Experience: Pushing Indoor Localization from Laboratory to the Wild

Jiazhi Ni[†*], Fusang Zhang[‡*], Jie Xiong[§], Qiang Huang[†], Zhaoxin Chang[‡], Junqi Ma[‡], BinBin Xie[§], Pengsen Wang[†], Guangyu Bian[†], Xin Li[†], Chang Liu[†]
[†] Tencent Inc., [‡] Institute of Software, Chinese Academy of Sciences, [§] University of Massachusetts Amherst
andyni@tencent.com, fusang@iscas.ac.cn, jxiong@cs.umass.edu, johnnhuang@tencent.com,
zhaoxin.chang@telecom-sudparis.eu, majunqi22@otcaix.iscas.ac.cn, binbinxie@cs.umass.edu,
{pengsenwang, tinybian, clarkxinli, levenliu}@tencent.com

ABSTRACT

While GPS-based outdoor localization has become a norm, very few indoor localization systems have been deployed and used. In this paper, we share our 5-year experience on the design, development and evaluation of a large-scale WiFi indoor localization system. We address practical challenges encountered to bridge the gap between indoor localization research in the laboratory and system deployment in the wild. The system is currently used in 1469 shopping malls, 393 office buildings and 35 hospitals across 35 cities to provide location service to millions of users on a daily basis. We hope the shared experience can benefit the design of real-world indoor localization systems and the practical problems identified can change the focus of indoor localization research. We released our dataset that contains fingerprints collected from 1469 shopping malls and one office building.

CCS CONCEPTS

• **Information systems** → **Location based services**; **Global positioning systems**; • **Networks** → **Location based services**.

KEYWORDS

indoor localization, WiFi-based localization, labor-free fingerprint collection, WiFi access point name-location matching, large-scale localization, real-world deployment

1 INTRODUCTION

Outdoor localization has become an indispensable part of our daily lives. GPS-based location service is widely used for navigation and tracking. On the other hand, even with a tremendous amount of effort from both academia and industry in the last two decades, there is still no indoor localization system that can be widely adopted like GPS in outdoors [25, 40]. Different wireless technologies have been exploited for indoor localization including WiFi [8, 9, 20, 36, 51, 59, 60], Bluetooth [16, 23, 28, 34, 55] and UWB [10, 45, 63]. Bluetooth and UWB localization systems require dedicated hardware deployment. These systems can be deployed in one room or even in one building [24, 63] but they are not scalable for city-scale deployment. Based on our empirical studies, deploying a Bluetooth-based localization system in a large office building with a size of 31,695 m^2 requires more than 10,000 Bluetooth beacons and the whole system costs more than $200,000. The deployed system also needs to be maintained on a weekly basis to replace those beacons that stopped working. After a thorough study, we find that to achieve scalable indoor localization, the most practical approach is to utilize existing infrastructure and this makes WiFi the most promising candidate for indoor localization. In most shopping malls, enterprise and university buildings, WiFi infrastructure has been deployed and we do not need to deploy dedicated hardware if existing WiFi infrastructure can be utilized for localization.

In this paper, we share our 5-year experience on the design, deployment, and evaluation of a large-scale WiFi-based indoor localization system currently used by more than 20 million users across 35 cities. Besides our own App, there are more than 50 other Apps using our localization service to support their services such as food delivery and ride hailing. Among the 50+ Apps, 12 Apps have more than 10 million downloads. Our localization system utilizes 4.03 million WiFi access points (APs) deployed by the third parties to provide individual users and companies a large range of services including indoor localization, store navigation, and location-based advertising. During the 5-year process, we faced many practical challenges which were not paid attention to in the research community. In this paper, we share our experience in bridging the gap between indoor localization research in the laboratory and system deployment in real-world settings. We believe the shared experience can benefit the design and deployment of real-world indoor localization systems and the practical problems identified may also change the focus and methodology of indoor localization research.

To make large-scale WiFi localization happen in real-world settings, ***the first question we ask is what technique should be used?*** WiFi localization techniques can be broadly grouped into three categories, i.e., angle-based [31, 37, 56], time-based [22, 32, 57, 62] and fingerprint-based [21, 29, 47, 53, 54]. After a thorough investigation of these three techniques, we find that angle-based and time-based methods impose requirements on WiFi AP. Angle-based technique requires accurate signal phase measurements for angle calculation. However, only signal strength information can be extracted from commodity WiFi APs deployed in real-world settings. The Channel State Information (CSI) which contains phase readings is only available on few commodity WiFi cards (i.e., Intel 5300 [5] and Atheros AR series cards [1]). These WiFi cards are not

*Both authors contributed equally to this research.

ACM MobiCom '22, October 17–21, 2022, Sydney, NSW, Australia

ACM ISBN 978-1-4503-9181-8/22/10.
https://doi.org/10.1145/3495243.3560546

used in any of the 4.03 million actually deployed WiFi APs. On the other hand, time-based technique requires highly-accurate time measurements. The latest WiFi protocol 802.11mc [6, 42] supports exporting time information for localization. We explore the commodity WiFi APs on the market and find few models (e.g., Google WiFi AP [2] and Compulab WILD WiFi RTT router [3]) support 802.11mc. These APs supporting 802.11mc only account for 5% of WiFi APs on the market.

For fingerprint-based technique, both CSI [15, 38, 50, 54] and RSSI (Received Signal Strength Indicator) [14, 17, 18, 48, 59] fingerprints were used. We adopt the RSSI reading which is the only information available on the vast majority of WiFi-equipped devices for localization. Among the 4.03 million APs, 98.7% of them support reporting the RSSI readings. At the user side, almost all Android smartphones and windows-based laptops can report WiFi RSSI readings.

However, it is well known that fingerprint collection is labor-intensive [52, 58]. For a six-story shopping mall with a size of $36,582\ m^2$, it takes three persons ten days to collect RSSI readings at a granularity of 5 m × 5 m. This is not scalable for large-scale deployment and therefore our second question is ***how to build the RSSI fingerprint database without requiring any manual collection?*** To address this issue, we adopt a crowdsourcing-based method to collect fingerprints. The challenging part here is that the crowdsourced fingerprints do not have groundtruth locations. We propose a solution to estimate WiFi APs' coarse locations and then use the APs' locations to estimate the groundtruth locations for the crowdsourced fingerprints. Although the estimated "groundtruth locations" are not accurate initially, they are refined constantly leveraging the crowdsourced user feedback.

However, obtaining the locations of millions of APs deployed by the third parties is non-trivial. To tackle this challenge, we adopt a name-based matching scheme to obtain some of the APs' locations based on a key observation, i.e., in buildings such as shopping malls, a lot of WiFi APs have a name (SSID) related to the store. For example, the WiFi AP deployed in Starbucks can have the name STARBUCKS_WiFi. Based on our data containing 4.03 million indoor APs, we find that 19.63% of the APs have a name which can present us with the physical location of the APs in the building. For the rest APs, we infer their locations based on the proximity relationship with respect to those APs whose locations have been obtained from the AP names. We observe several other interesting real-world challenges such as power diversity across APs and the physical location change of the APs over time.

After we obtain the fingerprints, the next question is ***how to process crowdsourced fingerprint data for localization?*** Traditional fingerprint-based indoor localization systems only use the RSSI readings for localization. For a large-scale localization system, the large number of users bring unique opportunities to adopt the number of requests as a fingerprint to further improve system performance. Besides, we also discover interesting challenges such as indoor-outdoor boundary detection and large floor identification errors due to hollow-regions.

