Towards Area Classification for Large-scale Fingerprint-based System

Suining He Jiajie Tan S.-H. Gary Chan
Department of Computer Science and Engineering,
The Hong Kong University of Science and Technology, Hong Kong, China
{sheaa, jtanad, gchan}@cse.ust.hk

ABSTRACT

In spacious and multi-area buildings, fingerprint-based localization often suffers from expensive location search. Besides, context knowledge like inside/outside-region and floor area is important for complete location service. To address above issues, beyond the algorithms finding the exact location point, we study accurate and efficient indoor area classification for large-scale fingerprint-based system. We first study leveraging the one-class classification to conduct inside/outside-region detection given only the inside fingerprints. Then we discuss different area determination algorithms, and compare their detection accuracy and deployment efficiency. To further enhance accuracy, we also discuss rejecting unclassifiable signals and calibrating heterogeneous devices. We have implemented different algorithms on Android platforms. Experimental trials (totally over 30,000 fingerprints and 15,000 test data) at an international airport, a business building, a premium shopping mall and a university campus have evaluated practicability and deployability of different classification schemes. Our studies can also serve as design guidelines for area classification.

ACM Classification Keywords

C.5.m Computer System Implementation: Miscellaneous

General Terms

Experiment, System

Author Keywords

Fingerprinting; area classification; inside/outside-region detection; context-awareness; experimental comparison.

INTRODUCTION

Among the techniques explored for indoor location-based service (ILBS), fingerprinting localization has become a well-studied approach as it requires no knowledge of access point (or base station) locations and line-of-sight signal measurement [4, 13, 27, 45]. In the offline stage of fingerprinting scheme, the surveyors collect received signal strength indicators (RSSIs) from *access points* (APs) at predefined locations called *reference points* (RPs). The signals and the positions together form the so-called *fingerprints* [15]. Then in the online stage, a target (client or user) collects the vector of RSSIs, and ILBS compares the signals with stored fingerprints. Coordinates of the most similar fingerprints on the map are then returned as the final position (location point) of the target.

Previous fingerprint-based ILBS systems often focus on finding the exact location point of a target through some fine-grained localization algorithms in the 2-D map [4, 17, 53]. However, when we are given a spacious region (say, a shopping mall or an office building), with RF fingerprints collected at multiple areas (say, floors or rooms), the process becomes more complicated. Beyond finding the location point, two critical questions arise: 1) *how to detect whether the ILBS user is inside or outside the fingerprint region (i.e., inside/outside-region detection)?* 2) *how to identify which area (say, floors, rooms or buildings) within the fingerprint region the user is at (i.e., area identification)?* The above concerns can be summarized as the *area classification* problem.

Existing ILBS can benefit from solving the area classification problem in the following aspects:

- *Search scope and computation reduction*: Area classification can reduce the search scope of traditional fingerprint-based localization [4,53]. A target can be first mapped to an area like a floor or a building. Then fine-grained localization schemes like [4, 14, 45, 53] can simply focus on that area. It is essential especially in the spacious indoor sites like the international airports, shopping malls or skyscrapers with large quantities of RPs. If the search scope is not first significantly reduced through inside/outside filtering or area mapping, fine-grained localization cannot be effectively and efficiently conducted. With recent explosion of indoor Big Data, area classification is an important design in building the large-scale indoor localization systems.
- *Context awareness and notification*: Area classification also provides the current context information [12]. For example, inside/outside-region detection, represented by indoor/outdoor detection, is an important part for ILBS. The systems need to know whether the target is outdoor or indoor to seamlessly switch between the Google map and an indoor one. Inside the region of interests, retailers in a shopping mall aim at pushing specific advertisement or coupons to users at different areas. The site monitors of an airport conduct the flow analysis at different areas for instant site management (or so-called "geo-fencing").

UbiComp '16, September 12-16, 2016, Heidelberg, Germany
 ISBN 978-1-4503-4461-6/16/09…$15.00
DOI: http://dx.doi.org/10.1145/2971648.2971689

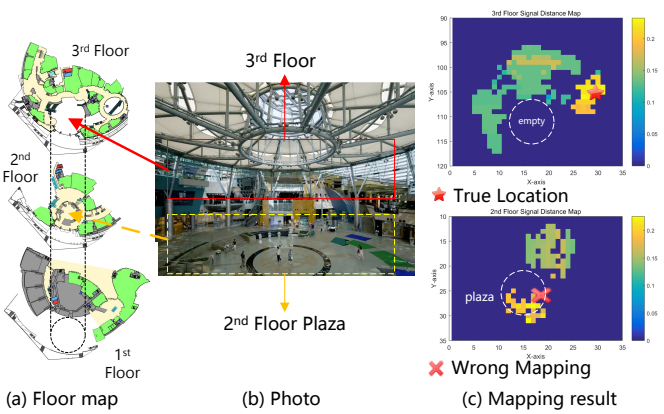

Figure 1: Floor plan (left) and photo (middle) of a business building (Hong Kong Cyberport) and the signal similarity map for a target in two floors (right).

