LAAFU: Indoor Localization and Fingerprint Update with Altered Access Points

Suining He Wenbin Lin S.-H. Gary Chan

Abstract—Fingerprinting is a promising approach for indoor localization due to its ease of deployment and high accuracy. As the signals from access points (APs) may change (due to, for examples, AP movement or power adjustment), the offline site survey often needs to be regularly conducted to maintain localization accuracy, which is costly and time-consuming.
In this paper, we propose LAAFU (Localization with Altered APs and Fingerprint Updating), which achieves both accurate indoor localization and automatic fingerprint update in the presence of altered APs without the need of extra site survey. LAAFU is based on implicit crowdsourcing. Using novel subset sampling, it is able to efficiently identify the altered APs and filter them out before localization, hence achieving high accuracy. With the client locations, the fingerprint signal due to the altered APs can be adaptively and transparently updated using the non-parametric Gaussian process regression method. We have implemented LAAFU and conducted extensive experiments in our campus. Our experimental results in our university campus, the international airport and a premium shopping mall further show that LAAFU is robust to altered AP signal changes to achieve high localization accuracy, and its fingerprint database is able to adapt to the current signal environment.

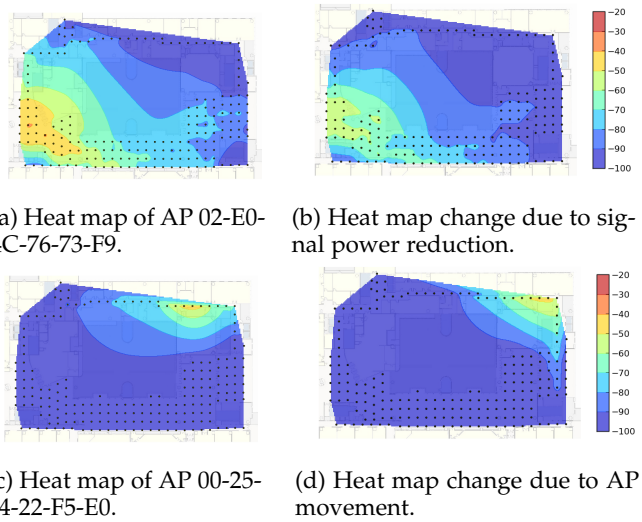
Index Terms—Indoor localization, fingerprinting, clustering, altered access point, database update, Gaussian process.

1 Introduction

Indoor location-based services (LBS) has attracted wide attention recently, and fingerprint-based techniques have been studied extensively for practical deployment. There are typically two phases in fingerprint-based localization, namely offline site survey and online location query. In the offline phase, a site survey is conducted to collect fingerprints at known physical locations called reference points (RPs). Each fingerprint is a vector of received signal strength (RSS) values from Wi-Fi access points (APs). The RSS values and their associated locations are stored in a fingerprint database. In the online phase, a mobile client measures the RSS values at its location. Upon receiving the signal measurement, the server matches it with its database to return the client location.

However, due to AP movement, power adjustment, wall partitioning, etc, the signal from APs may change significantly. Figure 1 shows how the heat map changes with two altered APs, for two site surveys conducted within three months. From the marked differences in the heat map, localization accuracy would be adversely affected. In order to keep the fingerprint database updated, usually a site survey has to be conducted again. This is, however, labor-intensive and time-consuming, especially when the APs may be altered frequently (usually over a month or so).

Normally, the number of altered APs is small (about 1 to 3) as compared with the total APs detected at a location (around 27 in our case). Given the altered APs, we need to filter them out and estimate the location more accurately. We have conducted an experiment on an RSS vector with two



(a) Heat map of AP 02-E0-4C-76-73-F9.

(b) Heat map change due to signal power reduction.

(c) Heat map of AP 00-25-84-22-F5-E0.

(d) Heat map change due to AP movement.

Fig. 1: Heat maps on Sep. 13, 2014 (left) and the changes on Dec. 22, 2014 (right) due to AP alteration.

altered APs. After generating several random subsets (i.e., a subset of detected APs forms a new vector), we compute the locations by a certain fingerprint-based localization algorithm. Figure 2 shows the locations, which are estimated using the RSS subsets without any altered APs, tend to *cluster* together around the ground truth location. To the contrary, locations estimated from the subset consisting of altered APs tend to *disperse* in nature. Based on this, the client location can be identified if we can find the dense cluster. Furthermore, given such client locations, the fingerprint database can be updated with the RSS vectors received by the clients.

Motivated by above observations, we propose **LAA-**

• *Suining He, Wenbin Lin and S.-H. Gary Chan are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong, China. E-mail:{sheaa,wlinab,gchan}@cse.ust.hk*

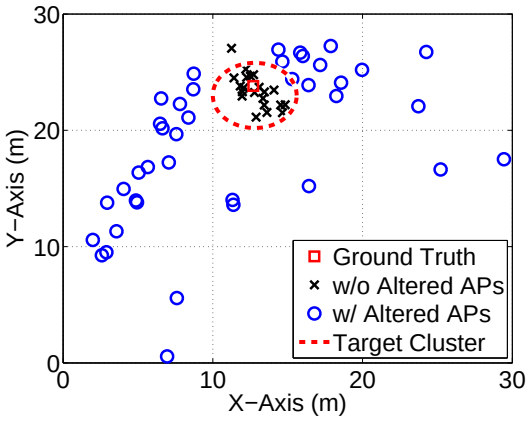


Fig. 2: Locations estimated using RSS subsets with or without altered APs.

FU, which achieves **L**ocalization with **A**ltered **A**Ps and **F**ingerprint **U**pdating. LAAFU achieves both accurate indoor localization and automatic fingerprint update in the presence of possible altered APs without the need of extra site survey. LAAFU first detects whether there is any altered AP in the RSS vector with a fast algorithm. If no altered AP is detected, it simply runs a fingerprint-based localization algorithm to estimate the client's location. Otherwise, using a novel subset sampling, LAAFU filters out altered APs with an efficient clustering algorithm and finds the correct location of the client. To update the fingerprint, LAAFU employs the non-parametric Gaussian process regression method [1]. With such implicit crowdsourcing approach, LAAFU can transparently adapt the fingerprint of the altered APs.

In this paper, we have the following contributions:

- *Efficient Fast Detection:* We propose a simple and novel fast detection algorithm which classifies the existence of altered APs. We partition the RSSI vector into multiple subsets and check the dispersiveness of these estimations. We successfully reduce the false alarms and improve the system practicability.
- *Robust Clustering-based Subset Localization Algorithm:* To filter the incorrect estimations resulted from altered APs, we propose a novel clustering-based localization using RSS subsets. We implement affinity clustering and kernel-based cluster weighting to accurately classify the altered APs and achieve high localization accuracy.
- *Joint Fingerprint Update Using Gaussian Process:* Previous fingerprint update is clumsily based on single point update, which does not consider the overall change of altered APs. In this work, we propose a Gaussian-process-based fingerprint update scheme to jointly reconstruct the altered database, which is more scalable and robust in real deployment.

Note that LAAFU is independent of, and hence may be used with, any localization algorithm [2] or any fingerprint signal [3]. We have implemented LAAFU with a simple KNN localization algorithm and conducted extensive experiments in our campus, the international airport and a leading shopping mall. Our results demonstrate that LAAFU is robust to altered AP signal changes to achieve high localization accuracy, as compared with the traditional and advanced fingerprint-based localization techniques.

The rest of this paper is organized as follows. We present related work in Section 2 and system overview in Section 3. Section 4 discusses the fast detection algorithm of altered APs in LAAFU. Localization with altered APs and fingerprint database update are discussed in Sections 5 and 6, respectively. We present illustrative experimental results in Section 7, and finally conclude in Section 8.

2 Related Work

Fingerprint-based techniques have attracted much attention recently, including deterministic and probabilistic methods [4]. Deterministic techniques [5], [6], [7] represent the signal strength of an AP at a location by a scalar value and use non-probabilistic approaches to estimate the user location. The pioneer work, RADAR [5], uses nearest neighborhood method to look up the user location from database. On the other hand, probabilistic techniques [8], [9] store the signal strength distributions in the fingerprint database and use probabilistic approaches to estimate the user location. Horus in [8] is a typical system using a Bayesian network method. As LAAFU focuses on altered APs filtering and fingerprint database update, our work is orthogonal to these localization techniques, any of which may be adopted in our system for location estimation.

Fingerprint database construction has been recently studied in [10], [11], [12]. WiFi-SLAM [10] utilizes the extra robot sensors to reconstruct the signal map. In [11], a ray-tracing simulation software with a building map is used to simulate the signal distribution, given the measured signals from deployed sniffers. WILL [12] explores the possibility to combine the movements of surveyor and Wi-Fi fingerprints to construct the radio map with lower survey cost. Different from above works, LAAFU does not require any extra infrastructures or calibrated motion sensors, achieving better scalability for real deployment.

Crowdsourcing has also attracted much attention [13], [14], [15], [16] in recent years. UnLoc [16] constructs the walking trajectories using the inertial smartphone sensors and the indoor signal landmark. Thus, UnLoc pinpoints the location of client and collects signal data used as fingerprint. While these approaches eliminate the need of surveyors, they require intrusive and explicit user participation as well as the prior indoor map. Their power consumption may be also high due to the inertial sensors. In contrast, LAAFU achieves high localization accuracy and automatic fingerprint update by the measured Wi-Fi (RF) signals from the clients.

Fingerprint adaptation have been studied by some works [17], [18], [19], [20]. The work in [17] introduces a modified Bayesian regression algorithm to estimate the fingerprint values at RPs based on input data. Nevertheless, it relies on additionally deployed referenced points while LAAFU adapts the database without any extra network hardware. LEMT [18] achieves adaptive temporal radio map by learning a functional relationship for one location and its neighbors based on a model tree method. Similarly, transfer learning techniques has also been applied to adapt RSS measurements [19], [20]. They accommodate the fingerprint

variations over different time as they both assume the relationship of RSS values between neighboring locations is stable. However, the underlying relationship does not hold once the altered APs occur and they cannot address our problem.

Gaussian process (GP) is a nonparametric nonlinear probabilistic modeling scheme that can flexibly adapt to the complex signal propagation indoors [10], [17], [21], [22], [23]. The uncertainty estimation for prediction, taking into account the local density of input data and the signal noise, also makes it suitable in RSS modeling. [10], [21], [22] require surveyors to collect calibration data and then apply Gaussian process model to construct fingerprint databases while the authors in [23] make use of a mobile robot. LAAFU can locate the clients accurately and update the database only using RSS measured by clients, without the need of surveyors or extra infrastructures.