Our system has been used in several typical indoor environments, i.e., 1469 shopping malls, 393 office buildings and 35 hospitals in 35 cities. We receive an average of 8.81 million location requests from these shopping malls per day. For real-life localization services, we are not able to obtain groundtruths to calculate the localization error. From January 2020 to March 2022, our localization service received an overall rating of 4.56 on a scale of 5.0. To show the detailed performance, we manually collect the groundtruths in four shopping malls, one office building and one hospital to evaluate the localization performance. The first shopping mall picked is the third largest shopping mall in the world with an area of $449,393\ m^2$ and the other three have an area of $81,600\ m^2$, $26,650\ m^2$, and $36,582\ m^2$, respectively. We show that our system can achieve a median localization accuracy of 6.82 m without any manual fingerprint collection using only APs deployed by third parties. In the office building, we demonstrate the capability of combining WiFi with IMU data to provide even higher location accuracy. In the hospital building, although the WiFi AP deployment is sparser, reasonably accurate results can still be achieved, demonstrating the wide applicability of our system.

2 FINGERPRINT COLLECTION

RSSI fingerprint-based indoor localization has been extensively studied [27, 43, 61]. It is well known that fingerprint collection is labor-intensive and time-consuming [52, 58]. Manual collection is not scalable for large-scale deployment and we propose a crowdsourcing-based method to collect fingerprints. The basic idea is to collect fingerprints by crowdsourcing information from millions of end users. However, as there is no groundtruth location information, large-scale fingerprint collection is challenging. We propose a two-stage method to enable large-scale fingerprint collection without requiring any manual effort. The first stage is to obtain the coarse location information of WiFi APs to serve as anchor points to initialize the process of fingerprint collection. Then we iterate the process by leveraging user feedback to obtain more accurate fingerprints. We detail the process of AP location estimation, fingerprint initialization and update in the following sections.

2.1 WiFi AP location estimation

To obtain the locations of millions of WiFi APs in a scalable manner, we adopt a name-matching method to first obtain some of the APs' locations and then use the obtained locations to infer the rest APs' locations. The name-matching method is based on an interesting observation that lots of Point of Interests (PoIs) such as stores, restaurants and coffee shops deploy their own WiFi APs and the names of the APs are related to the stores. We can therefore use the AP names to infer the physical locations of the APs with the help of the building floor plan. Also almost all the shopping malls would like to share the floor plan data with us to enable localization-related services inside the building.

2.1.1 Estimating APs' locations using name information. Before we estimate the APs' locations in a building, we first extract the AP list and the information of each AP (i.e., MAC address, SSID, and RSSI) through crowdsourcing. Specifically, we crowdsource information from millions of end user devices (e.g., smartphones). Whenever a user requests a location service, the request is sent through an App to our server. The request contains all the WiFi APs that can be overheard by the device and the information of each AP.

To match the WiFi AP name with a PoI to obtain the location of the AP, we adopt a natural language processing (NLP) model [7] to interpret the AP name. The adopted NLP model supports English, Spanish, Chinese and special characters such as @, *, - and &. To improve the matching accuracy, we also take abbreviation and similarity of words into consideration. For example, SB_WiFi can be used as the WiFi AP name and SB represents Starbucks.

Then we adopt an information retrieval model proposed by Google, i.e., deep & wide model [19] to match the WiFi AP name with the PoI name. Note that during the matching process, we only match the PoI names and AP names from the same building. For a total of 301,216 PoIs and 4.03 million APs, we can complete the matching in around three minutes on a distributed server system with ten servers. Each server is equipped with two Intel Xeon 14-core E5-2680v4 CPUs and 256 GB of memory. Among the 4.03 million WiFi APs, 791,893 APs (i.e., 19.63%) can be matched with PoIs. Among them, 38.23%, 13.50% and 7.65% of APs at shopping malls, office buildings and hospitals can be matched respectively. To evaluate the accuracy of the name-matching method, we manually collect the groundtruth of 8,000 APs in 1469 shopping malls. The achieved name matching accuracy is 98.4%.

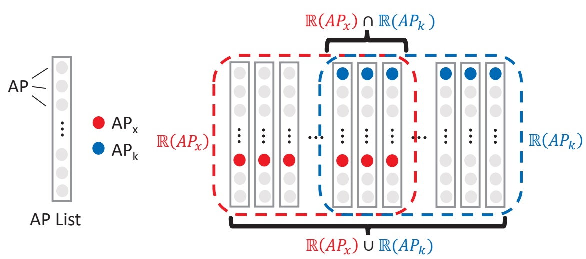


Figure 1: Each device reports a list of APs overheard. Two WiFi APs are overheard by different user devices.

2.1.2 Estimating the remaining APs' locations. The name-matching scheme can help obtain part of the APs' locations. For the rest APs, we propose a novel scheme which utilizes the crowdsourced data to obtain the proximity information between APs to obtain their locations.

Let us take one AP from the rest unknown-location APs and denote it as AP_x to illustrate how to estimate its location. If AP_x and another AP (AP_k) whose location has been obtained in the name-matching phase can be overheard by the same device, these two APs are not too far away from each other. To quantify how close these two APs are, we employ the AP lists crowdsourced from the end users. Every time a user sends a service request, an AP list is also sent to our server and recorded. As shown in Figure 1, the AP lists containing AP_x are denoted as $\mathbb{R}(AP_x)$ and the lists containing AP_k are denoted as $\mathbb{R}(AP_k)$. Those AP lists contain both AP_x and AP_k are denoted as $\mathbb{R}(AP_x)\bigcap\mathbb{R}(AP_k)$. The AP lists containing AP_x or AP_k can be denoted as $\mathbb{R}(AP_x)\bigcup\mathbb{R}(AP_k)$. If we use N and M to indicate the number of AP lists in $\mathbb{R}(AP_x)\bigcap\mathbb{R}(AP_k)$ and $\mathbb{R}(AP_x)\bigcup\mathbb{R}(AP_k)$, respectively, we can leverage the crowdsourced data to quantify the two APs' proximity as

$$Prox(AP_x, AP_k) = \frac{N}{M}. \tag{1}$$

The value of $Prox(AP_x, AP_k)$ is in the range of 0∼1. Value 1 indicates AP_x and AP_k always appear together on a list.

In real-world settings, Equation (1) is still coarse. A lot of AP pairs can achieve a proximity value higher than 0.95 because whether two APs on the same list (heard by the same device) is a low bar for proximity measurement. To obtain more fine-grained proximity information, after the two APs appear on the same list, we further calculate the RSSI difference of the two APs, i.e., $RSSI_x - RSSI_k$. A smaller RSSI difference means two APs have a chance to be closer to each other. We include this term in the denominator. Also if the APs' RSSI readings are larger, they are more accurate readings. So we include another term $Mean(RSSI_x, RSSI_k)$ in the numerator to characterize this effect.[1] By taking these two measures into consideration, Equation (1) can be rewritten as below

$$Prox(AP_x, AP_k) = \frac{\sum_{i=1}^{N} \frac{Mean(RSSI_{xi},\, RSSI_{ki})}{|RSSI_{xi} - RSSI_{ki}|}}{M}. \tag{2}$$

Note that for the two cases in Figure 2, the obtained values from $\frac{Mean(RSSI_{xi},\, RSSI_{ki})}{|RSSI_{xi} - RSSI_{ki}|}$ are both large. The beauty of Equation (2) is that it takes the number of AP lists (i.e., N) which have both APs into consideration. For case B in Figure 2, the N value would be small and therefore the final value of Equation (2) is still small.

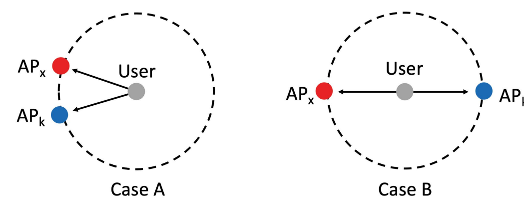


Figure 2: Two typical cases in AP proximity estimation.

For AP_x, we apply Equation (2) to rank those APs whose locations have been obtained. A larger value indicates a higher chance of being closer to AP_x. Before we determine the location of AP_x, we first infer the floor information of AP_x. Floor information is very important in many real-life scenarios [26, 39]. An interesting observation is that users can usually tolerate a large localization error but not a floor error and this observation was also verified in another recent study [34]. Among the APs (e.g., AP_k) whose locations have been obtained in the name-matching stage, we take the top N_f APs with the largest proximity values with respect to AP_x. If the top N_f closest known-location APs are all on the same floor, we determine the same floor as AP_x's floor. If the top N_f closest APs are not on the same floor, we defer the determination to future rounds. The rationale is that in this round, we are not able to get enough number of known-location APs on the same floor close to AP_x so we better defer the determination to later rounds as in each round, some APs' locations can be estimated and become known-location APs in the next round. We set the value of N_f as three considering the trade-off between accuracy and number of estimation rounds.