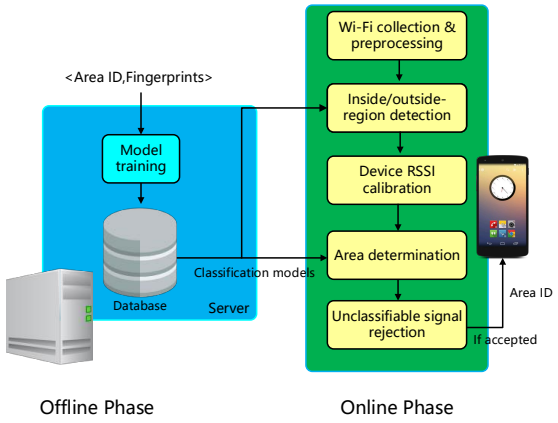

Figure 2: System design of the area classification system. It can serve as a plug-in for existing fingerprint systems.

Despite above benefits, the area-classification in large-scale indoor localization has not yet been thoroughly studied in many works [4, 27, 53]. Different area classification models or algorithms have not been compared in spacious sites like the airports or shopping malls. To fill this gap, we conduct extensive studies in this paper over the practical deployment of area classification. Different from the previous fine-grained localization studies, we need to address the following three unique challenges in deploying area classification:

1) *How to identify whether the target is inside the region given only the RF fingerprints within the region?* Seamless switching between inside/outside regions is important for the complete location-based service, especially for indoor/outdoor detection. As only the inside (say, indoor) fingerprints are provided due to survey cost, classifying an RSSI vector outside the region becomes challenging.

2) *How to efficiently map the target to the area?* Previous traditional fine-grained localization often needs to search over all the reference points [4], which is usually computationally inefficient especially in large-scale fingerprint systems. When we consider an efficient hierarchal way to locate the target position (i.e., first area mapping and then fine-grained localization), the area classification algorithm needs to be more efficient than the traditional fine-grained one such that the influence over the system overhead is minimal.

3) *How to accurately determine the area given noisy signals?* Besides reducing search scope, area classification algorithms should be robust to signal noise and accurate enough to map the target to the correct area. Otherwise, incorrect area mapping can lead to even worse user experience than the error of fine-grained localization. Figure 1 shows an example of floor localization in a business building of Hong Kong Cyberport. The grid color (corresponding to an RP $\mathbf{F}_n$ in site) represents its RSSI Euclidean distance to query $\mathbf{T}$ in signal space [4] (the redder in color, the closer in distance). Due to high similarity between RPs at the 2nd and 3rd floors, if the nearest neighbor matching is applied, the target is mapped from the 3rd floor to 2nd floor, which is unsatisfactory in user experience.

To address above issues, we conduct extensive experimental studies over the area classification schemes. Our contributions in this paper are in three folds as follows:

- *One-class inside/outside-region detection*: We formulate the inside/outside-region detection into a one-class classification or so-called *data description* problem. Through the one-class classification, we classify the outside signals as outliers to those inside ones. Our experiment shows that we do not need extra fingerprints from the outside environment, while achieving high accuracy in the indoor/outdoor switch detection. Such a scheme can be easily applied to other scenarios like "geo-fencing" or intrusion detection, given only the fingerprints in the region of interests.
- *Extensive studies over area classification models*: Beyond the traditional fine-grained location estimation, area classification can serve as a preprocessing step to reduce the search scope first. The area mapping should be efficient and accurate in order to shed minimal influence upon the following fine-grained localization process. We extensively study and compare several machine learning algorithms (models) for large-scale area classification, including area identification (floor) and inside/outside-region detection (indoor/outdoor). We compare their accuracy, training/testing complexity, and adaptivity to different indoor sites, which shed more deployment insight for area classification.
- *Practical unclassifiable signal rejection and device calibration*: To support robust localization under noisy signal environment, we further design rejection schemes for above machine learning algorithms if sampled RF signals may not provide deterministic location results. As the same signal at different devices may have different readings, we also propose a practical online approach which calibrates the device signals based on the RF fingerprints. These studies can also serve as insightful design guidelines to build a practical area classification system for spacious indoor sites.