3 System Overview

Figure 3 shows the implementation framework of LAAFU, which consists of *fast detection*, *localization with altered APs*, and *fingerprint database update*. We discuss briefly these three phases in LAAFU as follows:

- *Fast Detection*: In practical deployment, the installation and configuration of a given AP does not change frequently. Furthermore, most APs installed in the survey site only cover part of the whole site, and therefore their alteration does not influence the localization if the target does not detect them. Therefore, a conventional fingerprinting localization technique already suffices to provide positioning service in normal cases. To ensure the efficiency when no significant AP alteration exists, the fast detection phase in LAAFU is designed to initially determine whether there is an altered AP or not among the RSS vector, measured by client at a location. If alteration may not exist, LAAFU conducts the existing localization without clustering-based subset sampling. Otherwise, LAAFU moves to the next phase, localization with altered APs, to search for the client's location.
- *Localization with Altered APs*: This phase aims at achieving robust localization in the presence of altered APs. Originally, LAAFU randomly generates sufficient number of RSS subset samples, and then estimates their corresponding locations. At this stage, the locations, calculated using the subsets without altered APs, form a dense cluster while others spread with a wide margin. LAAFU clusters over these estimated locations, and subsequently finds out the dense cluster, whose centroid's location yields the client's location. Beyond that, in this phase, LAAFU also identifies the altered APs, whose fingerprint values would be adapted in the update phase.
- *Fingerprint Database Update*: We aims at updating the RSSs of the altered APs in the fingerprint database. LAAFU associates the RSS with the estimated locations and feeds them into the GP regression. Using the Gaussian process modeling, we jointly update the RSS and the signal patterns based on the current environment.

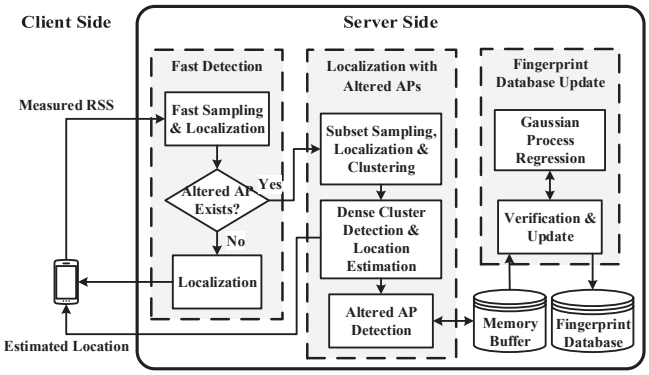


Fig. 3: System overview of LAAFU.

4 Fast Detection

We first present the algorithmic details of fast detection in this section. Table 1 lists the important notations.

Based on our deployment experience, AP alteration happens at a relatively large time scale (usually by weeks or months). Therefore, it is unnecessary to execute the phase of localization with altered APs, which has higher computational cost, and the fingerprint database update phase. We hence present fast detection to early detect the presence of altered APs in each location query. If no altered AP is observed, LAAFU runs traditional fingerprint-based localization algorithm to estimate the client's location and then returns the result with no further processing. If fingerprint inconsistency exists, LAAFU performs the localization with altered APs phase.

In the following, we present how to conduct fast subset sampling (Section 4.1), localization using these subsets (Section 4.2) and detect the AP alteration (Section 4.3).

4.1 Fast Subset Sampling

In this section, we present how to generate the subsets of the RSS vectors obtained from the target.

Specifically, let v_i be the target-measured RSS (mW) from AP i. Then the measured RSS vector at the target is defined as

$$\mathbf{V} = \{v_1, v_2, \ldots, v_i, \ldots v_P\}, \tag{1}$$

where $1 \leq i \leq P$ and P is the total number of APs in the whole site of interest. Note that $v_i = 0$ if the target does not detect AP i.

In order to generate one RSS subset vector, LAAFU firstly extracts the APs, which can be detected by the client at that location. Let $\mathbf{A}$ be a vector of MAC addresses, where

$$ap_i \in \mathbf{A}, \tag{2}$$

iff $v_i > 0$. Then LAAFU constructs a subset vector of MAC addresses, expressed as

$$\mathbf{A}_s \subseteq \mathbf{A}, \tag{3}$$

where $\mathbf{A}_s \neq \emptyset$. And hence, the RSS subset vector $\mathbf{V}_s$ is defined as

$$\mathbf{V}_s = \{v'_1, v'_2, \ldots, v'_i, \ldots v'_P\}, \tag{4}$$

where $v'_i = v_i$ if $ap_i \in \mathbf{A}_s$, otherwise 0.

TABLE 1: Major symbols used in LAAFU.

Notation	Definition
$\mathbf{V}$	RSS vector measured by client
ap_i	MAC address of AP i
$\mathbf{A}$	MAC of APs detected by client
$\mathbf{A}_s$	MAC address subset vector from $\mathbf{A}$
$\mathbf{V}_s$	RSS subset vector generated based on $\mathbf{A}_s$
$\mathbf{l}$	Physical location
$\mathbf{F}_j$	Fingerprint, RSS vector measured at RP j
v_{ij}	RSS value received from AP i at RP j
k	Number of neighbors in WKNN
γ	Distance threshold in Fast Detection
P	Number of APs in the whole site
R	Number of RPs in the whole site
M	Number of RSS subset vectors generated
Q	Number of nearest RPs used in cluster similarity
b	Bandwidth used in penalty term
W	Update interval for fingerprint database update
N	Training data size for signal regression
κ	Variance factor used in fingerprint database update
λ	Fingerprint database update weight

For the efficient performance in fast detection, LAAFU generates only a few random subset samples of the measured RSS vector to see whether the altered AP exists. Note that, if an altered AP exists, it may occur in all these RSS subset vector samples, which probably all result in similar estimated locations and lead to the wrong decision in fast detection. To reduce the incorrect decision, we construct the RSS subset vectors as follows. LAAFU randomly divides the MAC address vector $\mathbf{A}$ into two parts with the even sizes and obtain one two-part partition $\{\mathbf{A}_1, \mathbf{A}_2\}$, such that,

$$|\mathbf{A}_1| = |\mathbf{A}_2| = \frac{1}{2}|\mathbf{A}|, \tag{5}$$

where

$$\mathbf{A}_1 \cup \mathbf{A}_2 = \mathbf{A}, \quad \mathbf{A}_1 \cap \mathbf{A}_2 = \emptyset. \tag{6}$$

Then based on the $\mathbf{A}_1$ and $\mathbf{A}_2$, LAAFU constructs two RSS subset vectors from $\mathbf{V}$, respectively, by Equation (4). Similarly, three more RSS subsets are generated using the three partitions of $\mathbf{A}$. With the original measured RSS vector, we totally have six exclusive samples.

4.2 Localization with RSS Subsets

Given the generated RSS subset vector samples in the previous step, LAAFU implements weighted *k*-nearest-neighbor (WKNN) algorithm [7] to compute the locations for each of the RSS subset vectors. Note that LAAFU can be integrated with any other fingerprint-based localization algorithms [2].

Let R be the number of RPs in the survey site and j be the index of RP. Denote the 2-D coordinate of RP j as $\mathbf{l}_j = (l_j^1, l_j^2)$. Then the set of RPs is given by

$$\mathbf{L} = \{\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_j, \ldots, \mathbf{l}_R\}. \tag{7}$$

Similar to Equation (1), we denote the fingerprint at each RP j as

$$\mathbf{F}_j = \{v_1^j, v_1^j, \ldots, v_P^j\}. \tag{8}$$

And the set of fingerprints are given by

$$\mathbf{F} = \{\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_j, \ldots, \mathbf{F}_R\}, \tag{9}$$

Then we store $\mathbf{F}$ and $\mathbf{L}$ into the fingerprint database.

WKNN finds the top k nearest RPs whose fingerprints closely matches the target measured one. The comparison between RSS vectors $\mathbf{U}$ and $\mathbf{V}$ is based on cosine similarity, which is defined as

$$\cos(\mathbf{U}, \mathbf{V}) = \frac{\mathbf{U} \cdot \mathbf{V}}{|\mathbf{U}||\mathbf{V}|}. \tag{10}$$

Each of the k RPs is therefore assigned with weight, i.e.,

$$\omega_j = \cos(\mathbf{F}_j, \mathbf{V}). \tag{11}$$

WKNN computes the weighted sum of all the RP coordinates, and the estimated location $\hat{\mathbf{l}}$ is given by

$$\hat{\mathbf{l}} = \sum^{k} \frac{\omega_j}{\omega} \mathbf{l}_j, \tag{12}$$

where the normalizing factor ω is

$$\omega = \sum^{k} \omega_j. \tag{13}$$

4.3 AP Alteration Detection

The AP alteration leads to the dispersion in the locations estimated from RSS subsets, which has been illustrated in Figure 1. Given the above six estimated locations, Euclidean distance is then applied to measure the mutual dispersion between each pair $\mathbf{l}_i$ and $\mathbf{l}_j$, i.e.,

$$\| \mathbf{l}_i - \mathbf{l}_j \| = \sqrt{(\mathbf{l}_i - \mathbf{l}_j) \cdot (\mathbf{l}_i - \mathbf{l}_j)^T}. \tag{14}$$

Then we average all the mutual Euclidean distance. If the average mutual distance is less than a certain threshold γ, we conclude that AP alteration may not exist. Otherwise, we conduct further processing, which will be described in the following sections.

The complexity of fast detection is analyzed as follows:

- *Fast detection* which needs to randomly permute the APs ($\mathbf{A}$) and construct RSS subsets, each of which costs computational time $\mathcal{O}(|A|)$ and $\mathcal{O}(P)$, respectively ($|A| \leq P$). Therefore, it takes $\mathcal{O}(P)$ in all.
- *Subset localization*, where KNN takes $O\big(R(P+\log k)\big)$.