After the floor information of an AP is determined, we continue to calculate the AP's location $\mathbb{P}(AP_x)$ using the location information of the known-location APs on the same floor and weight the

[1] Note that RSSI difference and mean value need to be normalized to the range of 0∼1.

contribution based on the proximity value calculated in Equation (2)

$$\mathbb{P}(AP_x) = \sum_{k=1}^{T} W(AP_x, AP_k) \times \mathbb{P}(AP_k). \tag{3}$$

Here, $W(AP_x, AP_k)$ is the weight by normalizing the sum of the proximity value $Prox(AP_x, AP_k)$ as 1, and T is the number of known-location APs on the same floor. After the APs' locations are obtained, the APs are added to the known-location AP group to start a new round. We stop the process when no new APs' locations can be estimated. Based on the data from 1469 shopping malls, 393 office buildings, and 35 hospitals, after an average of 4.1 rounds, 7.3 rounds, and 8.7 rounds, respectively, the number of APs whose locations can be estimated saturates. For shopping malls with a higher percentage of name-matching APs (38.23%), less rounds are needed. In the three environments, eventually 91.10%, 91.40%, and 86.87% APs' locations can be estimated. Note that for fingerprint-based localization, we do not require to know all the AP's locations. We also study the AP location estimation accuracy. To evaluate the performance of AP location accuracy, we manually collect the groundtruth locations of 8,000 APs. The experiment results show that in shopping malls, office buildings and hospitals, the median errors are 5.16 m, 8.65 m and 23.61 m, respectively. We want to emphasize that the APs' locations are only used for the initialization of fingerprint collection. More accurate AP locations lead to more accurate initial fingerprints and accordingly a faster convergence. Less accurate AP locations do not necessarily mean less accurate localization performance.

2.2 Fingerprint collection

After the APs' locations are estimated, we now introduce how to build the fingerprint database without any manual effort. As shown in Figure 3, this approach consists of two modules, i.e., initialization module and user feedback-based update module.

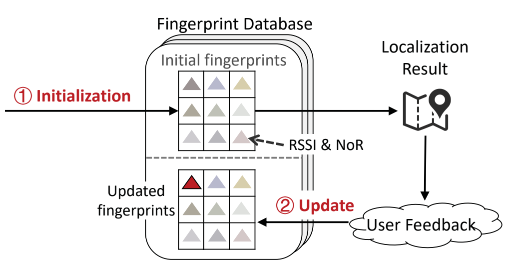


Figure 3: Fingerprint initialization and update.

2.2.1 Fingerprint composition. The fundamental two components of a fingerprint are (i) a location (i.e., a square grid), and (ii) unique features associated with this location. Traditional WiFi-based fingerprint only uses the RSSI readings overheard from multiple APs as the unique feature to form a fingerprint. In our work, we add one more feature, i.e., the number of requests (NoR) at a location which is available only when we have a large amount of data and we term this new feature data-driven feature. This data-driven feature can be utilized to further improve localization accuracy.

2.2.2 Fingerprint initialization. Based on the APs' location information obtained in Section 2.1, we start fingerprint initialization at each grid. The key difference between our large-scale automated fingerprint collection and traditional manual collection is that we do not have any groundtruth locations. The critical part is to obtain accurate location information to link it with the features (i.e., RSSI and NoR) to form the fingerprint.

We adopt an iteration-based scheme leveraging data crowd-sourced from users to initiate the process and employ user feedbacks to remove inaccurate fingerprints. Whenever a location service request is sent to our server, a list of overheard APs and the RSSI information of each AP are also sent to our server. Based on the APs' locations and the RSSI values, we can coarsely estimate the user's location $\boldsymbol{Loc_{user}} = (x_{user}, y_{user})$ by applying the following equation

$$\boldsymbol{Loc_{user}} = \frac{\sum_{i=1}^{N} RSSI_i \cdot \boldsymbol{Loc_i}}{\sum_{i=1}^{N} RSSI_i}, \tag{4}$$

where $RSSI_i$ is the normalized signal strength in the range of 0~1 to quantify the weight of each AP's contribution. $\boldsymbol{Loc_i} = (x_i, y_i)$ is the location of the ith AP and N is the number of overheard APs. The weights of all the APs add to 1. Larger RSSI means larger weights. The rationale is that if the RSSI value is larger, the AP has a higher chance to be close to the user location. Note that due to multipath, larger RSSI does not always indicate a closer location. With a large number of APs, the effect of multipath is mitigated. Also the feedback mechanism (Section 2.2.3) can further refine the estimates. Once the location where the user sends the request is estimated, the initial fingerprint at the grid where the user is located can be obtained. With lots of users, we can quickly obtain fingerprints at most locations. Based on our data from 1469 shopping malls, after an average of 6.6 days, the fingerprints at 87.4% locations can be collected. The remaining 12.6% locations are rarely visited by users and therefore they have little effect on our localization service. For the number of requests (NoR), the initial value is set to zero and each time a request is sent from a grid, the value for that grid is increased by one.

2.2.3 Fingerprint update. The initial fingerprints obtained in Section 2.2.2 may not be accurate. We further utilize a large amount of user feedback to refine the fingerprints. Based on our experience, with some incentive programs such as "win a coupon by giving us feedback", we can receive feedbacks from around 5% of the users. The feedback can be as simple as "Please rate our service on a scale of 1~5". To make sure the update is effective, we keep the fingerprints used in services rated "5" (very satisfied). For those services scored 1~4, we label the fingerprints involved as "need to be updated". Then we use the latest AP information and the method described in the initialization phase to obtain new fingerprints at those locations. We will only replace the original fingerprint with the new fingerprint if we receive an equal or higher feedback score in later services involving the new fingerprint. This process continues until all the fingerprints receive a score of 5. Note that a fingerprint with a score of 5 can receive a lower score later (e.g., 3). Our system will update it when a new score of 5 is received. This process can take 1-5 weeks. In shopping malls, it takes an average of 6.3 days for the fingerprints to become stable. It is worth mentioning that such an update scheme also makes the system robust against the changes of WiFi APs. Over the time, some APs disappear and new APs come in. It is interesting that the APs' location changes can be used to infer the migration of shops and tenants. In three typical cities (i.e., big, medium and small), we observe a 23.1%, 18.2% and 15.5% AP location change during one year.

3 DATA PROCESSING

In this section, we describe how to process the crowdsourced data for localization in real world.

3.1 Multi-feature based localization

Traditional fingerprint-based indoor localization systems rely on RSSI readings. The basic idea is to find the grid in the database which can present the highest RSSI similarity with the online RSSI measurements. In our system, we adopt features from traditional RSSI readings as well as the number of requests for localization. The processing pipeline of our approach is shown in Figure 4.

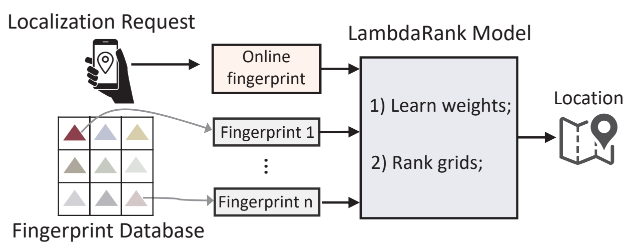


Figure 4: The processing pipeline of localization service.