Figure 2 shows the architecture of area classification which can be integrated with any Wi-Fi fingerprint-based ILBS system [4, 24]. It is basically divided into *offline* phase and *online*

phase. Each area (like a floor or a building), where the ILBS systems are applied, is initially assigned with a unique area ID. During the offline phase, given the fingerprints and their corresponding area IDs, we first train certain classification models and store their model parameters within the database. Then in the online phase, the smartphone of the user measures Wi-Fi signals and then queries for area. The *inside/outside-region detection* first checks whether the user is inside or outside the region of interests (say, indoor or outdoor). If he/she is outdoor, outdoor location service (say, Google map) is provided. If given he/she is indoor, the system then checks whether the device type is different from that in database. Online *device RSSI calibration* is conducted upon the different devices, and the device model parameter is stored into the database for later use. The RSSI vectors are then fed to the *area determination* module to find the floors or rooms. Given decision variables (for example, the probabilities at different areas), the *unclassifiable signal rejection* module discards the collected signals if the decision is highly uncertain. Otherwise, the area ID is returned to the client for fine-grained location estimation.

We have built our LBS system upon Android platforms, and conducted extensive experimental trials (overall $30,960$ Wi-Fi fingerprints and $15,764$ test RSSI vector) in an international airport, a business building, a premium shopping mall and our university campus. Our experiment compares these algorithms in classification accuracy and efficiency, and validates the practical system design. We present that our proposed calibration scheme effectively compensates the device heterogeneity for accurate area localization. Our studies also show that one-class classification algorithms can efficiently provide seamless indoor/outdoor switching, requiring no outdoor fingerprint collection. Though we consider here Wi-Fi RSSIs for ease of ILBS prototyping on smartphones, the algorithms studied can be easily applied to other existing or emerging fingerprints, including Bluetooth and channel state information.

Note that in this paper we generalize floors, outdoor/indoor and building/room into "areas" given RF fingerprints and the area ID. Building/room classification is qualitatively similar to the floor localization. RPs of different buildings or rooms can be first partitioned into multiple groups via manual operation or automatic clustering, and then area classification can be conducted upon these groups. Therefore, we focus on the floor identification and indoor/outdoor detection in our case studies.

The remainder of this paper is organized as follows. After briefly reviewing related works, we first discuss the preliminaries of the area classification with Wi-Fi fingerprints. Then we present the classification algorithms (models) on the inside/outside-region detection and then the area determination. After that, we discuss the unclassifiable signal rejection and effective device calibration to further enhance classification accuracy. We then present the experimental results with the design considerations and finally conclude our paper.

RELATED WORK

We briefly review the related works as follows. Fingerprint-based algorithms for indoor localization have been extensively studied for decades, including RADAR [4], Horus [53], P-CA [10], PiLoc [26], MapCraft [45], Modellet [24] and INTRI [18]. They either utilize some similarity metrics between RP fingerprints and target signals, or consider certain probabilistic modeling over the RSSI. Though promising results have been achieved, few of them have discussed large-scale area classification for the floor, indoor/outdoor or building switching issues. Our paper, however, investigates the area classification problem, and provides the practical insights and system designs when addressing it. Our area classification design is also generic enough as a plug-in for above state-of-the-art systems [16, 24] for accuracy and efficiency improvement.

Some recent works have begun to study area localization [25, 30, 43, 52], ranging from floor classification, indoor/outdoor switching and room (building) classification. Floor localization has attracted attention in recent works [2, 6, 38]. SkyLoc [43] utilizes RSSIs from cell towers to locate the floors on skyscrapers. However, cell signals are likely to be different across various cell service providers, which may not be widely deployed for pervasive fingerprint-based localization. Inertial sensors like accelerometer readings can indicate the transition between different floors [51]. F-Loc [49] leverages crowdsourcing and inertial sensors to improve floor localization accuracy. However, the inertial sensors require the initial location input and constant calibration. Sensor readings may suffer from external user input noise during crowdsourcing. Implementing barometer for floor localization has attracted much attentions recently [31]. Nevertheless, the performance of barometer may degrade under thermal changes [31], and requires initial floor input or overtime compensation [47]. Orthogonal to above works, our studies utilize commercial and pervasive Wi-Fi signals beyond the existing networks. We focus on finding a general classification design, which is more scalable to existing RF-based ILBS systems and different scenarios in practice. Above works are also amendable to our study for more advanced fusion applications.

As a typical application of inside/outside-region detection, indoor/outdoor switching has also been studied recently [34, 35, 37, 56]. Using the signal absence or drop in accuracy of GPS indoors may not work when the smartphone users are surrounded by high buildings and detect few satellites outdoors [44]. Weather (cloudy or rainy) may also affect GPS detection, as the number of satellites detected may also vary. Moreover, energy consumption in GPS on mobiles can be large. The works in [35] and [56] utilize GPS and the fusion of multiple sensors (inertial motion sensors, light detector and magnetometer) to classify the indoor/outdoor environments. In [34], semi-supervised learning is proposed to detect indoor or outdoor, given unlabeled sensor measurements.