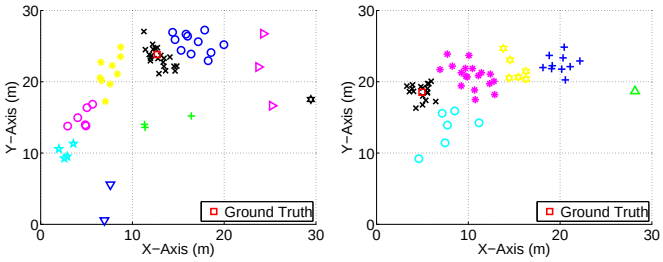
The decision takes $\mathcal{O}(1)$, and the whole fast detection takes $\mathcal{O}(RP)$.

5 Localization with Altered APs

In this section, we discuss how to search for location when a few altered APs may exist in the measured RSS vector after the fast detection. We present as follows how to extract these altered APs for later fingerprint update.

Recall that in Section 1, locations estimated from RSS subset vectors without altered APs tend to be close and hence form a dense cluster, otherwise they tend to disperse. LAAFU needs to assign the locations, estimated using subset samples, into different clusters, and then distinguishes the dense cluster, whose centroid is around the client location. Other disperse clusters may contain the altered APs.

Based on above observations, we first present how to conduct subset sampling, location estimation and clustering in Section 5.1. Then we describe how to detect the AP alteration in Section 5.2.



(a) Clustering on locations into nine clusters.

(b) Clustering on locations into six clusters.

Fig. 4: Illustration of location clustering.

5.1 Subset Sampling, Localization & Clustering

We present as follows how to conduct subset sampling, localization and clustering:

1) *RSS Subset Sampling*: The subset sampling is similar to the one in fast detection. To more effectively find the altered APs, more subsets are needed. The complexity to generate all the possible subsets of the measured RSS vector $\mathbf{V}$, is exponential and expensive. To reduce the computation, we consider as follows generating certain number of subset samples.
   Let $\mathbf{A}$ be the list of APs detected by the target. To construct a subset $\mathbf{A}_s$, we toss a *fair coin* for each $ap_i \in \mathbf{A}$ to decide whether to put it into $\mathbf{A}_s$ or not, i.e., $ap_i \in \mathbf{A}_s$ if coin i is head.
   Note that $\mathbf{A}_s$ is discarded if $|\mathbf{A}_s|$ is too small for sufficient localization (say, $|\mathbf{A}_s| \geq 3$). Given $\mathbf{A}_s$, LAAFU generates one RSS subset vector sample $\mathbf{V}_s$ using Equation (4). We repeat the above process until totally M RSS subset vector samples are generated, which are then used to estimate locations with WKNN (presented in Section 4.2).
2) *Location Clustering*: To effectively discover the altered APs, we consider clustering the estimated locations given above. As the dispersion of estimated locations is random, it is unlikely to initially decide cluster number. To address this, we implement the affinity propagation clustering [24]. Figure 4(a) and Figure 4(b) show the results using affinity propagation clustering, where different number of clusters is generated adaptively. Note that any other suitable clustering algorithm can be applied in LAAFU.
   Specifically, affinity propagation method takes in an M-by-M square matrix of similarities between any two estimated locations as input, where the similarity, denoted as $s(i,j)$, is given by Euclidean distance between the estimated locations (Equation (14)). During the clustering, two kinds of messages, *responsibilities* and *availabilities* are exchanged between the locations:
   a) Responsibility $r(i,j)$, sent from location i to j, reflecting how proper j can serves as the centroid for i comparing with other potential centroids.
   b) Availability $a(i,j)$, sent from location j to i, revealing accumulatively how appropriate to choose point j as the centroid for i.

   The responsibility $r(i,j)$ is given by

$$r(i,j) = s(i,j) - \max_{j's.t.j' \neq j} \{a(i,j') + s(i,j')\}, \quad (15)$$

   where the availabilities $a(i,j)$ are all initialized to zero in the first iteration.
   Availability $a(i,j)$ is defined as

$$a(i,j) = \min\Big\{0, r(j,j) + \sum_{i's.t.i' \notin \{i,j\}} \max\{0, r(i',j)\}\Big\}, \quad (16)$$

   where $i \neq j$. And the self-availability $a(i,i)$ is updated differently as

$$a(i,i) = \sum_{i's.t.i' \neq i} \max\{0, r(i',i)\}. \quad (17)$$

   Therefore, $r(i,j)$'s and $a(i,j)$'s are iteratively updated in order to maximize the net similarity, denoted as τ_i, at each location i, i.e.,

$$\tau_i = \max_j \{a(i,j) + r(i,j)\}. \quad (18)$$

   If $i = j$, i is identified as the centroid of a cluster. Otherwise i is classified as the cluster whose centroid is j. Such iteration ends when the clustered points do not change.
3) *Dense Cluster Detection & Location Estimation*: Given the clustered locations, LAAFU distinguishes the dense cluster from the others. We find the dense one based on the following two rules:
   a) *High average similarity*: In the dense cluster, all the estimated locations are close to the client's one in signal space. In other words, their corresponding RSS subsets should have high similarity with the target RSS vector. We measure the closeness using cosine similarity in Equation (10).
   Specifically, for each cluster C, LAAFU selects several nearest RPs around the centroid of this cluster using Euclidean distance (Equation 14). Then, we compute the average of similarities between each subset vector $\mathbf{V}_i^c$ in C and each $\mathbf{F}_j^c$ among the nearest RPs as the similarity of C, i.e.,

$$\varrho_c = \frac{1}{|C|Q} \sum_{i}^{|C|} \sum_{j}^{Q} \cos(\mathbf{V}_i^c, \mathbf{F}_j^c), \quad (19)$$

   where $|C|$ is the number of location points in cluster C and Q is the number of nearest RPs around the centroid.
   b) *Large cluster size*: Besides average similarity in signal space, LAAFU also considers the size of each cluster. It is mainly because small cluster may still lead to high average similarity, and they are likely to deviate from other locations due to presence of altered APs. To address this, we use Gaussian kernel function [25] to transform the cluster size into a penalty term, ranging from zero to one, i.e.,

$$\nu_c = \exp\left(-\frac{(|C| - |C|_{min})^2}{2b^2}\right), \quad (20)$$

   where the bandwidth parameter b controls the kernel sensitivity and $|C|_{min}$ represents the size of the

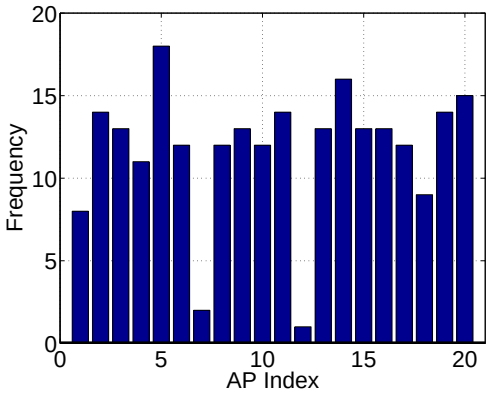


Fig. 5: Frequency in dense cluster versus index of APs detected by the client.

smallest cluster. In other words, it penalizes more as the cluster size decreases.

Jointly considering the above rules, we have the final score of cluster C as

$$\zeta_c = \varrho_c - \nu_c. \tag{21}$$

The cluster with the highest score is chosen as our target dense cluster. Its centroid (average of 2-D coordinates) is therefore returned as the estimated location of the target.

5.2 Altered AP Detection

Altered APs are likely to be excluded from subsets within the dense cluster, while the unaltered ones are likely to distributed evenly inside. To classify them, for each $ap_i \in \mathbf{A}$ in the selected dense cluster, LAAFU counts the number of RSS subset vectors which include AP i, as the frequency of ap_i. Figure 5 shows a counting result, where the client detects overall 20 APs, while the size of the dense cluster is 23. As the frequency of the altered AP is numerically distant from those of unaltered ones, we may observe two-class clustering problem in one dimension, which can be efficiently solved using Jenks natural breaks optimization method [26].

LAAFU begins the detection by sorting the frequency f in an increasing order. Next, it divides the ordered data into two classes, denoted as $\mathcal{C}_1$ and $\mathcal{C}_2$, and then calculates the *sum of squared deviations from the class means* (SDCM) as

$$SDCM = \sum_{i=1}^{2} \sum_{f \in \mathcal{C}_i} (x - \bar{f})^2, \tag{22}$$

where $\bar{f}$ is the mean of f's within class $\mathcal{C}_i$. LAAFU checks all possible combinations, which is linear with number of f's. After all combinations are examined, the break point with the lowest SDCM is selected, which means the smallest frequency variation within class. Then LAAFU marks the APs in the low class with smaller frequencies as altered ones.

To prevent unaltered APs from mislabeling, we look at long term reports in a sliding window. LAAFU records the times of APs being recognized as altered in the site. Given W location queries from the clients, LAAFU reports the times of each AP, ranging from 0 to W. These counts are clustered again using above method into two classes, and the APs within the more reporting times are therefore classified as altered.

We briefly analyze the time complexity as follows.

- In *RSS subset sampling*, LAAFU takes $\mathcal{O}(|\mathbf{A}|)$ to toss a coin for each $ap_i \in \mathbf{A}$ in subset sampling, and $\mathcal{O}(P)$ for one RSS subset vector. With M RSS subset vectors, subset sampling takes overall $\mathcal{O}(MP)$ and WKNN localization requires $\mathcal{O}(MRP)$ time (WKNN positioning takes $\mathcal{O}(RP)$ for each subset).
- *Location clustering* takes $\mathcal{O}(IM^2)$, where I is the number of iterations. *Dense cluster detection* takes $\mathcal{O}\big(M(R\log Q + QP)\big)$, as for each cluster c it takes $\mathcal{O}(R\log Q)$ to find the Q nearest RPs (using a heap with size Q), and $\mathcal{O}(|C|QP)$ in score computation. To summarize, the whole online localization needs $\mathcal{O}(MRP)$ time.
- For *altered AP detection*, with at most P APs, sorting costs $\mathcal{O}(P\log P)$ time, SDCM calculation is bounded by $\mathcal{O}(P)$ as there are $\mathcal{O}(P)$ potential break points. Overall it takes $\mathcal{O}(P^2)$ to detect AP alteration.

6 Fingerprint Database Update

Given the discovered altered APs, we consider how to update their signal values. The RSS vectors measured by the clients capture the signal characteristics in the survey site, especially under crowdsourcing. In this section, we present a novel method to jointly update the signal map in the survey site, instead of updating the fingerprint points individually. Given the query data (measured RSS vectors) and the estimated locations, we update the signal map of altered APs, i.e., updating the signal values at the fingerprint points.