RSSI feature. RSSI readings are unique at different locations. RSSI reading from just one AP is not enough to differentiate between locations and multiple APs are required. To calculate the similarity between the online RSSI readings and the RSSI readings in the database, we need to first identify the common APs on the two lists, i.e., the AP list sent through the service request and the AP list associated with the grid stored in the database. We then calculate the similarity of RSSI readings between the common APs on the two lists using four different metrics, i.e., Euclidean distance, rank distance, distribute distance and reverse-pair rate [30, 33, 49]. For all the four metrics, a smaller value means the two RSSIs are more similar. Note that if the number of common APs between two lists is less than 3, the grid will be excluded from consideration. This is because too few APs cannot guarantee the uniqueness of the RSSI fingerprint.

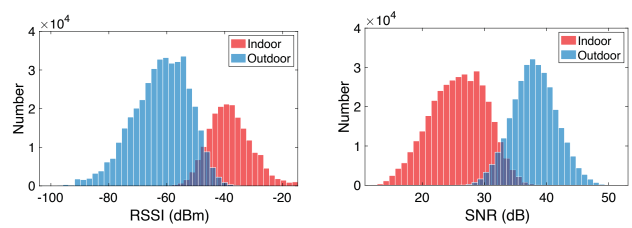
Number of requests (NoR) feature. The number of historical service requests is another feature used in our system. We find that the number of requests queried at different locations (i.e., grids) presents meaningful information to help differentiate two grids. The NoR feature tells which location is more popular when users request a location service. We observe a much higher NoR near the entrance of a PoI (e.g., a restaurant) compared to the area deep inside the PoI. A higher NoR value indicates a higher probability the request is from this grid. We consider the number of requests in the previous 30 days to obtain the NoR feature.

To effectively combine the two features, we adopt the well-known LambdaRank model [13] to learn the appropriate weight for each feature using the large amount of data crowdsourced. The LambdaRank model takes the features of all the grids as input and ranks the grids. The top grid is the estimated location. As our system does not require manually collecting any groundtruths for model training, we leverage the user feedback to identify more accurate fingerprints to be used for training. If a user rates the service high (e.g., very satisfied), the fingerprints involved in this localization service will be used as the training data to update the model.

3.2 Practical challenges in real world

In this section, we discuss two real-world challenges we encountered during the process of design and deployment.

3.2.1 Indoor-outdoor boundary detection. Boundary detection is critical for accurate localization. For example, when a user enters a shopping mall, if GPS service is still used, a large error would occur. Existing works [64, 65] usually employ sensor data extracted from smartphones such as luminance and atmospheric pressure to achieve accurate indoor-outdoor boundary detection. The luminance information can be obtained from the light sensor and the atmospheric pressure can be extracted from the barometer. However, not all smartphones have these two sensors. The barometer is usually only available in high-end smartphones. On the other hand, the performance of light sensor can be affected by weather and time of day. In our system, we use just the GPS and WiFi signal strength distribution for indoor-outdoor detection. Figure 5a and Figure 5b show one measured example of WiFi RSSI distribution and the GPS SNR distribution in indoor and outdoor environments. The higher the WiFi RSSI and the lower the GPS SNR, the more likely it is indoor. We collect the WiFi and GPS signal strength distribution at various locations and train a model for indoor-outdoor differentiation. We can achieve an accuracy of 98.7% which is comparable to that achieved with expensive high-end sensors.

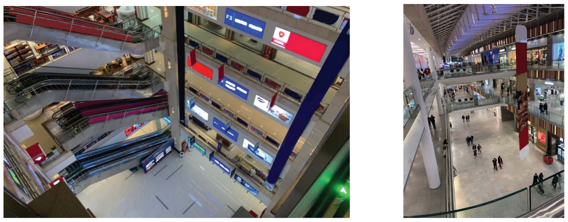


(a) WiFi RSSI distribution **(b) GPS SNR distribution**

Figure 5: The distributions utilized for boundary detection.

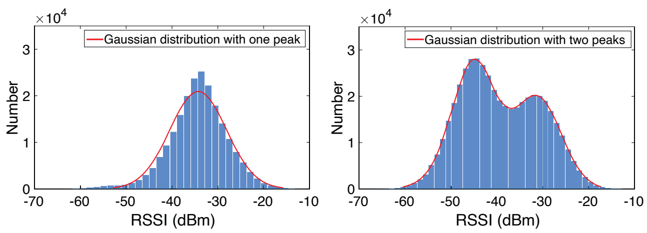
3.2.2 Hollow region in a building. We observe large floor identification errors when there are hollow regions in a building as shown in Figure 6. This is because floor identification is based on RSSI difference between APs on different floors. Due to the attenuation of the floor ground, the RSSI strengths from the APs on the same floor are higher than those on different floors. However, the hollow regions make this invalid because the floor attenuation disappears.

We present a novel method to handle this issue in real world. We observe that the RSSI distribution of APs at the edge of hollow



(a) Escalator (b) Indoor bridge

Figure 6: Two example hollow regions in shopping malls.



(a) Without a hollow region (b) With a hollow region

Figure 7: Dramatically different RSSI distributions can be observed in shopping malls with and without hollow regions.

region is different from that of other APs. As shown in Figure 7, the WiFi RSSI distribution of APs at the edge of a hollow region shows two peaks in the Gaussian distribution while that of other APs shows just one peak. This is a very interesting result. In the hollow regions, the first peak corresponds to those RSSI readings measured at locations on the other floors. The peak is higher because the number of readings from multiple other floors is more than that from the same floor. The second peak corresponds to those RSSI readings measured at locations on the same floor. The RSSI values are larger (stronger signal) but the number is smaller. Based on this observation, we use the number of RSSI distribution peaks to determine if one AP is located at the edge of a hollow region. For floor identification, we exclude those APs located at the edge of hollow regions because they can cause large errors.

4 LARGE-SCALE SYSTEM EVALUATION

In this section, we evaluate the system performance in real-world environments, i.e., shopping malls, office buildings, and hospitals.

4.1 Localization in shopping malls

Localization in large shopping malls plays a critical role in navigating users to locations of interest. Our system is used in 1469 shopping malls. All the WiFi APs are deployed by third parties and we do not have any control over them. The fingerprint dataset is available at: https://github.com/IndoorFingerprint/IndoorFingerprintData. The dataset[2] includes the RSSI fingerprints at a granularity of 5 m × 5 m in the 1469 shopping malls and the anonymous store information (mall ID, shop ID, and location).

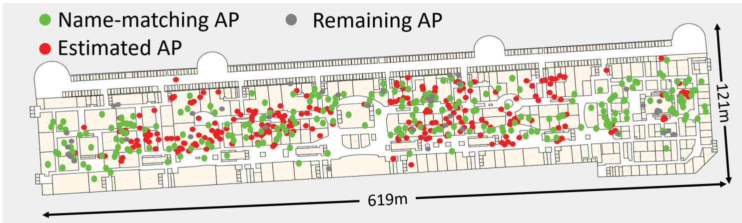
Overall performance. We receive an average of 8.81 million location requests per day in shipping malls. For localization services in real world, we are not able to obtain groundtruths to calculate the location error in meters. What we receive is the user rating. From January 2020 to March 2022, our localization service received an overall rating of 4.56 on a scale of 5.0 based on 41,500 ratings.

To show the detailed errors, we pick four shopping malls, including one large, one medium, one small, and one special shape mall. The groundtruths are manually collected. To reduce human labor cost, we select those areas from where we received most service requests and collect groundtruth measurements in those areas. For example, we collect groundtruth near the elevators and stairs, in front of shops/stores and at the corridor corners. Note that this is different from the routine operation adopted in lab evaluation that usually selects locations evenly distributed inside the area of interest. Then, we use a laser meter to measure the distance with respect to the reference such as walls which can be easily identified on the floor plan. After the groundtruth location is selected, the volunteer stops at the location and sends out location service requests.