Different from these works on sensor fusion, we use Wi-Fi fingerprint signals for more general area classification (like classifying the floors where GPS, light or magnetic field may not be differentiable), and better scalability over different mobile devices (like some simple wearable hardwares). Our novel inside/outside-region detection with one-class classification [41] requires no extra outdoor fingerprints beyond the existing indoor localization systems. Furthermore, our studies are also amendable to the semi-supervised learning [34] to accommodate and leverage the unlabeled fingerprints.

Room/building classification for context awareness has been recently investigated. Represented by Surround-Sense [3], classification using sound [39, 42] and light [46] has attracted much attention. However, calibration or modification over off-the-shelf smartphones for above signals is needed before their practical and large-scale deployment. Further hardware modification may be also needed for better detection accuracy. The work in [25] searches against the entire Wi-Fi fingerprint database of different rooms and buildings, which is computationally expensive for large-scale ILBS deployment. Different from above, our paper focuses on RF fingerprints for general deployment purpose and studies the efficient area classification schemes.

PRELIMINARIES OF FINGERPRINTS

In this section, we briefly introduce the preliminaries of RSSI fingerprints for area classification. Fingerprints used for area classification can be the same as those provided for traditional fingerprinting systems [4]. The survey can be conducted at the predefined regular grids [24]. Each fingerprint is assigned with an area ID corresponding to the area (say, the floor or the building). Note that the area ID is given in the settings of our paper, and can be derived through some simple map partitioning or RF fingerprint clustering.

Let N be the number of reference points (RPs), and L be the number of access points (APs) deployed in the site. Denote the received signal strength indicator (RSSI) from AP l at RP n as f_n^l (dBm). At each predefined RP, we collect multiple fingerprint signals in order to mitigate the random effect. The mean RSSI $\bar{f}_n^l$ at each RP n from AP l is given by

$$\bar{f}_n^l = \frac{1}{S_n^l} \sum_{s=1}^{S_n^l} f_{ns}^l, \tag{1}$$

where f_{ns}^l is the s-th RSSI sample (in dBm) at RP n from AP l, and S_n^l is the number of RSSI samples collected. Then the RSSI vector at RP n is defined as

$$\mathbf{F}_n = \left[\bar{f}_n^1, \bar{f}_n^2, \ldots, \bar{f}_n^L\right]. \tag{2}$$

Meanwhile, let σ_n^l (dB) be the corresponding standard deviation of S_n^l collected RSSIs. By definition, if AP l is not detected at RP n, the signal is stored as 0.

Similarly, given the target measured RSSI t^l from AP l, the RSSI vector at target side, denoted as $\mathbf{T}$, can be defined as

$$\mathbf{T} = \left[t^1, t^2, \ldots, t^L\right]. \tag{3}$$

In order to differentiate the RSSIs in indoor environment, mW, instead of dBm, is used when we consider the signal level, i.e.,

$$\bar{f}_n^l|_{mW} = 10^{\left(\bar{f}_n^l|_{dBm}\right)/10}, \tag{4}$$

which transforms RSSIs from smartphones to values for better signal differentiation. Then we differentiate the strong and weak signal values in the classification. Correspondingly, we also transform RSSI values t^l's in $\mathbf{T}$ from dBm into mW.

Let $\mathbf{C}$ be the set of areas to be classified (i.e., classes in the machine learning), and $|\mathbf{C}|$ be its cardinality (then each area is indexed by $c \in \{1, \ldots, |\mathbf{C}|\}$). For each $\mathbf{F}_n$, we have the corresponding area ID $Y_n \in \mathbf{C}$. Similarly, for each target-measured $\mathbf{T}$, we denote its area ID as $y \in \mathbf{C}$, which is to be identified in our area classification study.

ONE-CLASS INSIDE/OUTSIDE-REGION DETECTION

In this section, we first discuss how to detect whether the user is inside the region of interest. Usually we are not given RF fingerprints outside the region due to large survey cost (for example, in the outdoor environment around the target site). To support ubiquitous localization and efficient inside/outside-region detection, we consider formulating it into *one-class classification* or the so-called *data description* problem. The goal of one-class classification is to distinguish between a set of known objects and all other possible ones [40]. Given a set of known data (say, inside-region fingerprints), we find a data description model over them. If a new object (signal) does resemble the data description, it is likely to belong to the target data (i.e., inside-region signals in our case). Otherwise, it is likely to be an outlier or called a *novelty* [1] (i.e., outside-region signals). Based on above settings, we can formulate the inside/outside-region detection problem into a typical data description problem, given only fingerprints in the inside region [41]. In the experiment, we train the data description models based on all the RSSI fingerprints indoors. Then given the trained models, we classify the incoming RSSI vectors.