We first introduce the GP regression formulation in Section 6.1. Then we discuss how to estimate the hyperparameters in the formulation in Section 6.2. Finally, we present the verification and signal update in Section 6.3.

6.1 Gaussian Process Regression

The basic idea of the fingerprint update is to regress the signals within the survey site and build up the new fingerprint database. However, due to the wall partitioning and signal fluctuation, the signal propagation may have local patterns, such as increase with tunneling effect or drop after a wall. It is thus not effective to regress using simple propagation model which only consider overall patterns.

To address this, we implement Gaussian process (GP) which preserves the overall signal characteristics while adapting fingerprints towards the local distribution.

1) *Basics in Formulating GP:* We first start from a standard linear signal regression model with Gaussian noise

$$v = f(\mathbf{l}) + \varepsilon, \tag{23}$$

where $\mathbf{l}$ is the input 2-D location (here we consider regressing the signals floor by floor), v is the target RSS value and $f(\cdot)$ is the transfer function. We assume RSS v differs from $f(\mathbf{l})$ by an additive noise due to

imperfection within modeling. ε is with zero mean and variance σ_n^2, i.e.,

$$\varepsilon \sim \mathcal{N}(0, \sigma_n^2). \tag{24}$$

A GP is a statistical distribution, from which any finite number of samples has a joint Gaussian distribution. It can be specified by mean $m(\mathbf{l})$ and covariance $k(\mathbf{l}, \mathbf{x}')$, which is given by

$$f(\mathbf{l}) \sim \mathcal{GP}\big(m(\mathbf{l}), k(\mathbf{l}, \mathbf{x}')\big). \tag{25}$$

The covariance function $k(\mathbf{l}, \mathbf{x}')$ indicates how two RSSs correlate depending on the input locations $\mathbf{l}$ and $\mathbf{x}'$. Note that $f(\mathbf{l}_i)$ and $f(\mathbf{l}_j)$ are unknown while noisy measurements v_i and v_j are given. We express the covariance between any two input locations as

$$cov(v_i, v_j) = k(\mathbf{l}_i, \mathbf{l}_j) + \sigma_n^2 \delta_{ij}, \tag{26}$$

where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. Let the N-by-2 matrix $\mathbf{L}$ be the aggregation of the N input vectors. Then, the covariance over $\mathbf{y}$, the vector of the RSSs corresponding to $\mathbf{L}$, is given by

$$cov(\mathbf{v}) = \mathbf{K} + \sigma_n^2 \mathbf{I}, \tag{27}$$

where $\mathbf{K}$ is the N-by-N covariance matrix over all N input vectors and $\mathbf{I}$ is the identity matrix of size N. The input RSS values follow a joint Gaussian distribution, i.e.,

$$\mathbf{v} \sim \mathcal{N}(m(\mathbf{L}), \mathbf{K} + \sigma_n^2 \mathbf{I}). \tag{28}$$

Conditioned on the training locations $\mathbf{L}$ and RSS $\mathbf{v}$ collected on them, we consider the RSS prediction at location $\star$ as

$$\mathbf{f}_\star|_{\mathbf{l}^\star, \mathbf{L}, \mathbf{v}} \sim \mathcal{N}(\mu_\star, \sigma_\star^2), \tag{29}$$

where the predictive mean RSS is

$$\mu_\star = m(\mathbf{l}^\star) + k(\mathbf{l}^\star, \mathbf{L})^T [\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1} \big(\mathbf{y} - m(\mathbf{L})\big), \tag{30}$$

and the predictive variance of the RSS is given by

$$\sigma_\star^2 = k(\mathbf{l}^\star, \mathbf{l}^\star) - k(\mathbf{l}^\star, \mathbf{L})^T [\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1} k(\mathbf{l}^\star, \mathbf{L}). \tag{31}$$

2) *Formulating RSS Estimation with Location Uncertainty:* Clearly, the input locations $\mathbf{l}$ also contains uncertainty due to location decision error. Such error usually results from uncertainty of finding site survey grids (offline phase) and target location errors (online phase). Therefore, we consider further beyond Equation (23) the input locations with noise, i.e.,

$$\mathbf{l} = \tilde{\mathbf{l}} + \boldsymbol{\varepsilon}_l, \tag{32}$$

where $\tilde{\mathbf{l}}$ is the actual locations and the noise is

$$\boldsymbol{\varepsilon}_l \sim \mathcal{N}(\mathbf{0}, \Sigma_l). \tag{33}$$

The 2-by-2 matrix Σ_l is a diagonal matrix assuming each dimension is independent, i.e.,

$$\Sigma_l[i, i] = \sigma_{l^i}^2, \tag{34}$$

and all the off-diagonal elements are zero. Therefore, the relationship between RSS signals and locations are represented as

$$v = f\left(\tilde{\mathbf{l}} + \boldsymbol{\varepsilon}_l\right) + \varepsilon. \tag{35}$$

For ease of computation [27], we expand the Taylor form and approximate the output RSS based on noisy input $\mathbf{l}$ as

$$v = f(\mathbf{l}) + \boldsymbol{\varepsilon}_l^T \partial \boldsymbol{f} + \varepsilon, \tag{36}$$

where the 2-dimension vector

$$\partial \boldsymbol{f} = \frac{\partial f(\mathbf{l})}{\partial \mathbf{l}} \tag{37}$$

is the derivative of GP function with respect to $\mathbf{l}$. Then the output function v can be reformulated as

$$v = f(\mathbf{l}) + \varepsilon_v, \tag{38}$$

where

$$\varepsilon_v \sim \mathcal{N}(0, \sigma_n^2 + \partial \boldsymbol{f}^T \Sigma_l \partial \boldsymbol{f}). \tag{39}$$

Therefore, Equation (30) is rewritten into

$$\mu_\star = m(\mathbf{l}^*) + k(\mathbf{l}^*, \mathbf{L})^T \left[\mathbf{K} + \sigma_n^2 \mathbf{I} + \mathrm{diag}\{\Delta_f \Sigma_l \Delta_f^T\}\right]^{-1} \big(\mathbf{v} - m(\mathbf{L})\big),$$

where Δ_f is an N-by-2 matrix with the derivative of N function values $\partial \boldsymbol{f}$'s, and $\mathrm{diag}\{\cdot\}$ denotes the diagonal matrix. Similarly, we rewrite the RSS variance in Equation (31) as

$$\sigma_\star^2 = k(\mathbf{l}^*, \mathbf{l}^*) - k(\mathbf{l}^*, \mathbf{L})^T \left[\mathbf{K} + \sigma_n^2 \mathbf{I} + \mathrm{diag}\{\Delta_f \Sigma_x \Delta_f^T\}\right]^{-1} k(\mathbf{l}^*, \mathbf{L}).$$

3) *Calculation of Mean Function and Variance*: During the fingerprint update, LAAFU processes RSSs for each altered AP i independently. Users conduct location query at different locations of the site. Given discovered altered APs, we calculate the mean function $\mu_\star$ and variance $\sigma_\star$ at the fingerprint point $\star$ where this AP is detected.

Each input location $\mathbf{l}_j$ in input matrix $\mathbf{L}$ corresponds to an estimated location $\hat{\mathbf{l}}_j$, where $1 \leq j \leq N$. And each v_j in $\mathbf{v}$ is the RSS at estimated location $\mathbf{l}_j$ from altered the AP. Let $\mathbf{l}_{ap}$ be the locations of the corresponding AP. For ease of prototyping, we adopt log-distance path loss model [28] in calculating mean $m(\mathbf{l}^\star)$ at a location $\star$ to be predicted

$$m(\mathbf{l}^\star) = \alpha + \beta \log_{10}\left(\frac{\parallel \mathbf{l}^\star - \mathbf{l}_{ap} \parallel}{d_0}\right), \tag{40}$$

where α is the received power (dBm) at reference distance $d_0 = 1$ m, β is the path loss exponent. By default, LAAFU discards the input locations if its RSS value $v_{j'}$ is zero. The covariance among input locations is defined as

$$k(\mathbf{l}_i, \mathbf{l}_j) = \sigma_f^2 \exp\left(-\frac{1}{2d^2}(\mathbf{l}_i - \mathbf{l}_j)^T (\mathbf{l}_i - \mathbf{l}_j)\right), \tag{41}$$

where d represents the length scale and σ_f^2 is the RSS variance. Equation (41) represents the sensitivity of signal change between two different locations.

6.2 GP Hyperparameter Estimation

Note that the parameters $(\alpha, \beta, \mathbf{l}_{ap}, \sigma_n, \sigma_f, d)$ need to be decided before the GP can be applied. In the following, we present how we estimate the parameters.

1) *Calculation of* $\langle \alpha, \beta, \mathbf{l}_{ap} \rangle$*:* We first regress ($\alpha$, β and $\mathbf{l}_{AP}$) in the mean function $m(\mathbf{l})$ since it captures the overall characteristics of signals in site.
Given the target measured RSS values, the regression is to minimize the total RSS error, defined as the sum of the squared difference between mean function values and input target RSSs, i.e.,

$$E = \sum_{i}^{N} \big(m(\mathbf{l}_i) - v_i\big)^2, \tag{42}$$

which can be solved using an efficient gradient-descent algorithm like Limited-memory BFGS (L-BFGS). Specifically, LAAFU firstly calculates the partial derivatives of the parameters given by

$$\frac{\partial E}{\partial \theta_j} = 2 \sum_{i}^{N} \big(m(\mathbf{l}_i) - v_i\big) \frac{\partial m(\mathbf{l}_i)}{\partial \theta_j}, \tag{43}$$

where $\boldsymbol{\theta} = \langle \alpha, \beta, \mathbf{l}_{AP} \rangle$ and the subsequent partial derivatives as

$$\begin{aligned} \frac{\partial m}{\partial \alpha} = 1, \frac{\partial m}{\partial \beta} = \log_{10}(\parallel \mathbf{l}_i - \mathbf{l}_{AP} \parallel), \\ \frac{\partial m}{\partial \mathbf{l}_{AP}} = \frac{\beta(\mathbf{l}_{AP} - \mathbf{l}_i)}{(\mathbf{l}_{AP} - \mathbf{l}_i)^T (\mathbf{l}_{AP} - \mathbf{l}_i)}. \end{aligned} \tag{44}$$

Then, L-BFGS algorithm takes the objective function and partial derivatives as input and returns the parameter results after computation.

