, 2, 3]. However, most POI visit information is obtained from users' check-in data on location-based social networks, such as Yelp, Foursquare, and Facebook Places. Due to that POI visit information is pushed by a user manually, there are many missing and fake POI visits. Moreover, if the user pushed the visit information when they left the POI, the best time to push ads will be missed. Therefore, we wanted to build an accurate localization system which can identify real POI visits when users arrive at the POI.

Some researchers also tried to use GPS trajectory information to identify POI visit [4, 5, 6]. However, many POIs are indoor POIs. For these POIs, the GPS signal will be blocked, and the


---

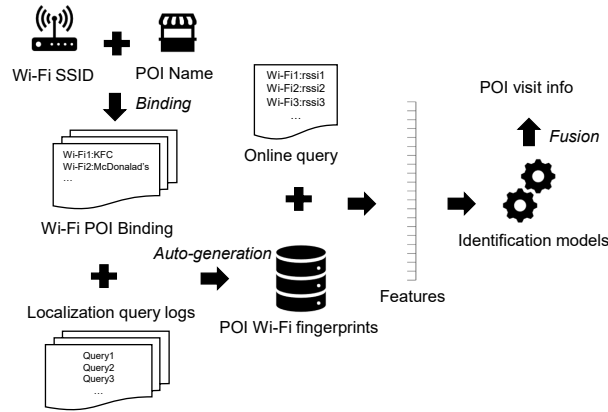
*IPIN 2021 WiP Proceedings, November 29 – December 2, 2021, Lloret de Mar, Spain*

✉ johnnhuang@tencent.com (Q. Huang); allenxli@tencent.com (X. Li); clarkxinli@tencent.com (X. Li); andyni@tencent.com (J. Ni); deanxzhang@tencent.com (X. Zhang); ariesxiao@tencent.com (N. Xiao); hongyiliu@tencent.com (H. Liu); levenliu@tencent.com (C. Liu); youchenwang@tencent.com (Y. Wang)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)



**Figure 1:** Framework of WiVisit.

GPS trajectory information will be missing as well. For that reason, utilizing GPS trajectories for indoor POI visit identification is impracticable. In recent years, Wi-Fi devices have been broadly used, and Wi-Fi fingerprint-based localization techniques have been also widely applied in both outdoor and indoor environment. Based on pre-built Wi-Fi fingerprints for each POI, a localization system [7, 8, 9, 10, 11] can be used to identify if the user is at a POI and which POI they are visiting.

However, in order to achieve this goal, there are several challenges. First, the amount of POI is colossal, and one POI may change frequently due to renovation or relocation. Thus, POI fingerprints are very difficult to collect and update manually. Second, different types of POI have diverse Wi-Fi environments; therefore, the features of fingerprints for distinct POIs will vary considerably.

In order to solve these challenges, we implemented an auto-generated Wi-Fi fingerprint-based POI visit identification system (WiVisit). Figure 1 shows the framework of WiVisit system. First, we proposed an automatic POI fingerprints collection method. The basic idea is, if the scanned Wi-Fi list in the localization query contains a POI's Wi-Fi and the RSSI is strong enough, the user is most likely to be visiting that POI. That is, they are very close to the Wi-Fi device based on the path loss model [12] at the POI. In order to obtain POI's Wi-Fi information, we designed a POI-Wi-Fi binding module based on the similarity between names of the POI and Wi-Fi. Applying the POI-Wi-Fi binding information, many in-POI localization queries will be collected automatically, of which the scanned Wi-Fi list contains a POI's Wi-Fi and the RSSI is strong enough. Then, the POI's fingerprints could be built from these queries. Secondly, based on the POIs' type information, we divided all POIs into different groups, and trained different POI visit identification models for each of them. At last, we proposed a multi-model fusion mechanism to combine these models in one. The following are our main contributions in this paper:

- We proposed a POI-Wi-Fi binding module, which could discover Wi-Fi devices that are placed in the POIs. Based on the POI-Wi-Fi binding information, the WiVisit system could automatically collect and update POI fingerprints without any human effort.

- We designed a multi-model fusion based POI visit identification method which can be used for different types of POI.
- We also conducted extensive experiments to test the performance and effectiveness of WiVisit system in the real world.

The following sections are organized as follows: Section 2 introduces the related work and Section 3 gives a formal definition of POI visit identification. Section 4 is a detail description about how to generate POI fingerprints automatically. Section 5 introduces the POI visit identification model. Section 6 is the experiments and evaluations about WiVisit, and Section 7 is the conclusion.

## **2. Related Work**

### **2.1. Indoor Localization**

Many POIs are placed in an indoor environment, such as stores, shopping malls, restaurants, and bars. Determining whether a user is visiting a POI, rather than just passing through it, can be treated as an indoor localization problem.

Over the years, many indoor localization technologies have been developed including cameras [13][14], sound [15, 16], radio frequency [7, 17, 18, 19], etc. Camera-based technologies [13, 14] can achieve a high accuracy, but the privacy concern of them is a big issue for real-life deployment. Sound-based solutions [15, 16] are vulnerable to environmental noises and the coverage is quite small. In addition, dedicated infrastructure is needed for UWB-based [19, 20, 21] and Bluetooth-based [17] solutions. Since Wi-Fi access points are deployed ubiquitously, we focused on Wi-Fi based indoor localization methods in our work.