In the following, we discuss several approaches for one-class classification in our studies:

- *Support vector data description* (SVDD): The kernel function of support vector machine (SVM) forms a nonlinear hyperplane classifying the data points. Adaptation of SVM can serve as outlier detection based on such a nonlinear hyperplane [36], forming the SVDD. The SVDD hyperplane separates the data points from the origin (in the feature space), and maximizes the distance from the hyperplane to the origin [29]. Then the incoming RSSIs outside the hyperplane are classified as outliers (i.e., outdoor).

- *Self-Organizing Map (SOM)* [21]: SOM uses a self-organizing map to describe the fingerprints. The construction of SOM over the fingerprints is to map them into the feature space which retains their neighboring distances via a neural network [21]. Then the outlier detection is based on reconstruction error, which is the difference between this fingerprint vector and its closest neuron in the trained SOM.

- *Minimax probability machine (MPM)* [11]: MPM finds the linear classifier that separates the data from the origin in the signal space [40]. The linear classifier represents the hyperplane such that with at least a certain probability that the given fingerprints lies within the inside-region, given the mean and covariance matrix. The mean and covariance matrix of the fingerprints are used to determine outliers [11].

- *Principal component analysis (PCA)* [7, 40]: PCA data description maps the target data into a linear subspace constructed from principal components. PCA determines a subspace hyperplane which minimizes the squared projection error. The observations which lie far away from the

projection directions are classified as the outliers (outside-region RSSI vectors). Specifically, PCA data description checks the reconstruction error of the input fingerprint vector, i.e., the difference between the original object and its projection onto the subspace.

While in this paper we focus on indoor/outdoor detection, one-class classification can be easily applied to other region detection such as the site monitoring and geo-fencing.

AREA IDENTIFICATION

In this section we discuss how to identify the target area, given the decision that the user may be in the site of interest. Different from inside/outside-region detection, the existing fingerprint systems often provides available at different areas of the site. We discuss several models: nearest neighbor (NN), signal heuristic classification (SHC), artificial neural network (ANN), deep belief network (DBN) and support vector machine (SVM). For each algorithm, we discuss how to apply it on area determination and its computational complexity.

Nearest Neighbor (NN)

NN finds the nearest neighbor to the target signals from all the areas [25, 43]. Specifically, NN calculates the Euclidean distance between target RSSI vector and each of the fingerprints at RP n [4, 24], i.e.,

$$E_n = \sum_{l=1}^{L} (t^l - \vec{f}_n^l)^2, \tag{5}$$

and finds the one with the smallest distance (or through Bayesian schemes like [53]). In our experiment, we implement the algorithm in [4, 43], find the RP with minimum signal distance and returns its area ID to the user. NN can be seamlessly applied to many existing localization systems [4]. Once the nearest neighbors in signals are discovered, we can figure out both the area and the exact location point.

Given $|\mathbf{C}|$ areas, L APs and N RPs in each area, the overall complexity using NN is $\mathcal{O}(|\mathbf{C}|LN)$. Note that $\mathcal{O}(|\mathbf{C}|N)$ is usually large in the spacious site and hence the computation is heavy. Further using k-nearest neighbors algorithm introduces higher accuracy but heavier computation.

Signal Heuristic Classification (SHC)

Based on some recent empirical studies [6], the number of APs shared between the area (say, the floor) and the maximum signal strength among these shared APs are two important factors for area localization. Specifically, we first preprocess the fingerprints to find the group of APs that covers each area to be classified. The more APs are shared between the target RSSI vectors and the group of APs within the area, the more likely that the target is there. Meanwhile, we count the number of strong signals within the shared APs. Strong RSSIs of the common APs indicate that the target is at that area. The area satisfying above two rules is returned to the user [6].

Offline computation complexity is $\mathcal{O}(|\mathbf{C}|NL)$, as we need to sweep through all RPs in sites to find the strong APs. Online complexity depends on the AP number, which takes $\mathcal{O}(|\mathbf{C}|L)$ in finding the corresponding AP features in each area.

Artificial Neural Network (ANN)

To classify different areas, we construct a traditional multi-layer neural network according to Back Propagation (BP) network [8]. We use two hidden layers, a shallow structure, to balance between accuracy and training complexity. Each neuron in the input layer corresponds to an AP, while each neuron in the output layer corresponds to a specific area. In the testing phase, the output layer returns the scores, and we find the area ID with the largest value in the output.

The online classification of ANN is fast with $\mathcal{O}(\#\text{Layers} \times L)$. However, offline computation time in training is usually heavy due to tedious training process and random parameter initialization. In practice, the number of iteration is often intractable to converge for the multi-layer neural network [32].