2) *Calculation of* $\langle \sigma_n, \sigma_f, d \rangle$*:* Given the mean function, we then formulate the log likelihood of $\mathbf{v}$ [1] as

$$\log p(\mathbf{v}|\mathbf{L}, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{z}^T \mathbf{K}_v^{-1} \mathbf{z} - \frac{1}{2} \log |\mathbf{K_v}| - \frac{n}{2} \log 2\pi, \tag{45}$$

where $\boldsymbol{\theta} = \langle \sigma_n, \sigma_f, d \rangle$ are the hyperparameters to be estimated, and the covariance function between signals is

$$\mathbf{K}_v = \mathbf{K} + \sigma_n^2 \mathbf{I}, \tag{46}$$

and the difference between measured RSSs and mean function

$$\mathbf{z} = \mathbf{v} - m(\mathbf{L}). \tag{47}$$

L-BFGS algorithm is used to solve the optimization problem with the partial derivatives of the log likelihood [10] as

$$\begin{aligned} &\frac{\partial}{\partial \theta_j} \log p(\mathbf{v}|\mathbf{L}, \boldsymbol{\theta}) = \frac{1}{2} \mathbf{z}^T {\mathbf{K}_v^{-1}}^T \frac{\partial \mathbf{K}_v}{\partial \theta_j} \mathbf{K}_v^{-1} \mathbf{z} \\ &- \frac{1}{2} \operatorname{tr}\left(K_{\mathbf{y}}^{-1} \frac{\partial \mathbf{K_v}}{\partial \theta_j}\right) \\ &= \frac{1}{2} \operatorname{tr}\left[\left((\mathbf{K}_v^{-1}\mathbf{z})(\mathbf{K}_v^{-1}\mathbf{z})^T - \mathbf{K}_v^{-1}\right) \frac{\partial \mathbf{K}_v}{\partial \theta_j}\right]. \end{aligned} \tag{48}$$

And LAAFU computes the partial derivative for each hyperparameter as

$$\begin{aligned} \frac{\partial \mathbf{K}_v[i,j]}{\partial \sigma_n} = 2\sigma_n \delta_{ij}, \frac{\partial \mathbf{K}_v[i,j]}{\partial \sigma_f} = 2\sigma_f \exp\left(-\frac{t_{ij}}{2l^2}\right), \\ \frac{\partial \mathbf{K}_v[i,j]}{\partial d} = \frac{t_{ij}\sigma_f^2}{d^3} \exp\left(-\frac{t_{ij}}{2d^2}\right), \end{aligned} \tag{49}$$

where $\delta_{ij} = 1$ if $i = j$, otherwise 0 and $t_{ij} = (\mathbf{l}_i - \mathbf{l}_j)^T (\mathbf{l}_i - \mathbf{l}_j)$.

3) *Hyperparameter Estimation with Location Errors:* Note that Equation (40) contains Δ_f, which represents the derivative of f with respect to locations. Its existence makes the direct solution of $\langle \sigma_n, \sigma_f, d \rangle$ difficult.
To estimate the hyperparameters, the we implement two step iteration [27]. Firstly, LAAFU estimates the hyperparameters of a standard Gaussian process model without the input noise using the training data $\mathbf{L}$ and $\mathbf{v}$ using Equations (30) and (31). Secondly, it computes Δ_f at each of the input locations as

$$\boldsymbol{\partial}_f = \frac{\partial m}{\partial \mathbf{l}} + \frac{\partial k(\mathbf{l}, \mathbf{L})^T}{\partial \mathbf{l}} \mathbf{K}^{-1} \big(\mathbf{v} - m(\mathbf{L})\big). \tag{50}$$

Then LAAFU updates the covariance matrix

$$\mathbf{K}_v = \mathbf{K} + \sigma_n^2 \mathbf{I} + \operatorname{diag}\{\Delta_f \Sigma_l \Delta_f^T\}, \tag{51}$$

which has input noise variance. And then it estimates all hyperparameters by again maximizing the log likelihood using Equation (48). The partial derivative of σ_{l^j} is therefore given by

$$\frac{\partial \mathbf{K}_v[i,i]}{\partial \sigma_{l^j}} = 2 \left(\Delta_f[i,j]\right)^2 \sigma_{l^j}, \tag{52}$$

while all the other off-diagonal entries in $\mathbf{K}_v$ are zero. LAAFU repeats these two steps until the hyperparameters converge.

6.3 Fingerprint Verification, Update & Complexity Analysis

Recall that in Section 5.2, LAAFU obtains W measured RSS vectors from clients with the corresponding estimated locations and a set of altered APs during localization. Instead of using all W data to train GP model, LAAFU randomly selects N samples from these W data for training, and the remaining ones are used to verify the accuracy of the regression prediction. In this way, LAAFU can repeat for several times and choose the best fit in signal map to prevent overfitting.

Specifically, for each altered AP i, taking the selected N samples as input data, LAAFU estimates the hyperparameters based on Section 6.2. Then via this preliminary model, LAAFU predicts the signal value μ_{it} at location $\mathbf{l}_t$ for each remaining data, which consists of estimated location $\mathbf{l}_t$ and corresponding signal value v_{it} (dBm), where $1 \le t \le W - N$. Then LAAFU compares the predicted value μ_{it} with the ground truth value v_{it} and computes the total RSS error as

$$e_i = \sum_{t}^{W-N} |\mu_{it} - v_{it}|. \tag{53}$$

Repeating this process above for several times, LAAFU chooses the one with the smallest total RSS error.

After the model verification, we update the signals at the RPs as follows. For each altered AP i, a regression model is first generated. Then LAAFU calculates the predictive signal mean μ_{ij} and the uncertainty standard deviation σ_{ij} at each RP j. Note that LAAFU discards the data if the predictive signal mean μ_{ij} is smaller than the minimum RSS values in Wi-Fi measurement. In addition to the fingerprint value v_{ij} from AP i at RP j in database, we also have the common average standard deviation σ_i of the measurement noise for AP i using the site survey data.

If the absolute difference $|\mu_{ij} - v_{ij}|$ between the two RSS values μ_{ij} and v_{ij} is larger than the product of a factor κ and the constructed standard deviation σ'_{ij}, where

$$\sigma'_{ij} = \sqrt{\sigma_{ij}^2 + \sigma_i^2}, \tag{54}$$

LAAFU concludes that there is a significant signal change of the AP i at RP j, rather than the signal fluctuation. Considering the different timestamps, autoregressive-moving-average (ARMA) model is applied with weight value $\lambda (0 \leq \lambda \leq 1)$ to conduct the fingerprint update. Thus, the new signal value becomes v'_{ij} (dBm) based on

$$v'_{ij} = (1 - \lambda) \cdot v_{ij} + \lambda \cdot \mu_{ij}. \tag{55}$$

Note that device dependency in RSS measurements is outside the scope of this paper. Interested readers may refer to works like [25], [29] for further details.

We briefly analyze fingerprint update complexity as follows. Given N samples for model training, the computation, including objective function and the partial derivatives, is $O(N)$ in estimating the mean function. As for GP regression, in each iteration it involves the inversion of the covariance matrix, which takes $O(N^3)$. Therefore, the model training sums up to $O(N^3)$.

Given the GP model, it takes $O(N)$ time to predict an RSS and update the fingerprint at one RP, which takes totally $O(RN)$ at all RPs for each altered AP. Note that the fingerprint update is conducted in a separate server and the localization performance will not be affected.

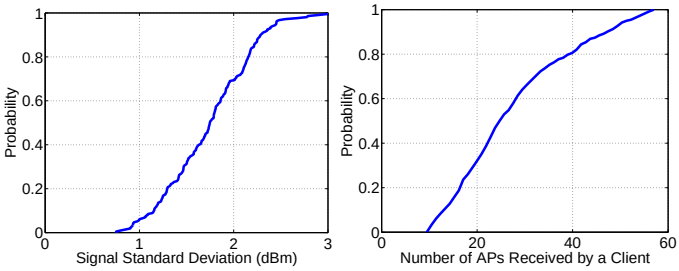
7 Experimental Evaluation

We present as follows the experimental results at our HKUST campus, the Hong Kong International Airport (HKIA) and the Hong Kong Olympia City (HKOC, a leading shopping mall in Hong Kong).

7.1 Experimental Setup & Metrics

We show in Figure 6(a) our experimental site in HKUST, which is between the center for engineering education innovation (E^2I) and the lecture theatre K (LTK) on the second floor of HKUST academic building ($5,400$ m^2). In a three-meter interval, 210 RPs are produced, represented by black dot points shown on the floor plan, in the corridors and open indoor spatial areas.

At each RP of HKUST, we collected 60 fingerprints using smartphone Lenovo A680 and recorded their average values in the database. We calculate the average standard deviation of measurement noise for each AP at all RPs. The cumulative distribution function (CDF) of the average standard deviation in RSS for each AP is shown in Figure 7(a). In addition to the fingerprint database, we collect the client data for location query from 900 random locations over the whole site. Totally, 156 APs are measured in the environment after filtering the APs with only a few data and the mobile APs set by smartphones. Figure 7(b) shows the CDF of the number of distinctive APs that a client can detect at a location (average number at each client is 27).



(a) CDF of the measurement noise for one AP.

(b) CDF of received AP number at a target.

Fig. 7: Signal information in the survey site.

The default parameters in our system are set as follow: $k = 5$ in WKNN algorithm, which is determined empirically. Distance threshold $\gamma = 5$ m in fast detection phase. To demonstrate the effect of the altered APs, we randomly select several APs and modify the transmission power. Signal change factor is selected as 15 dB and the number of altered APs is 2. $M = 60$ subset vector samples are generated each time. We choose $Q = 5$ nearest RPs calculating the cluster similarity while the bandwidth $b = 10$. The update interval is $W = 200$ localization queries. We select randomly $N = 100$ data for regression training, factor $\kappa = 2$ for signal change decision, and $\lambda = 0.5$ as the update weight in fingerprint database update.