Wi-Fi-based indoor localization has drawn a lot of attention from researchers in the last decade. RADAR [7] is a pioneering system employing Wi-Fi RSSI information as the fingerprint for localization. After which, many RSSI-based indoor fingerprint localization methods have been developed to reduce the offline training load or to improve the accuracy [8, 9, 10, 11, 22]. However, these methods still required some human efforts on fingerprint collection and updating, which is impossible for a large number of POIs. In the past few years, some fine-grained CSI-based indoor localization methods have been proposed [23, 24, 25, 26, 27, 28]. However, only a few commercial Wi-Fi chips could provide the CSI information to users. Hence, CSI-based indoor localization methods cannot be adopted widely in the real world.

Compared with these methods, WiVisit can automatically generate and update POIs' Wi-Fi fingerprints from localization queries without any human effort. Meanwhile, WiVisit system adopts RSSI information as the fingerprints, which are available on almost all existing Wi-Fi chips. Therefore, the WiVisit system can be deployed in the real world rapidly and widely.

### **2.2. POI Visit Mining from GPS Trajectory**

GPS system is widely used for outdoor localization. Consequently, in recent years, some researchers tried to mine POI visit events from users' GPS trajectory data [4, 5, 6]. They first extracted stay-points from a GPS trajectory as first, then they detected semantic locations

from these stay-points and assigned a POI visit information for each location. However, due to privacy concerns, users would not open GPS localization service all the time. For this reason, getting a complete GPS trajectory of a user for POI visit mining is very difficult in the real world. Additionally, in an indoor POI, users may not be able to receive a GPS signal, so they cannot get the GPS location information which is required for POI visit mining.

In comparison with these methods, WiVisit system does not require users' trajectories. Whenever the user needs position information, the WiVisit system can identify the POI visit based on the user's localization query. Moreover, as Wi-Fi devices are ubiquitous in the world, almost all POIs have deployed their own Wi-Fi devices that WiVisit system can use for POI visit identification.

### 3. Definition of POI Visit Identification

**Localization query:** A localization query is a user scanned Wi-Fi list that contains Wi-Fi MAC address ( $m$ ) and RSSI value ( $r$ ):

$$query = \{m_1 : r_1, m_2 : r_2, \dots, m_K : r_K\} \quad (1)$$

where  $K$  represents the number of scanned Wi-Fi.

**POI fingerprint:** A POI fingerprint ( $FP_{p_j}$ ) records Wi-Fi information in a period of localization queries of the POI ( $p_j$ ):

$$FP_{p_j} = \{m_{j1} : (r_{j1}, o_{j1}); m_{j2} : (r_{j2}, o_{j2}), \dots, m_{jN} : (r_{jN}, o_{jN})\} \quad (2)$$

where  $N$  is the number of Wi-Fis,  $r_{jn}$  is the median RSSI of the Wi-Fi  $m_{jn}$  in the POI ( $p_j$ ) localization queries, used to build fingerprints and  $o_{jn}$  is the rate of occurrence:

$$o_{jn} = \frac{|\{query | m_{jn} \in query \wedge query \in p_j\}|}{|\{query | query \in p_j\}|} \quad (3)$$

**POI visit identification:** The POI visit identification's task is to determine if a localization query is from a POI and which POI it is from, based on the scanned Wi-Fi list in the query and POIs' Wi-Fi fingerprints:

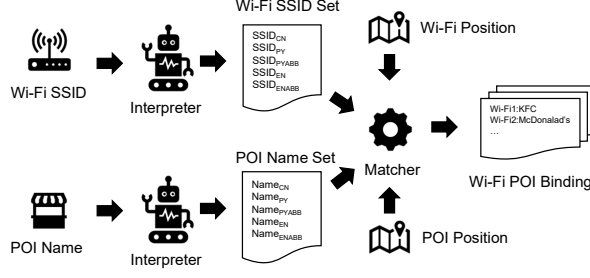
$$p = \begin{cases} \arg \max_i P(p_i | query), & P_{max} \geq \theta, \\ None, & P_{max} < \theta \end{cases} \quad (4)$$

$\theta$  is the probability threshold used to determine whether the query is in a POI.

## 4. Auto-Generated POIs' Fingerprint

### 4.1. Wi-Fi POI binding

Most of Wi-Fi fingerprint based indoor localization systems [7, 8, 9, 10, 11, 22] collect fingerprints information by human efforts. However, it is not feasible to create a mass pool of POI Wi-Fi



**Figure 2:** POI-Wi-Fi binding module. Based on the similarity among different forms of POI name, Wi-Fi SSID and their coordinate distance, a set of precise Wi-Fi POI binding pairs were created.

fingerprints manually. Therefore, we proposed an effective Wi-Fi binding method, which is the key point to build Wi-Fi fingerprints of POIs automatically.

Wi-Fi is ubiquitous and most of POIs have their own Wi-Fis. We noticed that most POIs have an SSID (Service Set Identifier, the WiFi name that users can see) similar to their POI name. For instance, in China, McDonald’s-related Wi-Fi SSIDs are mcd-chinanet or mcdonald’s. Because of this, we proposed a POI-Wi-Fi binding method, shown in Figure 2. First, for each POI name, we used a natural language processing (NLP) interpreter [29, 30] to get its’ Chinese name (CN), Chinese pinyin (PY), Chinese pinyin abbreviation (PYABB), English name (EN) and English abbreviation(ENABB), shown in equation 5. Then, a text similarity between each pair of POI and Wi-Fi was computed as following:

$$S(m, p) = \max_{lab \in \{CN, PY, PYABB, EN, ENABB\}} S(m_{lab}, p_{lab}), \quad (5)$$

where  $S(m_{lab}, p_{lab})$  is defined as the text hamming distance. Similarity score is the maximum value among different Wi-Fi POI names.