Deep Belief Network (DBN)

A DBN can be viewed as a composition of simple learning modules, each of which is a restricted type of *restricted Boltzmann machine* (RBM) [23]. RBM contains a layer of *visible units* that represent the input data, and a layer of *hidden units* that learn to represent features of observations and capture high-order correlations in the fingerprints. An RBM with multiple hidden units is a parametric model of the joint distribution between the hidden and observed variables. RBMs are stacked and trained in a greedy manner. Given the fingerprints at all floors (say, fingerprint data $\mathbf{F}$), we initially train the hidden units based on the RBMs [23]. Then DBN conducts unsupervised learning in order to learn the weight matrix $\mathbf{W}$ in RBM, which represents the correlation between dimensions. After such initialization, the DBN is unfolded into a neural network. Then we retrain the neural network over the given fingerprints and conduct further supervised learning and classification.

The major implementation issue with DBN is that the training process is computationally expensive due to multiple RBMs. Therefore, a greedy approach has been implemented in order to learn the deep belief network one layer at a time to reduce the learning time [19]. Once the DBN is trained, we can simply utilize the ANN to classify the RSSI vectors online, which takes $\mathcal{O}(\#\text{Layers} \times L)$.

Probabilistic Support Vector Machine (SVM)

We formulate SVM classification in “one-against-all" form. For each area, we find an SVM model to distinguish it from others, which are together considered as a single class [8]. $|\mathbf{C}|$ areas hence have $|\mathbf{C}|$ SVM models. For each online query SVM finds the probability that the target is at this area against others. Given $|\mathbf{C}|$ probabilities, we then find the area with maximum probability and return its ID to the user. In our area classification, we implement the C-SVM [9].

We implement probabilistic SVM which outputs the probability that the target is in each area. Its outputs can be further used for signal rejection or probabilistic localization. The training phase takes $\mathcal{O}(|\mathbf{C}|^3N^3)$ [9]. For the online classification, SVM takes linear time with respect to number of support vectors (SVs) for each classification, i.e., $\mathcal{O}(\#\text{SVs})$. In practice, we also find that the number of support vectors is much less than $\mathcal{O}(L)$. Given $|\mathbf{C}|$ areas, the overall computation for each area query takes $\mathcal{O}(|\mathbf{C}| \times \#\text{SVs})$, which is smaller than $\mathcal{O}(|\mathbf{C}|L)$.

SIGNAL REJECTION & DEVICE CALIBRATION

To further improve the accuracy, we discuss how to reject unclassifiable signals and calibrate heterogeneous devices.

Unclassifiable Signal Rejection

Due to measurement noise or missing of AP signals, RSSI vectors may be similar across different areas and may lead to incorrect results. Through our deployment experience, we have observed that the decision values of the algorithm may be also similar at different areas. If the area decision is highly uncertain (unclassifiable), rejecting this signal can prevent misclassification and improve the user experience. Inspired by above, we empirically study and find the rejection schemes for each algorithm as follows:

- *NN*: We reject a given RSSI query if the top k RPs with the most similar signals are not at the same floor.
- *SHC*: If multiple areas match the same signal heuristics (strongest signals within common APs), we reject this $\mathbf{T}$ for further classification.
- *ANN & DBN*: Recall that the target area is mapped to the one with the highest output value from ANN (similarly, in the output layer of DBN a neural network is applied). We accept the RSSI signal to be classifiable if the maximum in all output values are less than a given threshold α, or their standard deviation is greater than a certain β.
- *SVM*: We reject $\mathbf{T}$ if all of the probabilities calculated are less than a predefined threshold, or the difference between the largest and the second largest probabilities is less than a certain value. Specifically, we sort the p_c's in descending order and reject the query signal $\mathbf{T}$ if

$$\forall c,\ p_c \leq \lambda,\ \text{or}\ p_1 - p_2 \leq \gamma p_1, \tag{6}$$

  where p_1 and p_2 are the two largest probabilities, while λ and γ are the two predefined threshold parameters.

Besides discarding the unclassifiable Wi-Fi samples, we can also implicitly send (crowdsource [48]) them to the cloud server for further data analysis and system adaptation, which reduces the rejection rate. To determine these rejection parameters, in practical deployment, we may collect some test data for optimal parameter verification. In our experimental settings, we will provide the corresponding parameters with respect to each classification model.