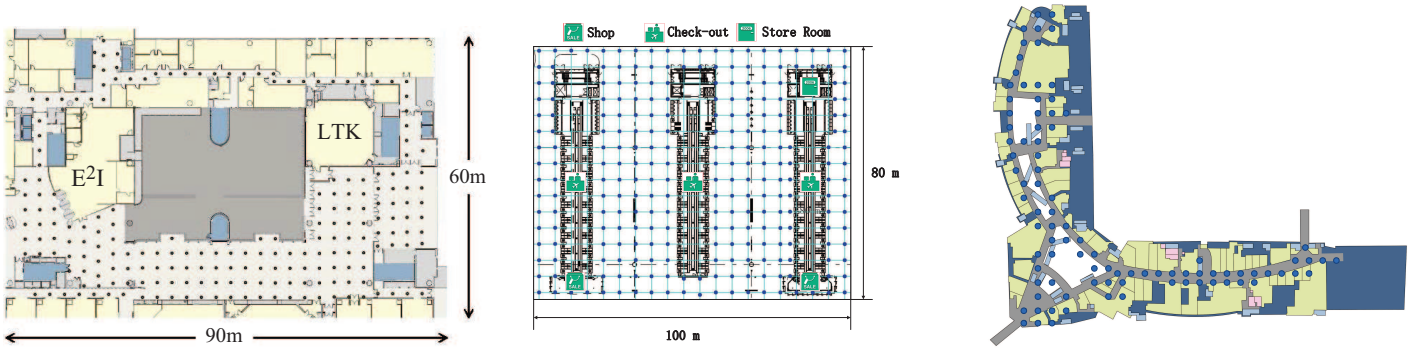
We have also conducted extensive studies in HKIA airport and HKOC shopping mall. Figures 6(b) and 6(c) show the floor plan of HKIA ($8,000$ m^2) and HKOC ($25,000$ m^2), respectively. At the airport and the mall, we survey on 340 and 376 RPs, respectively. The grid sizes during site survey in HKIA and HKOC are both 5 m. The survey, query process and baseline parameters are the same as those in HKUST trials.

In the following, we present the metrics and comparison schemes used in performance evaluation of LAAFU. We define PC as the number of positive testing cases that altered APs actually exist, and NC as the number of negative ones that actually no altered AP exists. Then the true positive rate (TPR) represents the portion of positives which are correctly classified as positive (altered), i.e.,

$$TPR = \frac{TP}{TP + FN}, \tag{56}$$

where TP is the number of correctly classified cases that there are altered APs, and FN is the number of incorrect decision that alteration actually exists. Similarly, true negative rate (TNR) measures the proportion of negatives which are correctly identified as unaltered, i.e.,

$$TNR = \frac{TN}{TN + FP}, \tag{57}$$



(a) Floor plan of a hall in HKUST. (b) Floor plan of check-in area of HKIA. (c) Floor plan of an HKOC mall.

Fig. 6: Floor map in HKUST campus ($5,400$ m^2), HKIA airport ($8,000$ m^2) and HKOC shopping mall ($25,000$ m^2).

where TN is the number of negative cases which are correctly predicted as negative, and FP is the number of positive cases which are incorrectly classified. Finally, the accuracy of fast detection is given by

$$ACC = \frac{TP + TN}{PC + NC}. \tag{58}$$

For localization performance comparison, we measure the error by Euclidean distance between the estimated location and the client's true location. We also compare LAAFU with the following two well-known fingerprint-based localization algorithms:

A. WKNN [7], which is given in Section 4.2. In other words, LAAFU can be integrated within any existing localization algorithm.
B. *Bayesian* method [8], which searches for the client's location by maximizing the likelihood estimator at RP j. The estimation is made by using the weighted average of the RPs with the highest probability.

Besides localization error, we also evaluate our regression model adopted in LAAFU comparing with the log-distance path loss (LDPL) model [28]. As for the signal update accuracy, we calculate the *absolute difference* between predicted RSS value and ground truth value at every RP as the error metric in signal regression process. Note that when using traditional LDPL regression, LAAFU does not have the knowledge of signal variance (Equation (54)) due to simplicity of LDPL. Therefore, in comparing these two models, we focus on the prediction error in RSS signals.

In evaluating the performance of signal update decision, we also use TPR, TNP and ACC. Under such a scenario, we let TP (TN) be the number of correct classifications that RSSs from AP i are actually altered (unaltered) at RP j and FP (FN) be the number of mis-classifications that signal value from AP i are actually unaltered (altered) at RP j.

Finally, in evaluating the update process, we use the *average mutual Euclidean distance* between the locations estimated from RSS subsets. It reflects the gap between the fingerprints and the measured RSS vector since greater fingerprint inconsistency produces larger average mutual Euclidean distance. If this distance is smaller than the predefined threshold γ in fast detection, we conclude that the fingerprint database has been successfully updated to the current environment.

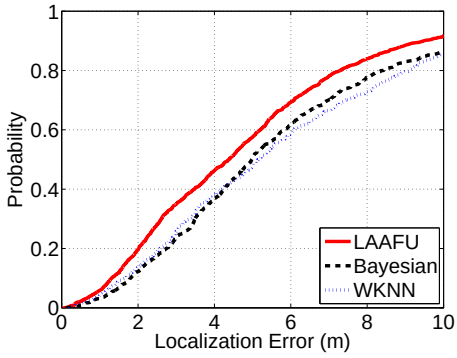
7.2 Illustrative Results in HKUST

For fast detection, we randomly select two APs and generate test client data from the collected ones using different signal change factors to form the positive cases (with altered APs). Such random selection is repeated twenty times. Meanwhile, we have the originally collected clients as negative cases (unaltered). Figure 8(a) shows TNR, ACC and TPR versus the distance threshold γ. When γ is small, it is sensitive that almost all the clients are classified as positive. TPR is low and the unaltered cases are also classified as altered. However, the cases that are identified as negative are basically all true negative, leading to a high TNR. In contrast, when γ is large, most of the clients are predicted as negative, leading to low TNR and high TPR. TNR, ACC and TPR are the highest when $\gamma = 5$.

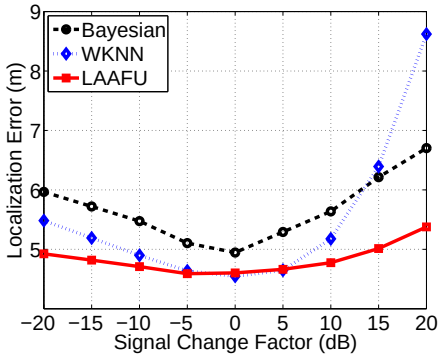
With $\gamma = 5$ m, we show in Figure 8(b) the average mutual distance versus the user arrivals, which corresponds to the optimal point in Figure 9. Obviously, before the introduction of altered APs, the average mutual distance is small, leading to correct classification as negative cases (unaltered). After introduction of altered APs, the mutual distance rises sharply since the altered APs lead to dispersed estimated locations. By detecting such a change, LAFUU can effectively detect the presence of the altered APs via fast detection.

Using the baseline parameter, we adjust the power of two APs with signal change factor 15 dB to evaluate the effect of altered APs. Figure 9 shows the location error versus the user arrivals, which forms the time series of location queries. Before introduction of altered APs at index 40, performances of all three algorithms are similar with small localization errors. Given altered APs, the two traditional algorithms have high localization errors because the altered APs lead incorrect RP matching, which corresponds to the observation in Figure 2. Different from these schemes, LAAFU filters the altered APs from the measured RSS vectors and maintains the higher localization accuracy.

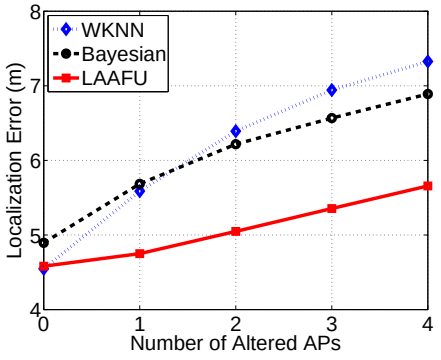
We study the localization error of all algorithm in Figure 10(a). We randomly select two APs with signal change factor 15 dB to evaluate their influence. Figure 10(a) shows the CDFs of localization errors of three different localization schemes in the presence of these altered APs. We can see that both WKNN and Bayesian perform with higher errors compared with LAAFU as these schemes fail to consider



(a) Probability versus localization error in the presence of altered APs.

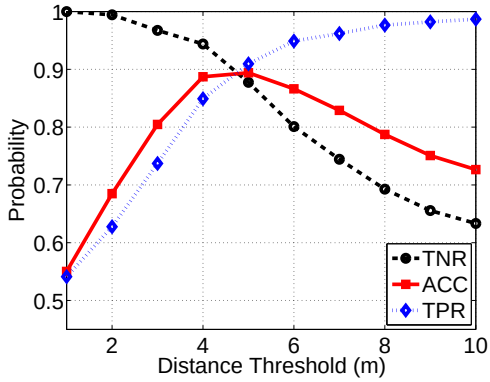


(b) Localization error versus signal change factor.

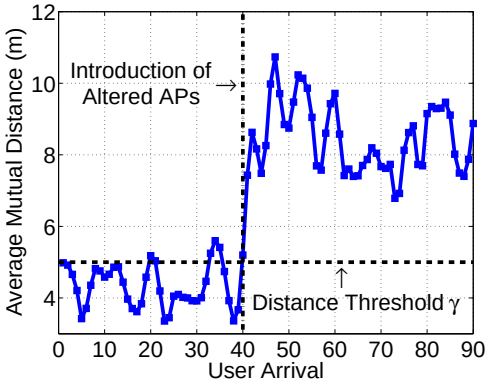


(c) Localization error versus number of altered APs.

Fig. 10: Comparison of different localization algorithms in localization error.



(a) TNR, ACC and TPR versus distance threshold.



(b) Average mutual distance before/after AP alteration.

Fig. 8: Performance of fast detection in LAAFU.

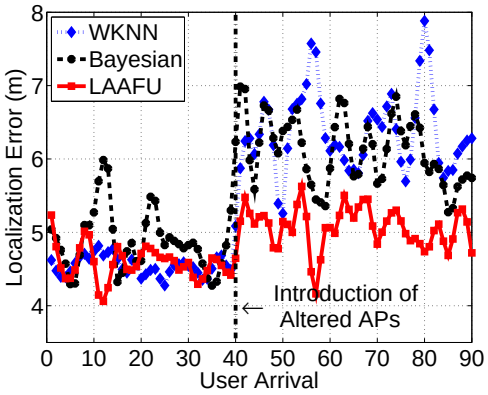


Fig. 9: Mean localization error before and after the introduction of altered APs.