In addition to Wi-Fi SSID and POI name’s similarity, we also integrated the coordinate distance to the Wi-Fi POI matching score. It is given as:

$$G(m, p) = \begin{cases} S(m, p) - \gamma D(m, p), & D(m, p) \leq D_{th} \\ 0, & D(m, p) > D_{th} \end{cases} \quad (6)$$

where  $\gamma$  is a normalize factor between similarity and distance,  $D(m, p)$  is the distance between Wi-Fi and POI, and  $D_{th}$  is the distance threshold value to select candidate POI Wi-Fi pair sets for binding process<sup>1</sup>.

Finally, we assigned each Wi-Fi to the POI which has the maximum score greater than  $G_{th}$  among all candidate POIs. It is formulated as follows:

$$\langle p, m \rangle = \begin{cases} \operatorname{argmax}_i G(m, p_i), & G_{max} \geq G_{th} \\ \langle p_i, m \rangle & \\ None, & G_{max} < G_{th} \end{cases} \quad (7)$$

where  $G_{th}$  is the score threshold for choosing the real matched POI-Wi-Fi-pair.

<sup>1</sup>Wi-Fi positions were calculated from history localization logs, which is omitted due to out of scope of this paper. POI positions obtained from Tencent Map [31].

## 4.2. Automatic POI fingerprints generation

Applying the POI-Wi-Fi binding information, POI fingerprints can be collected automatically. Based on the path loss model [12], the further away the user is from the Wi-Fi device, the lower the power of received Wi-Fi signal is. Thus, if a user visits a POI, the scanned POI Wi-Fi's RSSI will be higher than the user outside of the POI. For a localization query, if a scanned Wi-Fi in the query is a POI-Wi-Fi and the RSSI value is high enough<sup>2</sup>, the localization query is likely to have taken place in the POI, and was used to build the POI fingerprint information. This way, based on a period of localization queries, the fingerprint information of POIs can be generated.

For WiVisit system, the POI fingerprint records all Wi-Fi that were scanned in the history localization queries at that POI. For each Wi-Fi, it contains two statistics information, median RSSI value and the rate of occurrence, which are defined in Section 3.

## 5. POI Visit Identification Model

### 5.1. Sample Extraction

To train the POI visit identification model, first, we need extract a set of in-POI and out-POI queries as train samples. Similar to fingerprint generation, for WiVisit system, the in-POI queries are those that contain a POI Wi-Fi and the RSSI value is high enough and without additional GPS info. In addition, for the WiVisit system, to make sure that the out-POI queries take place not only truly outside the POIs, but also not far from them, the out-POI queries are extracted based on three criteria: (1) they contain GPS info and their GPS location accuracy [33], [34] is no larger than 30 meters; (2) their distance from POI positions is less than 100 meters; (3) they contain one binding Wi-Fi in their scanned Wi-Fi lists at least. Due to that the Wi-Fi signal can penetrate walls, a Wi-Fi can be scanned in multiple neighboring POIs, meaning a Wi-Fi can appear in different POI fingerprints. Consequently, each query can generate multiple feature vectors for different POIs, whose fingerprint contains at least one Wi-Fi that is in the scanned Wi-Fi list of the query, which makes a query sample potentially correspond to multiple training samples for our identification model. For this reason, we label each sample as follows:

1. If query recalls a different POI from its raw-extracted POI, the corresponding feature vector is treated as a negative sample to the recalled POIs.
2. If a query recalls the same POI as its raw-extracted POI, the original label is used.

### 5.2. Feature Extraction

Recently, almost all POIs deploy their own Wi-Fi routers to provide Wi-Fi services for their employees and visitors. Therefore, when a user visits a POI, the POI's Wi-Fi devices will appear on the scanned Wi-Fi list of the user. However, the range of that a Wi-Fi can be scanned is limited. If a user does not visit a POI, the user is likely to fail to scan the POI's Wi-Fi. There is a strong correlation between scanned Wi-Fi and POI visit, which is crucial for POI visit identification. Thus, given a Wi-Fi query and a POI  $p$ ,  $P(p|query)$  is the probability of the query occurring in POI. We extracted several features from four dimensions to reflect  $P(p|query)$ .

---

<sup>2</sup>For WiVisit system, we choose  $-50db$  as the threshold.