Device RSSI Calibration

For the same RSSI, different smartphones may have different measurement values due to their Wi-Fi network interface difference [28]. For each target RSSI $\widetilde{t}^l$ from AP l, a linear shift d from RSSI t^l using a different device [24] has been reported, i.e., $\widetilde{t}^l = t^l + d$. We consider a scalable online calibration in order to reduce offline manual efforts. We first calculate the similarity between the target RSSI vector and that of each RP. After that, the RPs with similar signal vectors can be leveraged for online signal calibration in order to get d. RSSI vector comparison uses cosine similarity [20], denoted as $cos(\mathbf{T}, \mathbf{F}_n)$, between $\mathbf{T}$ and $\mathbf{F}_n$. The cosine similarity compares the relative signal trend of different APs rather than the absolute RSSI values. For each $\widetilde{t}^l \in \mathbf{T}$, we find the corresponding the corresponding f_n^l's at RPs from AP l. Given pairs of $[\widetilde{t}^l, f_n^l]$, we conduct the linear regression [22] and obtain the corresponding offset d for target RSSI $\widetilde{t}^l$. To mitigate the effect of random noise, we find the top several RPs with $cos(\mathbf{T}, \mathbf{F}_n) > \eta$ (say, $\eta = 0.95$ in our experiment) for linear RSSI calibration [18].

The device calibration can be conducted in a crowdsourced manner. We can leverage the ILBS user data for calibration and store these d's for different phone models. At the beginning, given the MACs and Wi-Fi interface vendors, some smartphones get online calibrated and their RSSI offsets are stored in the database. The same smartphone models of the later users can then benefit from these crowdsourced parameters.

EXPERIMENTAL EVALUATION

In this section, we present the experimental studies over these classification models. We first present the experimental settings and our comparison metrics. Then we discuss the experimental results and the deployment recommendations.

Experimental Settings & Performance Metrics

In our extensive experiment trials, we have tested the floor localization on the following sites:

- Hong Kong Cyberport (HKCP): A premium business building (Figure 1) where we collect $1,012$ fingerprints (4 m grid size) and $1,949$ target data from 3 floors (overall more than $70,000$ m^2). Samsung S3, Coolpad F1, Lenovo A680 and HTC One X are utilized for target data collection.
- Hong Kong Olympian City (HKOC): A premium shopping mall (Figure 3), where we collect $3,021$ fingerprints (5 m grid size) and $5,441$ targets from three floors (overall more than $50,000$ m^2).
- Hong Kong International Airport (HKIA): where we conduct studies over $5,592$ fingerprints or RPs (5 m grid size in totally more than $20,000$ m^2 area) and $1,950$ targets on two neighboring sites, i.e., the 5th (the departure gate) and the 7th floor (the check-in counter). We conduct experiment on these two floors as they are the authorized areas and assessed by most airline passengers (Figure 4).
- HKUST campus (Figure 5): the university campus where we collect $18,177$ fingerprints (3 m grid size) and 869 targets from 5 floors (the 2nd to the 6th floor; totally more than $15,000$ m^2 area). For indoor/outdoor detection, we also conduct extensive fingerprint collection over a $1,000$ m^2 indoor yard (320 RPs).

During the site survey, we utilize different smartphones, including HTC One X, Coolpad F1, Lenovo A680 and Samsung S3 for fingerprint and target signal collection. Our online RSSI calibration is applied when the target device model is different from that stored in the database. During fingerprint preprocessing, we filter the APs tethered by the smartphones, and combine the virtual APs according to their MAC addresses.

Note that these sites are often spacious without explicit wall partitions between them (like Figure 1). At each RP, we collect fingerprints from four different directions (north, south, west

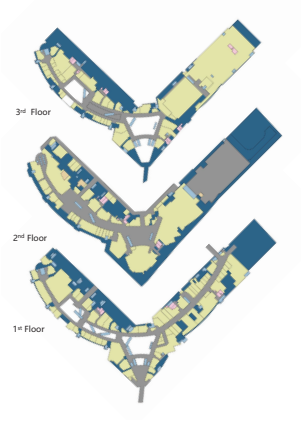

Figure 3: Floor map of the HKOP shopping mall (1st, 2nd and 3rd floor).

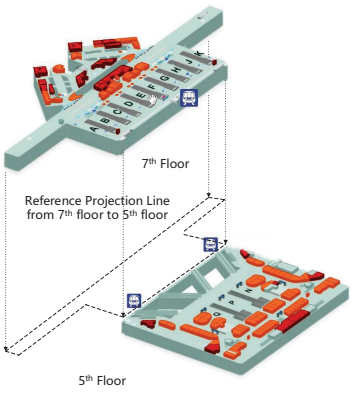

Figure 4: Floor plan of HKIA (5th floor for arrival and 7th floor for departure).

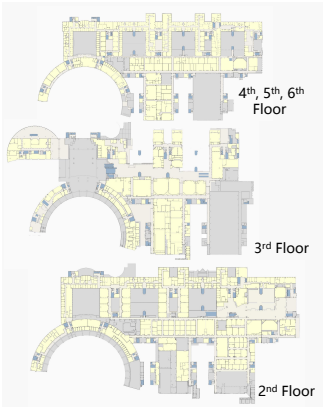

Figure 5: Floor map of HKUST campus (4th $\sim$ 6th floors have the similar floor map).