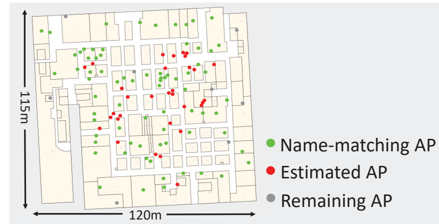
Figure 8 shows the floor plans of the first floor of the selected shopping malls with all the APs marked. The first 6-story large shopping mall is the third largest shopping mall in the world with a total area of $449,393\ m^2$. There are 3012 WiFi APs deployed by third parties. The rest three shopping malls have a total area of $81,600\ m^2$, $26,650\ m^2$, and $36,582\ m^2$ with 561, 212, and 442 APs deployed, respectively. We choose the large and the medium shopping malls for evaluation because we would like to compare our localization service with iPhone's localization service. iPhone does not report RSSI readings to third parties and Apple provides its own localization service [4]. In these two large shopping malls, Apple actually collaborates with a company to manually collect fingerprints for localization. We compare the performance of our system without any manual fingerprint collection to Apple's localization service with manual collection. In the small and special shape shopping malls, Apple does not perform any manual fingerprint collection.

We pick a total of 31,140 locations in the four shopping malls, i.e., 20,300 in the large one, 5,230 in the medium one, 2,510 in the small one, and 3,100 in the special shape one, to evaluate the accuracy of localization services provided by our system and by Apple. For iPhone, we use iPhone 8 Plus, iPhone XS MAX and iPhone 13. For Android smartphones, we include Google Pixel 6, Samsung Galaxy S9, Xiaomi 8 and Huawei Mate 9. The overall performance (localization error) is shown in Figure 9. We can see that in the large and medium shopping malls, our system without any manual collection can achieve a median error of 6.12 m and 6.43 m, which is comparable to Apple's service (i.e., 6.48 m and 6.50 m) with manual fingerprint collection. In the small and special shape shopping malls, without manual collection, the performance of Apple's localization service (i.e., 17.24 m and 12.36 m) degrades significantly while our system still achieves a high accuracy (i.e., 8.74 m and 6.92 m). We also compare the system performance when users use different brands of Android smartphones. The results are shown in Figure 10 and we do not see a clear difference between smartphone brands.

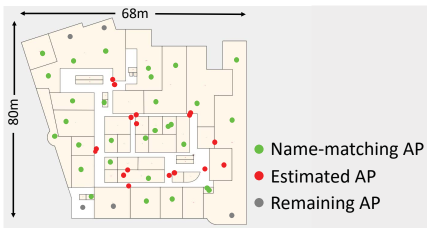
[2]Due to the space limit of github (1 GB), we can only upload part of the dataset. The full dataset (150 GB) is available upon request.



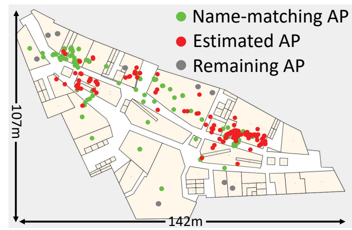
(a) First floor floorplan of the large shopping mall



(b) First floor floorplan of the medium shopping mall



(c) First floor floorplan of the small shopping mall



(d) First floor of the special shape shopping mall

Figure 8: AP deployment in four different shopping malls. The name-matching APs, the further estimated APs and the remaining APs are marked in green, red and gray, respectively.

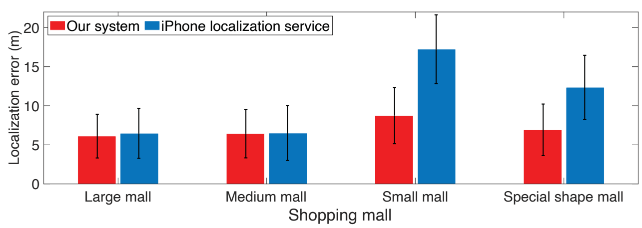


Figure 9: The performance comparison between our system and Apple's location service in the four shopping malls.

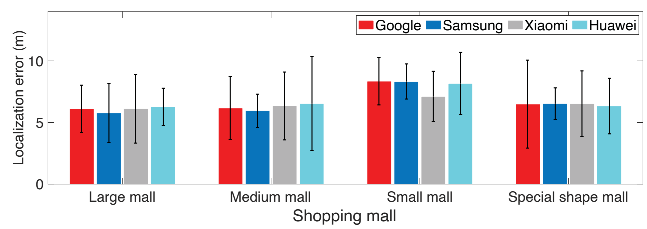
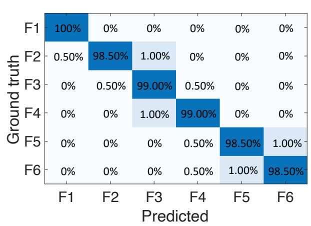
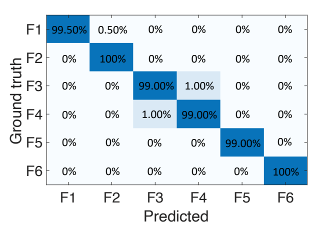


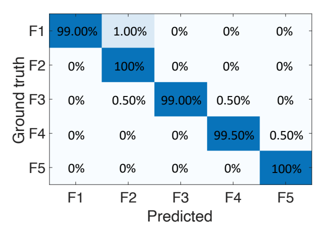
Figure 10: The localization performance comparison among different smartphone brands in the four shopping malls.



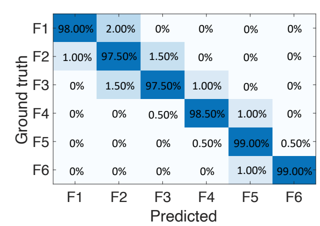
(a) Large shopping mall



(b) Medium shopping mall



(c) Small shopping mall



(d) Special shape shopping mall

Figure 11: Confusion matrix of floor detection.

AP location estimation. As illustrated in Section 2.1, we do not manually measure the AP locations, but localize APs using the proposed name-matching scheme and proximity information between APs. In the four shopping malls, we can use the name-matching scheme to obtain the locations of 1592, 288, 101, and 176 APs, accounting for 52.86%, 51.33%, 47.64%, and 39.82% of all APs, respectively. As shown in Figure 8, we highlight the name-matched APs in green. The proximity-based scheme is then used to infer the location of another 1251, 237, 84, and 230 APs, respectively. These APs are highlighted in red. The total percentages of APs whose locations can be estimated by our system are 94.38%, 93.58%, 87.26% and 91.86% in the four shopping malls, respectively. The very few remaining APs are marked in gray.

Dealing with hollow region. In this experiment, we evaluate the performance of identifying the floor information for the 31,140 locations selected. By applying the proposed method to deal with

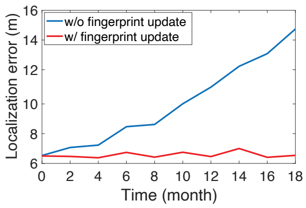


Figure 12: Fingerprint update addresses the issue of AP location change over time.

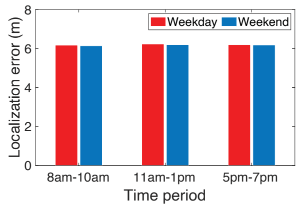


Figure 13: The effect of time of day with different amounts of multipath.

hollow-region induced errors, we improve the floor identification accuracy from 55.63% , 56.12%, 56.75%, and 53.82% to 98.92%, 99.31%, 99.53%, and 98.25% in the four malls. Figure 11 shows the confusion matrix of floor identification with the proposed method.

APs' physical location change. In our long-term evaluation in the two bigger shopping malls, we discover 19.1% of the WiFi APs are moved to different locations during one year. Larger errors would occur if fingerprints are not updated. The proposed method in Section 2.2.3 can address this issue by updating the fingerprint without any manual collection. Figure 12 shows the median localization errors on a monthly basis over a period of 18 months in the two shopping malls with and without fingerprint updates. The errors without fingerprint update increase with time. In the 18th month, the error without fingerprint update is 15.1 m while the error remains around 6.5 m with fingerprint update.