1.  $P(m|p_j)$  is defined as the probability of Wi-Fi  $m$  scanned when users visit a POI  $p_j$ . First, we assume that each Wi-Fi scanned in a query is independent from one another. Based on the bayes formula, the posterior probability  $P(p_j|query)$  is proportional to the likelihood function  $L(query|p_j)$ .

$$L(query|p_j) = \prod_{k=1}^{k=K} P(m_k|p_j) \quad (8)$$

where  $K$  is the number of scanned Wi-Fi.  $P(m_k|p_j)$  can be approximate as the rate of occurrence  $o_{jk}$ . Thus, the first posterior probability feature is:

$$F1 = \sum_{k=1}^{k=K} \log P(m_k|p_j) \quad (9)$$

However, not all of the Wi-Fi are independence with each other. Thus, we also chose some statistics features about  $P(m_k|p_j)$ , which do not require independence between the Wi-Fi:

$$F2 = \max(\log P(m_k|p_j)) \quad (10)$$

$$F3 = \text{mean}(\log P(m_k|p_j)) \quad (11)$$

2. Due to that Wi-Fi signals can penetrate walls, a POI's Wi-Fi can sometimes still be scanned even when the user is outside. However, in those cases, the user is usually further away from the POI's Wi-Fi device than in the POI, which makes the signal strength of the POI's Wi-Fi weaker. Moreover, due to the obstruction of walls, the received signal strength will also be weaker when the user is outside. For these reasons, the RSSI of POI's Wi-Fi is also very important for POI visit identification. Therefore, we calculated RSSI weighted posterior probability based on the following likelihood function:

$$P_\alpha(m_k, r_k|p_j) = \beta_k P(m_k|p_j) \quad (12)$$

where  $\beta_k = [(r_k + r_{max})/r_{mean}]^\alpha$ . Similar to Equation 9-11, we can get RSSI weighted posterior probability features:

$$F4 = \sum_{k=1}^{k=K} \log P_\alpha(m_k, r_k|p_j) \quad (13)$$

With different  $\alpha$ , we can get different likelihood functions, which will generate different posterior probability features based on Equation 13. Then, we can get a set of RSSI weighted posterior probability features.

3. In recent years, Wi-Fi is used not only to connect to the Internet, but also to communicate between smart home devices. In an indoor environment, for example, in addition to routers, there are many smart devices with Wi-Fi chips that can be scanned, such as smart TV, intelligent speakers, smart air-conditioners. Therefore, when a user is in a POI, the scanned RSSI of smart devices could also be very strong. Then, the proportion of strong



Wi-Fi in the scanned Wi-Fi list will be very high. However, if the user is outside the POI, due to the obstruction of walls and their increasing distance from the smart devices and routers, the proportion of strong Wi-Fi in the scanned Wi-Fi list will decrease. Based on this intuition, we also extracted posterior probability features between POI and Wi-Fi that is filtered by the absolute RSSI value. For these features, the likelihood function can be represented as follow:

$$P_f(m_k|p_j) = \begin{cases} P(m_k|p_j), & r_k \geq r_f \\ 0, & r_k < r_f \end{cases} \quad (14)$$

where  $r_f$  is the absolute RSSI threshold. For different absolute RSSI thresholds, based on Equation 14, the RSSI filtered posterior probability feature can be represented as:

$$F5 = \sum_{k=1}^{k=K} \log P_f(m_k|p_j) \quad (15)$$

With different  $r_f$ , we can get a set of absolute RSSI filtered features.

4. For commercial Wi-Fi devices, the RSSI measurements of Wi-Fi signal will be influenced by antennas, quality of Wi-Fi chips, the position of holding the phone, etc. Thus, even in the same location, different phones scanning the same Wi-Fi will have different RSSI values. Sometimes, features generated by the absolute RSSI value will introduce such bias. Therefore, we extracted some posterior probability features between the POI and Wi-Fi that is filtered by relative RSSI information to remove this bias. First, we extracted marginal features for the query::

$$F6 = \sum_{k \in Q_{top(x)}} \log P(m_k|p) \quad (16)$$

where  $Q_{top(x)}$  is the set of the highest  $x$  Wi-Fi based on RSSI value in the scanned Wi-Fi list of the query. Meanwhile, we extracted the edge distribution probability features for the POI:

$$F7 = \sum_{k \leq K \wedge m_k \in L_{top(s)}} \log P(m_k|p_j) \quad (17)$$

where  $L_{top(s)}$  means the highest  $s$  Wi-Fi based on median RSSI value in the fingerprint of the POI. And finally, we extracted the joint probability features:

$$F8 = \sum_{k \in Q_{top(x)} \wedge m_k \in L_{top(s)}} \log P(m_k|p_j) \quad (18)$$

Based on Equation 16 - 18, with different  $x$  and  $s$ , we can also get a set of relative RSSI filtered features.



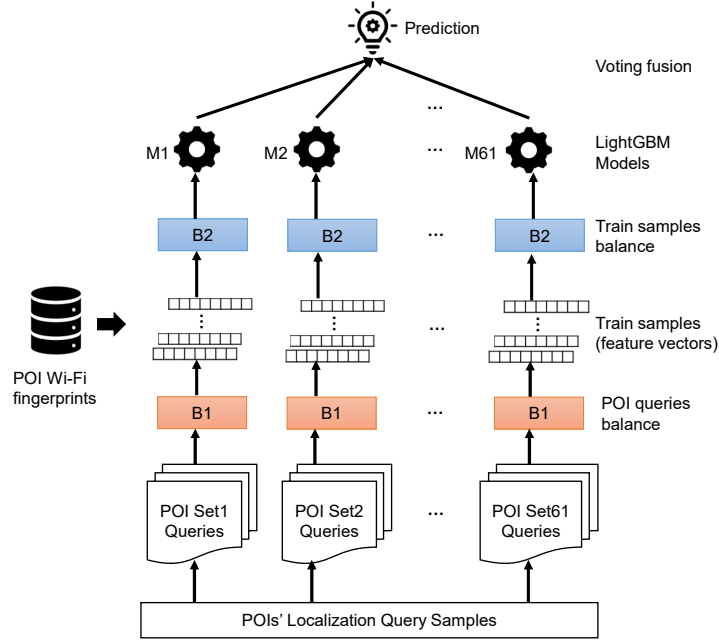


Figure 3: POI visit identification model training workflow.