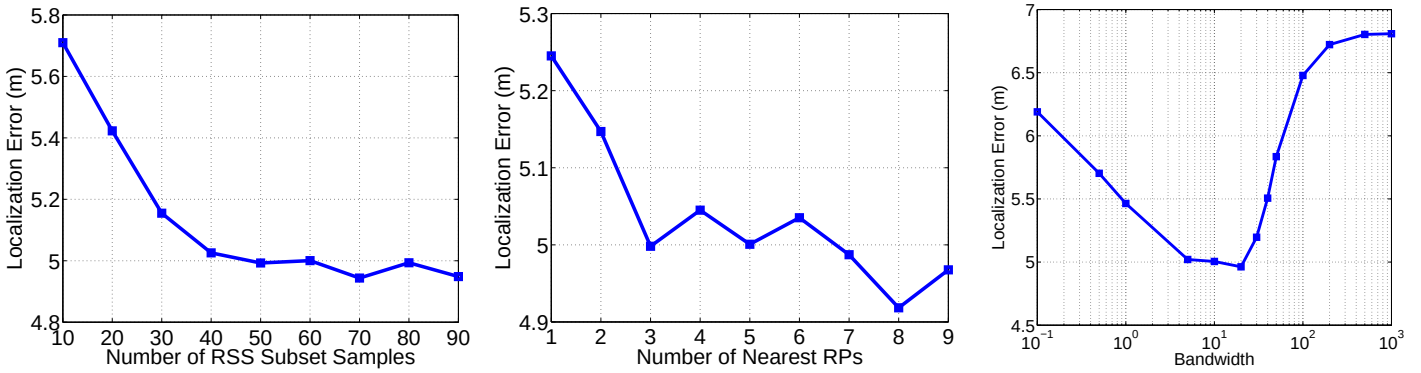
filtering the altered APs.

Figure 10(b) shows the mean localization errors versus the signal change factor between -20 dB and 20 dB. When AP signals are not altered, LAAFU has the same localization error as WKNN. LAAFU classifies the APs as unaltered and therefore runs the same WKNN localization algorithm. Given altered APs, WKNN and Bayesian methods suffers from the dispersion of location estimations. When change factor moves towards -20 dB, the increase of location error tends to converge. It is mainly because the coverage of altered APs decreases with the signal power and therefore the affected locations decrease (leading to smaller number of query data). Under all scenarios with AP alteration, LAAFU achieves better performance with AP filtering.

We show the mean localization error against the number of altered APs in Figure 10(c). Clearly, localization errors of both WKNN and Bayesian methods rise significantly with the increasing number of altered APs. In contrast, LAAFU shows more robustness against altered APs, as it successfully classifies the alteration and localizes the users with remaining unaltered APs.

Figure 11(a) shows the localization error versus the number of generated subset samples. We can observe that the more RSS subsets LAAFU generates, the higher localization accuracy it achieves. It is mainly because increase of subsets leads to more location estimations, which provide more information in distinguishing a dense cluster. After a certain number of subsets, the improvement becomes saturated as the existing samples are sufficient for clustering and classification.

Figure 11(b) shows the localization error versus the number of nearest RPs (Q in Equation (19)) in around the centroid for cluster similarity computation. When Q is small, the localization error is high as there exists matching error in finding few nearest RPs in signal space. When Q further increases, the error decreases because considering more nearest neighbors (RPs) reduces the effect of the random signal fluctuation. We select $Q = 5$ in our baseline since the improvement begins to saturated when approaching this value.



(a) Localization error versus number of RSS subset samples generated.

(b) Localization error versus number of nearest RPs set in cluster similarity.

(c) Localization error versus bandwidth set in penalty term.

Fig. 11: Localization performance of LAAFU with different parameters.

Figure 11(c) shows the mean localization error versus the bandwidth b in penalty term (Equation (20)) in a logarithmic scale, as we implement a Gaussian kernel. According to Equation (20), when b is very small, the penalty term ν_c is close to zero, showing a more uniform weight assignment with little differentiation. Solely using average cluster similarity ϱ_c therefore cannot distinguish those clusters with small size. When b increases, ν_c differs more sharply at different cluster sizes, which helps differentiate the clusters. However, when b further increases, the performance decreases as ν_c becomes too sensitive towards cluster size and dampens the effect from using ϱ_c.

In the following, we evaluate the effect of RSS update using LAAFU. We calculate the RSS errors between the predicted RSS and ground truth at every RP for each AP using GP and the LDPL regression model (LDPL) in LAAFU. Figure 12(a) shows the CDFs of RSS errors in signal regression. We can observe that GP outperforms the LDPL in regressing the signal values, as GP considers capturing the local signal patterns while preserving the overall signal propagation.

Figure 12(b) shows TNR, ACC and TPR versus the number factor κ used in signal update decision at RPs. It shows that in general ACC increases first and then decreases, while TNR (TPR) generally decreases (increases). When κ is small, most of the data that are identified as negative are true negative, which leads to high TNR. TPR is small as the update decision is too sensitive to the temporal signal fluctuation. As κ increases, FN decreases while both TPR and ACC increase. As factor κ further increases, ACC starts decreasing because LAAFU may also classify positive cases as negative, leading to higher FP and lower TNR. Therefore, we choose $\kappa = 2$, where ACC, TPR and TNR are all around 88%.

We randomly pick out two APs with signal change factor 15 dB. Given location queries, LAAFU conducts fast detection and subset localization with altered APs. Then LAAFU uses two different signal regression models, GP and LDPL, in fingerprint database update phase. After that we run the WKNN localization algorithm over the following fingerprints: (1) the ground truth database with known altered RSS values (ground truth); (2) the updated database using GP (LAAFU); (3) the updated database using LDPL (LDPL); (4) outdated fingerprint database (original).

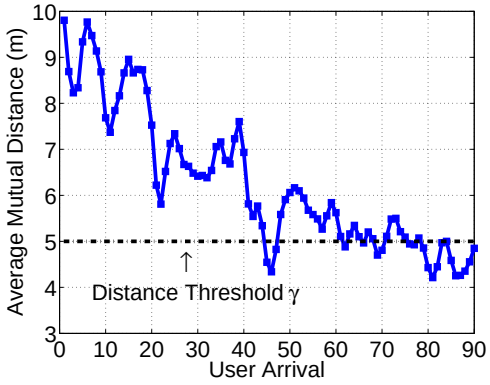
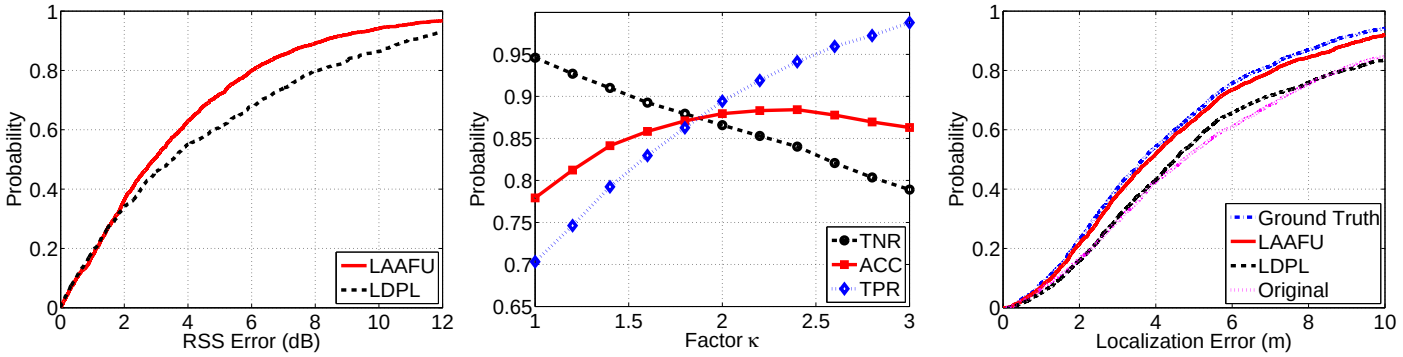


Fig. 13: Average mutual distance versus user arrival (location queries).

Figure 12(c) shows the CDFs of localization error using above fingerprint databases. The average error is 4.51 m, 4.72 m, 7.10 m and 7.28 m when using the ground truth database, the one updated by LAAFU, the one updated using LDPL model and the outdated one, respectively. It shows that using LAAFU, we can obtain very close localization error with the ground-truth database, meaning an effective signal update. Compared with LDPL, GP in LAAFU captures the signal change and the local signal patterns more accurately, leading to lower location errors.

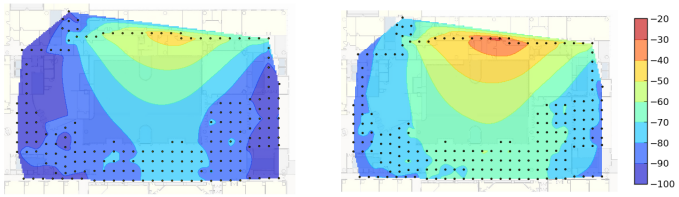
Figure 13 presents the average mutual distance in fast detection phase versus user arrivals, which shows the fingerprint database update result. The update process occurs four times at index 20, 40, 60 and 80, respectively. With the update weight $\lambda = 0.5$, the fingerprint is updated towards the ground truth and therefore the mutual distance decreases. We can observe that LAAFU successfully updates the fingerprint database with the subset sampling and altered AP classification.

To demonstrate the signal update process more clearly, we show the corresponding heat map changes for one of the two altered APs (MAC address is 04-4F-AA-4C-43-18). Figure 14(a) shows the heat map of its original fingerprints



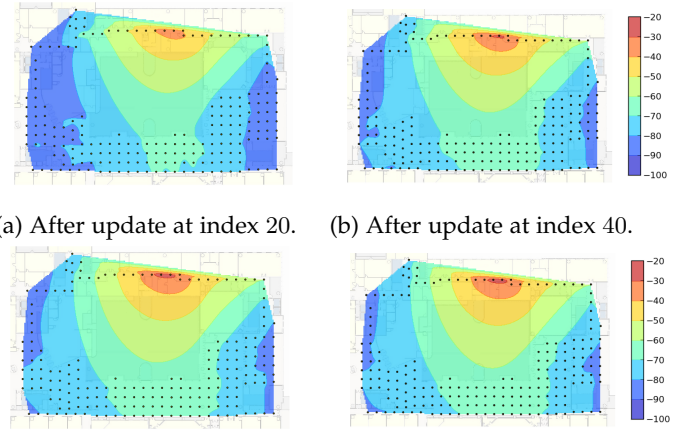
(a) CDF of RSS regression error. (b) Impact of factor κ in update decision. (c) Location error CDF using updated signals.