Multipath effect. Multipath is a key issue affecting the localization performance. More users can cause richer multipath and it is often evaluated in the laboratory environment to study the effect of multipath. In this experiment, we would like to see if it is a problem in real-world settings. We pick three time periods with different amount of customer flows in both weekdays and weekends. The time periods are 8:00 am~10:00 am, 11:00 am~1:00 pm, and 5:00 pm~7:00 pm. The average numbers of localization service requests in the two bigger shopping malls are 2697, 3959 and 2171 per day during the three time periods respectively in the weekdays. The numbers are changed to 2288, 4375 and 3522 in the weekends. We notice that the shopping malls located in the central business district have smaller number of customers during the weekend. The number of service requests can coarsely reflect the customer flow in the two shopping malls. Figure 13 shows the median localization errors in different time periods. We can see that the error slightly increases when the number of users increases. However, the performance degradation is very limited, demonstrating the robustness of the performance in real world.

Dealing with large-scale uneven service requests over time. We observe that the number of location service requests varies dramatically between weekdays and weekends/holidays. In some shopping malls, the number of service requests can increase by five times during holidays. This can cause larger service latency and even degraded service accuracy if not properly handled. To address this issue, we predict the number of service requests based on historical data and adapt our cloud server to ensure service quality (i.e., latency and accuracy). Specifically, we establish a LightGBM model [35] to predict the future changes in the number of service requests. It uses the historical service request data together with additional information such as the location of the shopping mall, weather, and temperature for prediction. This model can also be used to estimate the peak value of service requests to ensure sufficient server resources are reserved.

4.2 Localization in office buildings

Our system is able to achieve a median localization accuracy of 6.82 m in the four shopping malls. In office buildings, the achieved accuracy is lower. This is because in office buildings, the AP deployment is sparser due to optimized deployment strategy and the APs are usually deployed by the same operator, while in shopping malls, different shops/stores usually deploy their own APs without taking other PoI's AP deployment into consideration. However, in office buildings, higher localization accuracy is usually expected because office rooms are smaller than stores/shops. To improve the localization accuracy, we integrate the proposed localization method with the IMU sensor data widely available on smartphones. Compared to the low RSSI sample rate (0.0167 Hz ~ 0.5 Hz), the sample rate of IMU sensors can be up to 100 Hz [41, 46]. The sensor data from accelerometer, gyroscope, and magnetometer can provide the walking speed, direction and displacement information by applying Pedestrian Dead Reckoning (PDR) algorithm [11, 44]. As shown in Figure 14, we conduct experiments in an 11-story office building with a size of 160, 120 m^2 and a total of 312 APs.

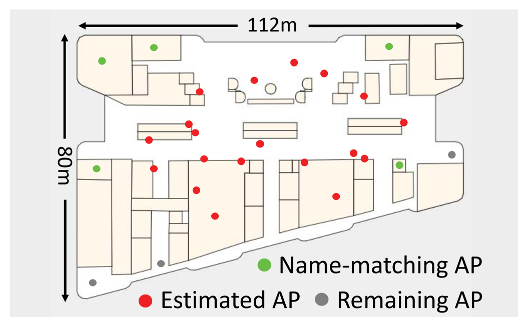
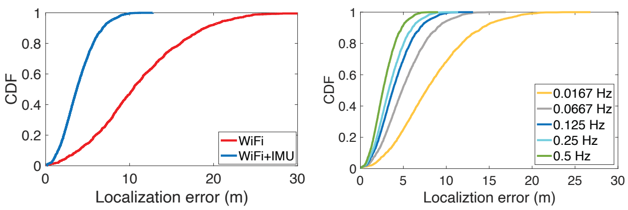


Figure 14: First floor floorplan of the office building.

Overall Performance. We compare the performance of real-time navigation with and without IMU sensor data. To get the groundtruths, we mark the trajectories on the ground and ask the volunteers to walk following the pre-defined trajectories. We test 10 different trajectories in the building with the length in the range of 50 to 150 m. 50 volunteers are asked to walk along these 10 trajectories with a phone in the hand. The overall performance is reported in Figure 15a. We can see that by applying the IMU data, we can reduce the median error from 9.13 m to 3.79 m.

The sample rate of RSSI readings. Meanwhile, we observe that in order to save battery power, the RSSI readings on smartphones are reported at a rate of 0.0167 Hz~0.5 Hz [12]. This is another parameter affecting the localization performance. Different Android OS versions support different RSSI sample rates. We thus evaluate the performance under different RSSI sample rates. As shown in Figure 15b, when both IMU and WiFi data are used, the median errors are 7.37 m, 4.76 m, 3.79 m, 3.26 m, and 2.55 m for the five different RSSI (i.e., 0.0167 Hz, 0.0667 Hz, 0.125 Hz, 0.25 Hz, and 0.5 Hz) sample rates, respectively.



(a) Trajectory tracking errors (b) Impact of RSSI sample rate.

Figure 15: Localization performance in an office building by combing WiFi fingerprinting with IMU data.

4.3 Localization in hospitals

We pick a large 11-story hospital with a total area of $59,400\ m^2$ and 223 WiFi APs for our experiment. The floor plan of the first floor is shown in Figure 16. The hospital environment is very different from shopping malls. Fewer WiFi APs are deployed, and a lot of AP names are not related to the Point of Interests (PoIs). However, we do observe something unique and interesting in hospital environment which can be leveraged for localization. We find that there are usually cashier counters in a hospital.[3] There are also self-service printer kiosks deployed for users to print inspection reports. To ensure a stable WiFi connection for payment and service, usually there are WiFi APs deployed nearby.

For other parts of the hospital, the name-matched APs are sparser but we do find matched APs with names such as "ICU" and "OAG". OAG is the abbreviation for Obstetrics and Gynecology Department. In this hospital, a median localization error of 11.85 m can be achieved. The slight accuracy decrease is mainly due to sparser AP deployment and fewer name-matched APs. However, the achieved performance is still reasonably good. We believe this is because fingerprint-based localization actually does not require a lot of APs to be overheard. 4–6 APs are usually enough for fingerprint-based location scheme to work reasonably well.

5 DISCUSSION

- **Applications and services.** Our localization system has been used to provide other services such as location-based advertising, ride hailing and food delivery. Each of these services poses unique new real-world challenges. For example, for ride hailing, the initial starting point is usually not the user's current location but the building's entrance. Another very interesting scenario which often requires localization service is the underground parking lot. However, the number of WiFi APs in parking lot is very limited, imposing challenges to enable service there.
- **Operation mode.** We provide three different ways to use the provided indoor location services, i.e., App, SDK (software development kit), and web RESTful API. We provide App download through web link (https://map.qq.com/) and App Store (https://apps.apple.com/us/app/id481623196) for end-users. We also offer localization SDK, mainly for third-party Apps, including ride hailing and food delivery services. RESTful API is primarily used to localize IoT devices such as smartwatches.
- **System bootstrapping.** As our company owns one of the most popular map Apps in China, i.e., the Tencent Map, which is used by millions of users, we had WiFi data (e.g., AP names, and building names) and GPS data before we launch this fine-grained indoor localization service. These data can help us bootstrap our indoor localization system by providing users with coarse localization service in the initial stage. Based on our experience, in a busy shopping mall, it takes just 1-2 weeks for our system to update the fingerprints based on user crowdsourcing. However, we believe this quick bootstrapping process benefits from a large number of users already using our company's other services. This process can take longer if a new company wants to provide this service. In this case, we recommend recruiting some volunteers to use the App deliberately to accelerate the bootstrapping process.
- **Server deployment.** We have cloud servers distributed in three different cities. The servers are not per-building based but are used to take care of all the localization services nationwide. For data processing in this paper, we utilize 10 distributed servers located in three cities, and each server is equipped with two Intel Xeon 14-core E5-2680v4 CPUs and 256 GB of memory.

[3]We note that in some Asian countries, cashier counters are common in hospitals.