### 5.3. Training Model

As described in section 5.2, based on different values of hyper-parameters, we used a 784-dimensions feature vector to represent each train sample. Meanwhile, the lightGBM [32] classification model was used to build the POI visit identification model. For the lightGBM model, we had chosen the gradient Boosting Decision tree (GBDT) as the boosting tree and binary log-loss as the loss function. The number of leaves for each tree was set as 127. Meanwhile, to avoid over-fitting, the feature fraction was set as 0.6 and the bagging fraction was set as 0.7. All other parameters of lightGBM model were used as their default values.

To deal with hybrid POI types, we trained 61 sub-models by dividing query samples into 61 parts based on their POI info, and then combined their predicted results to get the final result by voting. If the number of sub-models, which gives visit prediction, is larger than 30 (i.e. half of 61), the final prediction is visit and the highest score POI is the visited POI. Figure 3 shows the full POI visit identification model training workflow.

Moreover, in order to improve the robustness and generalization of WiVisit, we brought in several optimization methods:

1. **POI queries balance:** Different POIs will have different popularity. A popular POI will have more localization queries than an unpopular POI. The unbalanced query numbers of different POIs will affect the generalization of the model. Therefore, during the sample extraction phase, the maximum number of query samples in each POI is limited.
2. **Training samples balance:** For most of POIs, the number of negative samples is much larger than positive samples. In order to get more effective parameter estimation, we

adopted random under-sampling on negative samples for each POI. Finally, for each POI, the number of negative samples is no more than 1.2 times of positive samples.

#### 5.4. Online Prediction

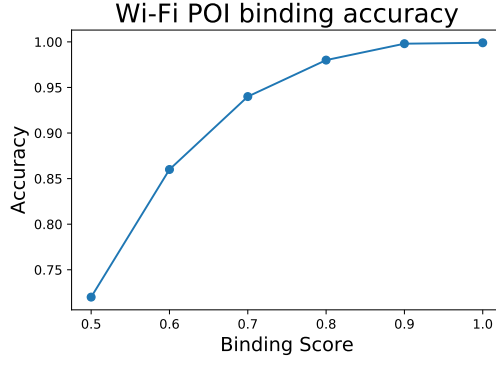
Finally, we can use the trained model to predict POI visit information. In detail, for a user localization query with scanned Wi-Fi list, the related POIs' fingerprints are extracted from the fingerprint database, whose fingerprint contains at least one Wi-Fi in the scanned Wi-Fi list of the query. For each POI, the feature vector is computed, and visit prediction of sub-models is added to obtain the final visiting score. If there is at least one POI predicted as visited finally, we selected the POI with the highest visiting score as the visiting result.

## 6. EVALUATION

### 6.1. Experiment Settings

In order to evaluate WiVisit system, we extracted the top 6 hot types ('Entertainment', 'Life services', 'Restaurant', 'Shopping', 'Fitness', 'Hotel') of POIs in Beijing from Tencent Map [31], which contained 210 sub-types and had more than 90,000 POIs in total. We then applied our POI Wi-Fi binding module, introduced in Section 4, to these POIs, to get POI Wi-Fi information. We extracted these Wi-Fis which occurred at least once in last two weeks of all Tencent localization queries as the candidate Wi-Fi set to binding POI. The size of candidate Wi-Fi set is about several billions. Due to the large number of Wi-Fi existing, for a given POI, all similar Wi-Fi to the given POI cannot be computed within a reasonable time. Therefore, given a POI, we only extracted the set of Wi-Fi whose distances from the POI is less than 200 meters ( $D_{th}$ ) which is large enough compared with common indoor Wi-Fi coverage area. The threshold  $D_{th}$  is used for filtering a small candidate binding Wi-Fi set to each POI. The normalization factor  $\gamma$  was set as 0.005, which means that the score of one POI-Wi-Fi pair will be equal to 0 if their similarity is 1 but their distance is larger than  $D_{th} = 200$ . To assess the accuracy of POI-Wi-Fi binding module, we manually labeled 20,000 POI-Wi-Fi pairs as the benchmark data set. The accuracy of different binding score thresholds was calculated and shown in Figure 4. Finally, POI-Wi-Fi pairs with binding scores larger than 0.8 and accuracy larger than 98% were used to generate POI fingerprints.

After POI-Wi-Fi binding, we obtained 42278 POIs which had binding Wi-Fi. For these POIs, we extracted raw user queries from 20210301 to 20210304 to build the fingerprint database. To train the POI identification model, we collected localization queries from 20210305 to 20210306 as the training set of WiVisit. To evaluate the identification performance, we collected localization queries from another two days (20210307, 20210308) as the test set to compare different methods. Moreover, we also manually collected real POI visited queries to evaluate the performance of WiVisit in the real world. For each POI, the visited queries are collected from two collectors with two different types of mobile phones and staying five minutes at least. The POI identification model consists of 61 sub-models. These sub-models were trained parallelly in cluster mode, whose training time is no more than half an hour. Online prediction is triggered on a server