Fig. 12: Performance in fingerprint database update of LAAFU.



(a) Original heat map (outdated after AP alteration). (b) Ground truth heat map after power alteration.

Fig. 14: Overall heat map change of AP 04-4F-AA-4C-43-18.



(a) After update at index 20. (b) After update at index 40.

(c) After update at index 60. (d) After update at index 80.

Fig. 15: Temporal heat map update of AP 04-4F-AA-4C-43-18 using LAAFU.

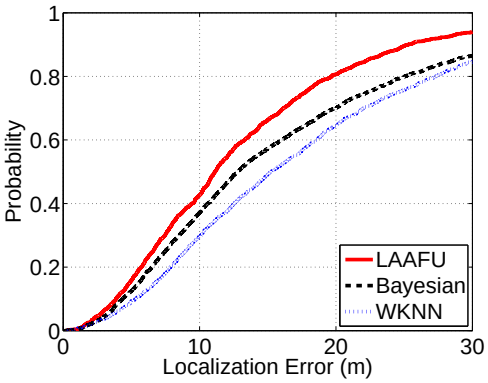


Fig. 16: CDF of localization errors (HKIA).

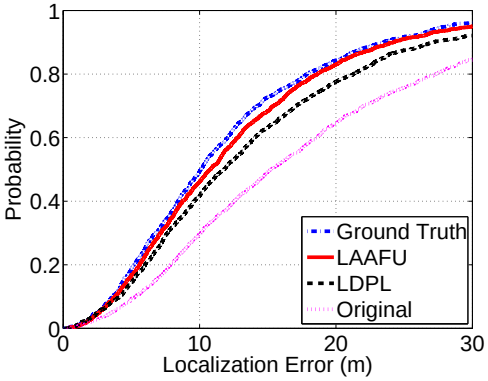


Fig. 17: CDF of errors using updated fingerprints (HKIA).

in the database. Figure 14(b) shows the ground truth signal distribution after power adjustment. We then show in Figure 15 the temporal update of signal map for the same AP with respect to index of user arrivals. In the update process at index 80, the Gaussian process regression model adapts $\sigma_n = 2.79$, $\sigma_f = 5.74$, $l = 23.08$, $\sigma_{x^1} = 3.4$ and $\sigma_{x^2} = 3.8$ as hyperparameter values, which are all learned via hyperparameter estimation.

7.3 Illustrative Results in HKIA & HKOC

We have also conducted similar studies within the site of HKIA and HKOC. In Figures 16 and 17, we show the localization error of LAAFU and the location accuracy using the updated fingerprints in HKIA. Similar results have also been shown in Figures 18 and 19 for HKOC trials. Note that the marked resemblance between them and those in HKUST (Figures 11(a) and 12(c)). As the results are qualitatively similar, we do not repeat the others here. Interested readers may refer to [30] for further details.

8 Conclusion

When AP signals are altered (through, for examples, AP movement, power adjustment, etc.), conventional fingerprint-based indoor localization schemes can no longer achieve high accuracy. In this paper, we have proposed

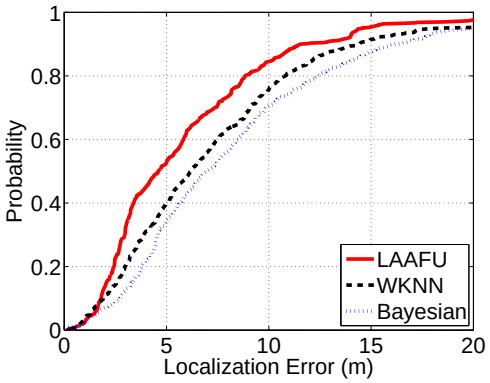


Fig. 18: CDF of localization errors (HKOC).

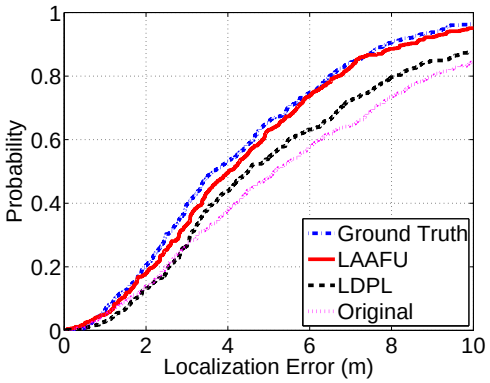


Fig. 19: CDF of errors using updated fingerprints (HKOC).

and studied LAAFU (Localization with Altered APs and Fingerprint Updating), which achieves accurate indoor localization and automatic fingerprint database update with possibly altered APs. Using novel subset sampling, LAAFU is able to efficiently detect the altered APs, filters them out in the measured RSS vector and hence finds the location of client. Given the RSS vectors received and the locations estimated, LAAFU transparently adapts the fingerprint database to the signal changes by applying the non-parametric Gaussian process regression method (i.e., implicit crowdsourcing). We have conducted extensive experiments on LAAFU at our campus. LAAFU is shown to be robust against altered APs to achieve high localization accuracy. It can also dynamically and automatically update its fingerprint database without the need of time-consuming offline site survey.

REFERENCES

[1] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
[2] C. Feng, W. Au, S. Valaee, and Z. Tan, "Received-signal-strength-based indoor positioning using compressive sensing," *IEEE Trans. Mobile Computing*, vol. 11, no. 12, pp. 1983–1993, Dec 2012.
[3] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. Ni, "Csi-based indoor localization," *IEEE Trans. Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1300–1309, July 2013.
[4] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, Nov 2007.
[5] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*, vol. 2, 2000, pp. 775–784.
[6] D. Han, S. Jung, M. Lee, and G. Yoon, "Building a practical Wi-Fi-based indoor navigation system," *IEEE Pervasive Computing*, vol. 13, no. 2, pp. 72–79, 2014.
[7] S. Han, C. Zhao, W. Meng, and C. Li, "Cosine similarity based fingerprinting algorithm in WLAN indoor positioning against device diversity," in *Proc. IEEE ICC*, 2015, pp. 4313–4317.
[8] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. ACM MobiSys*, 2005, pp. 205–218.
[9] P. Mirowski, D. Milioris, P. Whiting, and T. Kam Ho, "Probabilistic radio-frequency fingerprinting and localization on the run," *Bell Labs Technical Journal*, vol. 18, no. 4, pp. 111–133, 2014.
[10] B. Ferris, D. Hähnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *Proc. Robotics: Science and Systems*, 2006.
[11] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal, "ARIADNE: A dynamic indoor signal map construction and localization system," in *Proc. ACM MobiSys*, 2006, pp. 151–164.
[12] C. Wu, Z. Yang, Y. Liu, and W. Xi, "WILL: Wireless indoor localization without site survey," *IEEE Trans. Parallel and Distributed Systems*, vol. 24, no. 4, pp. 839–848, 2013.
[13] S. Yang, P. Dessai, M. Verma, and M. Gerla, "FreeLoc: Calibration-free crowdsourced indoor localization," in *Proc. IEEE INFOCOM*, 2013, pp. 2481–2489.
[14] J.-g. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie, "Growing an organic indoor location system," in *Proc. ACM MobiSys*, 2010, pp. 271–284.
[15] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Proc. ACM MobiCom*, 2012, pp. 269–280.
[16] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. ACM MobiSys*, 2012, pp. 197–210.
[17] M. Atia, A. Noureldin, and M. Korenberg, "Dynamic online-calibrated radio maps for indoor positioning in wireless local area networks," *IEEE Trans. Mobile Computing*, vol. 12, no. 9, pp. 1774–1787, Sept 2013.
[18] J. Yin, Q. Yang, and L. M. Ni, "Learning adaptive temporal radio maps for signal-strength-based location estimation," *IEEE Trans. Mobile Computing*, vol. 7, no. 7, pp. 869–883, 2008.
[19] Z. Sun, Y. Chen, J. Qi, and J. Liu, "Adaptive localization through transfer learning in indoor Wi-Fi environment," in *Proc. IEEE ICMLA*, 2008, pp. 331–336.
[20] V. W. Zheng, E. W. Xiang, Q. Yang, and D. Shen, "Transferring localization models over time," in *Proc. AAAI*, vol. 3, 2008, pp. 1421–1426.
[21] A. Schwaighofer, M. Grigoras, V. Tresp, and C. Hoffmann, "GPPS: A Gaussian process positioning system for cellular networks," in *Proc. NIPS*, 2003.
[22] B. Ferris, D. Fox, and N. D. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," in *Proc. IJCAI*, vol. 7, 2007, pp. 2480–2485.
[23] A. Brooks, A. Makarenko, and B. Upcroft, "Gaussian process models for indoor and outdoor sensor-centric robot localization," *IEEE Trans. Robotics*, vol. 24, no. 6, pp. 1341–1351, Dec 2008.
[24] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.
[25] L. Li, G. Shen, C. Zhao, T. Moscibroda, J.-H. Lin, and F. Zhao, "Experiencing and handling the diversity in data density and environmental locality in an indoor positioning service," in *Proc. ACM MobiCom*, 2014, pp. 459–470.
[26] G. F. Jenks, "The data model concept in statistical mapping," *International Yearbook of Cartography*, vol. 7, no. 1, pp. 186–190, 1967.
[27] A. McHutchon and C. E. Rasmussen, "Gaussian process training with input noise," in *Proc. NIPS*, 2011, pp. 1341–1349.
[28] H. Shin, Y. Chon, Y. Kim, and H. Cha, "MRI: Model-based radio interpolation for indoor war-walking," *IEEE Trans. Mobile Computing*, vol. 14, no. 6, pp. 1231–1244, June 2015.
[29] C. Xiang, P. Yang, C. Tian, H. Cai, and Y. Liu, "Calibrate without calibrating: An iterative approach in participatory sensing network," *IEEE Trans. Parallel and Distributed Systems*, vol. 26, no. 2, pp. 351–361, Feb 2015.
[30] W. Lin, "Indoor localization and fingerprint update with altered access points," Master's thesis, Department of CSE, The Hong Kong University of Science and Technology, 2015.