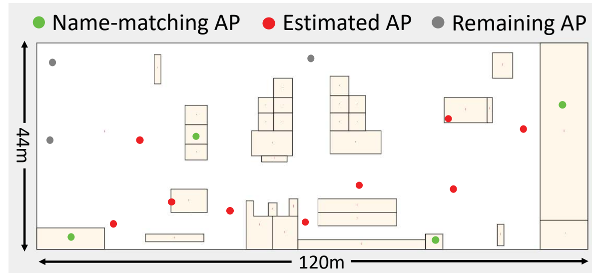


Figure 16: First floor floorplan of the hospital.

6 CONCLUSION

To conclude, in this paper, we share our insights and experience utilizing pervasive third-party WiFi infrastructure to provide scalable localization service to millions of users. We focus on those practical issues we encountered and present our solutions. We hope this work can help people rethink indoor localization and trigger more indoor localization systems deployed in real world.

ACKNOWLEDGMENTS

This work is partially supported by the National Natural Science Foundation of China (No. 62172394, No. 62072450), the Tencent Mobility Research Fund T160-CSIG-2022053100001, and the Youth Innovation Promotion Association, Chinese Academy of Sciences (No. 2020109).

REFERENCES

[1] 2015. *Atheros CSI tool.* https://wands.sg/research/wifi/AtherosCSI/

[2] 2016. *Google WiFi AP.* https://store.google.com/us/product/nest_wifi?hl=en-US

[3] 2018. *Compulab WILD WiFi RTT router.* https://fit-iot.com/web/products/wild/

[4] 2021. *Apple Indoor Maps and Positioning.* https://register.apple.com/resources/indoor/Apple-Indoor-Maps-Program.pdf

[5] 2021. *Intel 5300.* https://www.intel.com/content/www/us/en/support/products/70971/wireless/legacy-intel-wireless-products/intel-wireless-series/intel-wifi-link-5300.html

[6] 2021. *WLAN RTT IEEE 802.11mc.* https://source.android.com/devices/tech/connect/wifi-rtt

[7] 2022. *Tokenization and Sentence Segmentation in NLP using spaCy.* https://spacy.io/models

[8] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Rajkumar Sethi, Deepak Vasisht, and Dinesh Bharadia. 2020. Deep Learning based Wireless Localization for Indoor Navigation. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (MobiCom).* ACM, 1–14.

[9] Paramvir Bahl and Venkata N. Padmanabhan. 2000. RADAR: An In-building RF-based User Location and Tracking System. In *Proceedings of the 19th International Conference on Computer Communications (INFOCOM).* IEEE, 775–784.

[10] Luca Barbieri, Mattia Brambilla, Andrea Trabattoni, Stefano Mervic, and Monica Nicoli. 2021. UWB Localization in a Smart Factory: Augmentation Methods and Experimental Assessment. *IEEE Transactions on Instrumentation and Measurement.* 70 (2021), 1–18.

[11] Stephane Beauregard and Harald Haas. 2006. Pedestrian Dead Reckoning: A Basis for Personal Positioning. In *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication (WPNC).* IEEE, 27–35.

[12] Yanis Boussad, Mohamed Naoufal Mahfoudi, Arnaud Legout, Leonardo Lizzi, Fabien Ferrero, and Walid Dabbous. 2021. Evaluating Smartphone Accuracy for RSSI Measurements. *IEEE Transactions on Instrumentation and Measurement.* 70 (2021), 1–12.

[13] Christopher JC Burges. 2010. From Ranknet to Lambdarank to Lambdamart: An Overview. *Learning* 11, 23-581 (2010), 81.

[14] Liqiong Chang, Jie Xiong, Yu Wang, Xiaojiang Chen, Junhao Hu, and Dingyi Fang. 2017. iUpdater: Low Cost RSS Fingerprints Updating for Device-Free Localization. In *Proceedings of the 37th International Conference on Distributed Computing Systems (ICDCS).* IEEE, 900–910.

[15] Yogita Chapre, Aleksandar Ignjatovic, Aruna Seneviratne, and Sanjay Jha. 2014. CSI-MIMO: Indoor Wi-Fi Fingerprinting System. In *39th Annual IEEE Conference on Local Computer Networks (LCN).* IEEE, 202–209.

[16] Dongyao Chen, Kang G Shin, Yurong Jiang, and Kyu-Han Kim. 2017. Locating and Tracking BLE Beacons with Smartphones. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies (CONEXT).* ACM, 263–275.

[17] Pan Chen, Jianga Shang, and Fuqiang Gu. 2020. Learning RSSI Feature via Ranking Model for Wi-Fi Fingerprinting Localization. *IEEE Transactions on Vehicular Technology.* 69, 2 (2020), 1695–1705.

[18] Yin Chen, Dimitrios Lymberopoulos, Jie Liu, and Bodhi Priyantha. 2012. FM-based Indoor Localization. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys).* ACM, 169–182.

[19] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS).* ACM, 7—-10.

[20] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N Padmanabhan. 2010. Indoor Localization without The Pain. In *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking (MobiCom).* ACM, 173–184.

[21] Jaewoo Chung, Matt Donahoe, Chris Schmandt, Ig-Jae Kim, Pedram Razavai, and Micaela Wiseman. 2011. Indoor Location Sensing using Geo-magnetism. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys).* ACM, 141–154.

[22] M. Ciurana, F. Barcelo-Arroyo, and F. Izquierdo. 2007. A Ranging Method with IEEE 802.11 Data Frames for Indoor Localization. In *Wireless Communications and Networking Conference (WCNC).* IEEE, 2092–2096.

[23] Patrick Dickinson, Gregorz Cielniak, Olivier Szymanezyk, and Mike Mannion. 2016. Indoor Positioning of Shoppers using A Network of Bluetooth Low Energy Beacons. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN).* IEEE, 1–8.

[24] Yi Ding, Dongzhe Jiang, Yunhuai Liu, Desheng Zhang, and Tian He. 2022. SmartLOC: Indoor Localization with Smartphone Anchors for On-Demand Delivery. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 5, 4 (2022), 1–24.

[25] Wan Du, Panrong Tong, and Mo Li. 2020. UniLoc: A Unified Mobile Localization Framework Exploiting Scheme Diversity. *IEEE Transactions on Mobile Computing* 20, 7 (2020), 2505–2517.

[26] Rizanne Elbakly, Moustafa Elhamshary, and Moustafa Youssef. 2018. HyRise: A Robust and Ubiquitous Multi-sensor Fusion-based Floor Localization System. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 2, 3 (2018), 1–23.

[27] Sinem Coleri Ergen, Huseyin Serhat Tetikol, Mehmet Kontik, Raffi Sevlian, Ram Rajagopal, and Pravin Varaiya. 2013. RSSI-fingerprinting-based Mobile Phone Localization with Route Constraints. *IEEE Transactions on Vehicular Technology* 63, 1 (2013), 423–428.

[28] Ramsey Faragher and Robert Harle. 2015. Location Fingerprinting With Bluetooth Low Energy Beacons. *IEEE Journal on Selected Areas in Communications* 33, 11 (2015), 2418–2428.

[29] Chen Feng, Wain Sy Anthea Au, Shahrokh Valaee, and Zhenhui Tan. 2011. Received-signal-strength-based Indoor Positioning using Compressive Sensing. *IEEE Transactions on mobile computing* 11, 12 (2011), 1983–1993.

[30] Sebastian Gansemer, Uwe Großmann, and Syuzanna Hakobyan. 2010. RSSI-based Euclidean Distance Algorithm for Indoor Positioning Adapted for the Use in Dynamically Changing WLAN Environments and Multi-level Buildings. In *2010 International Conference on Indoor Positioning and Indoor Navigation.* IEEE, 1–6.

[31] Jon Gjengset, Jie Xiong, Graeme McPhillips, and Kyle Jamieson. 2014. Phaser: Enabling Phased Array Signal Processing on Commodity WiFi Access Points. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking (MobiCom).* ACM, 153–164.

[32] Omar Hashem, Moustafa Youssef, and Khaled A. Harras. 2020. WiNar: RTT-based Sub-meter Indoor Localization using Commercial Devices. In *IEEE International Conference on Pervasive Computing and Communications (PerCom).* IEEE, 1–10.