**Figure 4:** Wi-Fi POI binding score vs. accuracy.

**Table 1**  
Sample Sets

Data Set	Time	POI Set	In-POI	Out-POI
Train	20210305-20210306	42278	1178542	1994754
Test	20210307-20210308	19506	61349	61012
MANUAL	20210307-20210320	592	6237	0

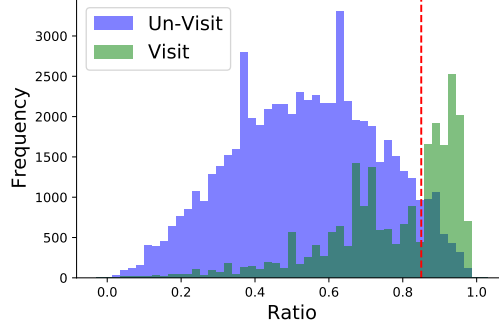
when one query is received. Based on our test, the average time of predicting is less than 30ms per query. The detailed information of these sample sets is shown in Table 1.

## 6.2. Compared Methods

1. **BASE0:** The rule for labeled visiting samples, that is, if at least one scanned Wi-Fi in the query is a binding Wi-Fi and the RSSI value is higher than -50db without any GPS info, then the query is visit sample and the corresponding POI is the visited POI. For evaluation, we only compare BASE0 with other methods in manual set.
2. **BASE1:** One intuitive assumption is that one localization query having more overlapped Wi-Fi with a POI fingerprint is more likely to occur in this POI. Based on this assumption, we computed ratios of Wi-Fi overlapping between a localization query and POIs' fingerprints.

$$R_j = \frac{|\{m|m \in query \wedge m \in FP_j\}|}{|\{m|m \in query\}|} \quad (19)$$

Where  $R_j$  is the ratio of Wi-Fi overlapping between the query and POI  $j$ . If the maximum overlapping ratio is larger than the given threshold, this query is labeled as a visit query and the POI corresponding to the maximum ratio is the visited POI. In order to choose the best threshold, we plotted the overlapping ratios between query and its associated POI



**Figure 5:** Ratio of overlapping Wi-Fi.

for each training sample shown in Figure 5<sup>3</sup>. As shown in the figure, the best threshold is 0.84.

3. **BASE2:** In addition to the ratio of Wi-Fi overlapping, we included the RSSI strength and the rate of occurrence of each Wi-Fi to define a visiting score as follows:

$$P_v(p_j|query) = \sum_{m_i} w_i o_{ji} \quad (20)$$

where  $P_v(p_j|query)$  is defined as the score of a user in POI  $p_j$  with the scanned query, and  $w_i$  is the weight of Wi-Fi  $m_i$  in query with  $r_i$ . Based on the path loss model [12], the weight function is  $w_{m_i} = 10^{(r_i - r_0)/r_{th}}$ , where  $r_i$  is the signal strength of Wi-Fi  $m_i$ ,  $r_0$  is the maximum of RSSIs of the given query, and  $r_{th}$  is a measure of divergence of all RSSIs. In our evaluation,  $r_{th}$  was fixed as 50.  $o_{ji}$  is the co-occur probability of  $p_j$  and  $m_i$ , which was defined in Section 3. For this method, the best threshold is 0.12, with the highest F score in the Train Set.

4. **WiVisit-1:** The same classification model as WiVisit with only one lightGBM model.
5. **WiVisit:** Our fusion POI-visiting model for different POI types.

### 6.3. Metrics

For POI visit identification, the objective is to not only determine if a localization query occurs in a POI, but also to identify which POI it is from. Thus, we used four dimensions metrics (precision, recall, F score, accuracy) to evaluate the performance. The following are definitions about these metrics:

- TP:  $\{query|(predicted\ visit) \wedge (real\ visit)\}$
- FP:  $\{query|(predicted\ visit) \wedge (real\ un-visit)\}$
- FN:  $\{query|(predicted\ un-visit) \wedge (real\ visit)\}$

<sup>3</sup>We remove samples whose overlapping ratio is 1 to make the figure easy to read. For negative samples, 1-overlapping-ratio samples are no more than 0.5%. For Positive sample, 1-overlapping-ratio samples are nearly 73.2%.

**Table 2**  
Test Set Performance

Method	Precision	Recall	F Score	Accuracy
BASE1	92.3%	80.8%	0.86	39.3%
BASE2	51.6%	83.6%	0.64	66.3%
WiVisit-1	81.1%	96.2%	0.88	73.4%
WiVisit	91.5%	96.4%	0.94	83.4%

- HP:  $\{query|(predicted\ POI) \equiv (real\ POI)\}$
- Precision:  $PPV = \frac{|TP|}{|TP|+|FP|}$
- Recall:  $TPR = \frac{|TP|}{|TP|+|FN|}$
- F score:  $\frac{2 \times PPV \times TPR}{PPV + TPR}$
- Accuracy:  $\frac{|HP|}{|TP|+|FP|}$

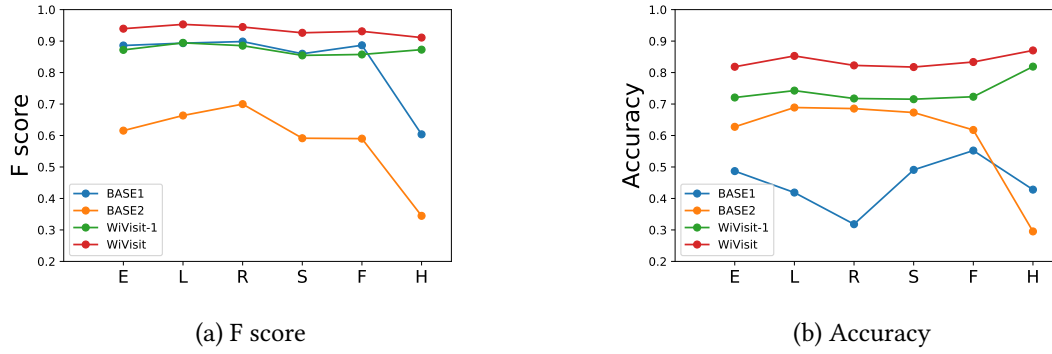
## 6.4. Results

### 6.4.1. Test Set

First, we compared these methods on the Test Set, which was collected by the same rules as the Training Set, but in different time period. Table 2 shows the evaluation result. BASE1 can accurately determine whether the user is visiting (92.3% Precision), but its predicted POI is not the real visited (39.3% Accuracy). That is, POIs with the maximum of overlapping Wi-Fis were not always the visited POIs. BASE2 had a higher recall rate and accuracy, which means the RSSI and rate of co-occurrence is more useful to find the right visited POI. However, the lower precision means it's not enough to decide whether a localization query is occurring in a POI. Compared with these two baseline methods, WiVisit-1 achieved much better performance in both POI visit judgment and final visited POI identification. Moreover, compared with WiVisit-1, since the fusion model is more robust in various POIs, WiVisit could obtained about 10% improvements in both precision and accuracy.

### 6.4.2. POI Types

We further compared these methods in different POI types, since the different types of POI had various Wi-Fi environments. As shown in Figure 6, for the 6 types of POIs, we used the F scores to show the performance on POI visit judgment and accuracies to show the performance on final visited POI identification. For POI visit judgment, as shown in the figure, BASE1 and WiVisit-1 yielded a similar performance while WiVisit-1 yielded a more robust result in different types of POIs. BASE2 gave in the worst F score with a much better accuracy for visited POI identification. Obviously, BASE1 and BASE2 were both affected by POI types, especially for "Hotel". WiVisit-1 and WiVisit achieved much robust performance in terms of F score and accuracy. Moreover, WiVisit obtained much better performance in all types of POIs compared with other methods.



**Figure 6:** Comparison of different methods over different POI types. POI types shorted as follows. E: Entertainment, L: Life services, R: Restaurant, S: Shopping, F: Fitness, H: Hotel

**Table 3**  
Manual Set Comparison

Methods	Recall	Accuracy
BASE0	32.4%	100%
BASE1	65.4%	63.7%
BASE2	88.4%	73.3%
WiVisit-1	84.4%	80.7%
WiVisit	87.2%	83.6%

### 6.4.3. Manual Set Comparison

We further compared the above methods on the Manual Set, shown in Table III. Since Manual Set only contained visit samples, we only used recall and accuracy to assess the performance. BASE0 achieved 100% accuracy, meaning the rule for visit query extraction was very effective. Meanwhile, strict rules have caused a lower recall rate 32.4% for BASE0. BASE1 obtained a remarkable improvement in recall rate. However, it did not achieve a reasonable accuracy. BASE2 had a much better balance performance in both recall and accuracy. The accuracy is much lower than the recall which means that both BASE1 and BASE2 still cannot distinguish the true visited POI among POI dense area. WiVisit1 and WiVisit still had the best performance than other methods. While WiVisit had a similar recall rate with BASE2, it achieved a significant accuracy improvement from 73.3% to 83.6%.

## 7. Conclusion

In this paper, we proposed an auto-generated Wi-Fi fingerprint-based POI visiting identification system, WiVisit, which collects accurate user' POI visit information, including indoor POIs. It is crucial for various LBS. Meanwhile, compared with traditional Wi-Fi fingerprint-based

localization methods, WiVisit does not require any human effort in fingerprint collection and updating. Therefore, WiVisit can be deployed in the real world widely and easily. Moreover, WiVisit system adopts a multi-model fusion based method for POI visit identification, which can deal with different types of POI in the real world. Based on our extensive experiments, the recall rate of WiVisit is around 90% and the accuracy is 83%, which already outperforms state-of-the-art. However, there are still many POIs cannot be bound with Wi-Fis, due to their irregular Wi-Fi SSID or no Wi-Fi device is deployed in them. In the future, crowd-sourcing methods can be used to collect more POIs' Wi-Fi information, which will make WiVisit system versatile for more POIs in the real world.