[33] AKM Mahtab Hossain and Wee-Seng Soh. 2015. A Survey of Calibration-free Indoor Positioning Systems. *Computer Communications* 66 (2015), 1–13.

[34] Yuming Hu, Feng Qian, Zhimeng Yin, Zhenhua Li, Zhe Ji, Yeqiang Han, Qiang Xu, and Wei Jiang. 2022. Experience: Practical Indoor Localization for Malls. In *Proceedings of the 28th Annual International Conference on Mobile Computing and Networking (MobiCom).* ACM, 12 pages.

[35] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).

[36] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. 2015. SpotFi: Decimeter Level Localization Using WiFi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM).* ACM, 269–282.

[37] Swarun Kumar, Stephanie Gil, Dina Katabi, and Daniela Rus. 2014. Accurate Indoor Localization with Zero Start-up Cost. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking (MobiCom).* ACM, 483–494.

[38] Wen Liu, Qianqian Cheng, Zhongliang Deng, Hong Chen, Xiao Fu, Xinyu Zheng, Shixuan Zheng, Cunzhe Chen, and Shuo Wang. 2019. Survey on CSI-based Indoor Positioning Systems and Recent Advances. In *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN).* 1–8.

[39] Chengwen Luo, Hande Hong, Mun Choon Chan, Jianqiang Li, Xinglin Zhang, and Zhong Ming. 2017. MPiLoc: Self-calibrating Multi-floor Indoor Localization Exploiting Participatory Sensing. *IEEE Transactions on Mobile Computing* 17, 1 (2017), 141–154.

[40] Dimitrios Lymberopoulos, Jie Liu, Xue Yang, Romit Roy Choudhury, Vlado Handziski, and Souvik Sen. 2015. A Realistic Evaluation and Comparison of Indoor Location Technologies: Experiences and Lessons Learned. In *Proceedings of the 14th international Conference on Information Processing in Sensor Networks (IPSN).* ACM, 178–189.

[41] Alex T Mariakakis, Souvik Sen, Jeongkeun Lee, and Kyu-Han Kim. 2014. Sail: Single Access Point-based Indoor Localization. In *Proceedings of the 12th International Conference on Mobile Systems, Applications, and Services (MobiSys).* ACM, 315–328.

[42] Israel Martin-Escalona and Enrica Zola. 2020. Ranging Estimation Error in WiFi Devices Running IEEE 802.11mc. In *IEEE Global Communications Conference (GLOBECOM).* IEEE, 1–7.

[43] Eduardo Navarro, Benjamin Peuker, Michael Quan, Advisors Christopher Clark, and Jennifer Jipson. 2010. *Wi-fi Localization using RSSI Fingerprinting.* Ph.D. Dissertation. California Polytechnic State University.

[44] Anshul Rai, Krishna Kant Chintalapudi, Venkata N Padmanabhan, and Rijurekha Sen. 2012. Zee: Zero-effort Crowdsourcing for Indoor Localization. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom).* ACM, 293–304.

[45] Matteo Ridolfi, Abdil Kaya, Rafael Berkvens, Maarten Weyn, Wout Joseph, and Eli De Poorter. 2021. Self-Calibration and Collaborative Localization for UWB Positioning Systems: A Survey and Future Research Directions. *Comput. Surveys* 54, 4 (2021).

[46] Nirupam Roy, He Wang, and Romit Roy Choudhury. 2014. I Am A Smartphone and I Can Tell My User's Walking Direction. In *Proceedings of the 12th International Conference on Mobile Systems, Applications, and Services (MobiSys).* ACM, 329–342.

[47] Jochen Seitz, Jasper Jahn, Javier Gutiérrez Boronat, Thorsten Vaupel, Steffen Meyer, and Jörn Thielecke. 2010. A Hidden Markov Model for Urban Navigation based on Fingerprinting and Pedestrian Dead Reckoning. In *13th International Conference on Information Fusion*. IEEE, 1–8.

[48] Masato Sugasaki and Masamichi Shimosaka. 2017. Robust Indoor Localization across Smartphone Models with Ellipsoid Features from Multiple RSSIs. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 1, 3 (2017).

[49] Peng Tang, Zhiqing Huang, Jun Lei, and Yue Guo. 2016. Wi-Fi Fingerprint Localization using RSSI-probability Radio Map and AP Weight Clustering. *Journal of Advances in Computer Networks* 4, 2 (2016), 121–124.

[50] Xinyu Tong, Yang Wan, Qianru Li, Xiaohua Tian, and Xinbing Wang. 2021. CSI Fingerprinting Localization With Low Human Efforts. *Transactions on Networking* 29, 1 (2021), 372–385.

[51] Deepak Vasisht, Swarun Kumar, and Dina Katabi. 2016. Decimeter-Level Localization with A Single WiFi Access Point. In *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation (NSDI)*. USENIX, 165–178.

[52] Bang Wang, Qiuyun Chen, Laurence T. Yang, and Han-Chieh Chao. 2016. Indoor smartphone localization via fingerprint crowdsourcing: challenges and approaches. *IEEE Wireless Communications* 23, 3 (2016), 82–89.

[53] Ju Wang, Hongbo Jiang, Jie Xiong, Kyle Jamieson, Xiaojiang Chen, Dingyi Fang, and Binbin Xie. 2016. LiFS: Low Human-effort, Device-free Localization with Fine-grained Subcarrier Information. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 243–256.

[54] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. 2017. CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach. *IEEE Transactions on Vehicular Technology* 66, 1 (2017), 763–776.

[55] Yixin Wang, Qiang Ye, Jie Cheng, and Lei Wang. 2015. RSSI-Based Bluetooth Indoor Localization. In *11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*. IEEE, 165–171.

[56] Jie Xiong and Kyle Jamieson. 2013. ArrayTrack: A Fine-Grained Indoor Location System. In *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX, 71–84.

[57] Jie Xiong, Karthikeyan Sundaresan, and Kyle Jamieson. 2015. Tonetrack: Leveraging Frequency-agile Radios for Time-based Indoor Wireless Localization. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 537–549.

[58] Qiang Xu and Rong Zheng. 2016. MobiBee: A Mobile Treasure Hunt Game for Location-Dependent Fingerprint Collection. In *Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct (UbiComp)*. ACM, 1472–1477.

[59] Zheng Yang, Chenshu Wu, and Yunhao Liu. 2012. Locating in Fingerprint Space: Wireless Indoor Localization with Little Human Intervention. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 269–280.

[60] Zuwei Yin, Chenshu Wu, Zheng Yang, and Yunhao Liu. 2017. Peer-to-peer Indoor Navigation using Smartphones. *IEEE Journal on Selected Areas in Communications* 35, 5 (2017), 1141–1153.

[61] Simon Yiu, Marzieh Dashti, Holger Claussen, and Fernando Perez-Cruz. 2017. Wireless RSSI Fingerprinting Localization. *Signal Processing* 131 (2017), 235–244.

[62] Xianan Zhang, Wei Wang, Xuedou Xiao, Hang Yang, Xinyu Zhang, and Tao Jiang. 2020. Peer-to-peer Localization for Single-antenna Devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 4, 3 (2020), 1–25.

[63] Minghui Zhao, Tyler Chang, Aditya Arun, Roshan Ayyalasomayajula, Chi Zhang, and Dinesh Bharadia. 2021. ULoc: Low-Power, Scalable and Cm-Accurate UWB-Tag Localization and Tracking for Indoor Applications. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 5, 3 (2021), 1–31.

[64] Pengfei Zhou, Yi Ding, Yang Li, Mo Li, Guobin Shen, and Tian He. 2022. Experience: Adopting Indoor Outdoor Detection in On-demand Food Delivery Business. In *Proceedings of the 28th Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 12 pages.

[65] Pengfei Zhou, Yuanqing Zheng, Zhenjiang Li, Mo Li, and Guobin Shen. 2012. IODetector: A Generic Service for Indoor Outdoor Detection. In *10th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 113–126.