## References

- [1] M. Ye, P. Yin, W.-C. Lee, D.-L. Lee, Exploiting geographical influence for collaborative point-of-interest recommendation, in: Proceedings of SIGIR'11, 2011, p. 325–334.
- [2] E. Cho, S. A. Myers, J. Leskovec, Friendship and mobility: User movement in location-based social networks, in: Proceedings of SIGKDD'11, 2011, p. 1082–1090.
- [3] H. Yin, W. Wang, H. Wang, L. Chen, X. Zhou, Spatial-aware hierarchical collaborative deep learning for poi recommendation, *IEEE Transactions on Knowledge and Data Engineering* 29 (2017) 2537–2551.
- [4] J. Suzuki, Y. Suhara, H. Toda, K. Nishida, Personalized visited-poi assignment to individual raw gps trajectories, *ACM Trans. Spatial Algorithms Syst.* 5 (2019).
- [5] D. Ashbrook, T. Starner, Learning significant locations and predicting user movement with gps, in: Proceedings. Sixth International Symposium on Wearable Computers., 2002, pp. 101–108.
- [6] X. Cao, G. Cong, C. S. Jensen, Mining significant semantic locations from gps data, *Proc. VLDB Endow.* 3 (2010) 1009–1020.
- [7] P. Bahl, V. Padmanabhan, Radar: an in-building rf-based user location and tracking system, in: Proceedings IEEE INFOCOM, volume 2, 2000, pp. 775–784 vol.2.
- [8] Y. Jiang, X. Pan, K. Li, Q. Lv, R. P. Dick, M. Hannigan, L. Shang, Ariel: Automatic wi-fi based room fingerprinting for indoor localization, in: Proceedings of UbiComp'12, 2012, p. 441–450.
- [9] H. Xu, Z. Yang, Z. Zhou, L. Shangguan, K. Yi, Y. Liu, Enhancing wifi-based localization with visual clues, in: Proceedings of UbiComp'15, 2015, p. 963–974.
- [10] M. Youssef, A. Agrawala, The horus wlan location determination system, in: Proceedings of MobiSys'05, 2005, p. 205–218.
- [11] H.-H. Liu, Y.-N. Yang, Wifi-based indoor positioning for multi-floor environment, in: TENCON 2011 - 2011 IEEE Region 10 Conference, 2011, pp. 597–601.
- [12] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed., Prentice Hall PTR, 2001.
- [13] Q. Cai, J. Aggarwal, Automatic tracking of human motion in indoor scenes across multiple synchronized video streams, in: Sixth International Conference on Computer Vision, 1998, pp. 356–362.



- [14] J. M. Chaquet, E. J. Carmona, A. Fernández-Caballero, A survey of video datasets for human action and activity recognition, *Comput. Vis. Image Underst.* 117 (2013) 633–659.
- [15] W. Mao, J. He, L. Qiu, Cat: High-precision acoustic motion tracking, in: *Proceedings of MobiCom'16*, 2016, p. 69–81.
- [16] S. Yun, Y.-C. Chen, L. Qiu, Turning a mobile device into a mouse in the air, in: *Proceedings of MobiSys'15*, 2015, p. 15–29.
- [17] M. Altini, D. Brunelli, E. Farella, L. Benini, Bluetooth indoor localization with multiple neural networks, in: *IEEE 5th International Symposium on Wireless Pervasive Computing*, 2010, pp. 295–300.
- [18] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, Y. Liu, Tagoram: Real-time tracking of mobile rfid tags to high precision using cots devices, in: *Proceedings of MobiCom'14*, 2014, p. 237–248.
- [19] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, Z. Sahinoglu, Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks, *IEEE Signal Processing Magazine* 22 (2005) 70–84.
- [20] A. Martinelli, S. Jayousi, S. Caputo, L. Mucchi, Uwb positioning for industrial applications: the galvanic plating case study, in: *IPIN'19*, 2019, pp. 1–7.
- [21] H. Perakis, V. Gikas, Evaluation of range error calibration models for indoor uwb positioning applications, in: *IPIN'18*, 2018, pp. 206–212.
- [22] S. Lembo, S. Horsmanheimo, M. Somersalo, M. Laukkanen, L. Tuomimäki, S. Huilla, Enhancing wifi rssi fingerprint positioning accuracy: lobe-forming in radiation pattern enabled by an air-gap, in: *IPIN'19*, 2019, pp. 1–8.
- [23] H. Abdel-Nasser, R. Samir, I. Sabek, M. Youssef, Monophy: Mono-stream-based device-free wlan localization via physical layer information, in: *IEEE Wireless Communications and Networking Conference*, 2013, pp. 4546–4551.
- [24] K. Chintalapudi, A. Padmanabha Iyer, V. N. Padmanabhan, Indoor localization without the pain, in: *Proceedings of MobiCom'10*, 2010, p. 173–184.
- [25] X. Li, S. Li, D. Zhang, J. Xiong, Y. Wang, H. Mei, Dynamic-music: Accurate device-free indoor localization, in: *Proceedings of UbiComp'16*, 2016, p. 196–207.
- [26] K. Wu, J. Xiao, Y. Yi, M. Gao, L. M. Ni, Fila: Fine-grained indoor localization, in: *Proceedings IEEE INFOCOM*, 2012, pp. 2210–2218.
- [27] D. Vasisht, S. Kumar, D. Katabi, Decimeter-level localization with a single wifi access point, in: *NSDI'16*, USENIX Association, 2016, pp. 165–178.
- [28] B. Berruet, O. Baala, A. Caminada, V. Guillet, E-loc: Enhanced csi fingerprinting localization for massive machine-type communications in wi-fi ambient connectivity, in: *IPIN'19*, 2019, pp. 1–8.
- [29] pypinyin 0.42.0, Accessed July 2, 2021. URL: <https://pypi.org/project/pypinyin>.
- [30] Tencent fanyijun, Accessed July 2, 2021. URL: <https://fanyi.qq.com>.
- [31] Tencent map, Accessed July 2, 2021. URL: <http://map.qq.com>.
- [32] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, Lightgbm: A highly efficient gradient boosting decision tree, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, volume 30, Curran Associates, Inc., 2017.