Figure 6: Site photos of indoor and outdoor sites for testing (university campus).

and east) respectively to reduce the effect of human body on signals. Before the site survey, we do not have the knowledge of AP locations and their exact number at different floors. The target data collection is collected at least 1 month after the site survey stage. Wi-Fi fingerprint and target data collection is conducted during working hours and therefore we do not exclude the cases when there are people (crowds) nearby.

We have empirically studied the optimal parameters for each algorithm, which are shown as follows. In NN, if the top $k = 5$ nearest neighbors are from different areas, this target RSSI vector is rejected. In ANN, we construct a typical four-layer artificial neural network with two hidden layers. The first hidden layer has 20 neurons, and the second one has 3 neurons. We empirically set $\alpha = 0.7$ (maximum score) and $\beta = 0.1$ (standard deviation) for signal rejection in ANN. In SVM, we use the grid search to find the parameters. We use the linear kernel, *C-SVC* [9], and $[\lambda, \gamma] = [0.6, 0.3]$ for signal rejection in our SVM. In DBN, the number of training epochs is set to be $1,000$, and 3 hidden layers and sigmoid functions are applied in its neural network classification. If the maximum score in the output layer of DBN is less than 0.95, the system rejects the signal.

To evaluate inside/outside-region detection, we also conduct indoor/outdoor switch tests near our campus hall. We use HTC One X and Lenovo A680 to collect fingerprints and query data (targets). The site survey process is similar to the above floor localization. Overall $3,158$ fingerprints (in 4 m survey grid) and $5,555$ targets are collected. In this experimental site, the glass roof structure above the campus hall allows penetration of GPS signals and makes simple GPS-based indoor/outdoor detection (i.e., whether GPS is not detected) difficult. Hence, we focus on Wi-Fi fingerprinting in the experiment. Meanwhile, we also test the IO-Detector [56] in this site.

Empirical parameter studies are also conducted for different schemes in indoor/outdoor detection, whose default settings are as follows. In PCA, we empirically select 35 principal components. The size of the self-organizing map in SOM is 40. Width parameter of RBF in SVDD is 0.5. In MDM, the width parameter in RBF is set to be 0.1.

We evaluate the area identification and inside/outside-region classifiers based on the following performance metrics:

- *Classification accuracy*: the number of correct decisions over that of all the classifiable samples (after signal rejection). It characterizes the robustness under different building structures and signal noise levels.

- *Online computational time*: which calculates the mean localization time of each given target measurement. The less time the calculation takes, the less energy the mobile device consumes and the shorter time the users have to wait for. We also measure the energy consumption of area detection to further compare their deployment efficiency [54].

- *Offline model training time:* We measure the computation time in the model learning. Based on our deployment experience, tedious offline training or fine-tuning may take great engineering efforts, while the simple model trainings are usually preferred for fast LBS deployment.

- *Rejection rate* (*RR*): the number of unclassifiable samples over the overall number of targets [33]. It also characterizes the robustness of the algorithms in classifying noisy and abnormal signals.

- *True negative rate* (TNR), which evaluates the performance of a rejection scheme, and is given by $TNR = \frac{TN}{TN+FP}$. TN denotes the number of true negative, i.e., correctly rejected samples that in fact have wrong floor mapping. FP denotes the number of false positive, i.e., wrongly rejected samples that actually can have right area mapping.

- *True positive rate* (TPR), which is given by $TPR = \frac{TP}{TP+FN}$. TP denotes the number of true positive, i.e., the accepted samples which in fact have correct floor mapping. FN denotes the number of false negative, i.e., the rejected samples which actually have correct floor mapping.

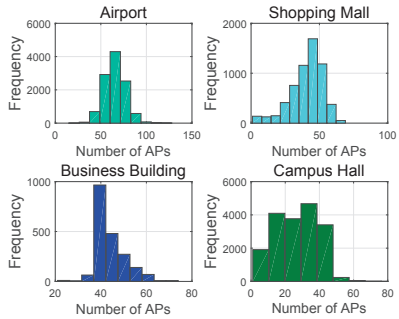

Figure 7: AP detection frequency of RPs at different sites (HTC One X).

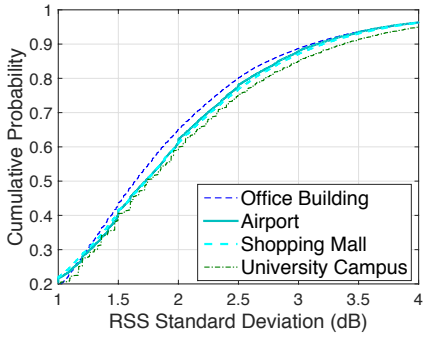

Figure 8: CDF of fingerprint signal noise (dB) in different sites using HTC One X.

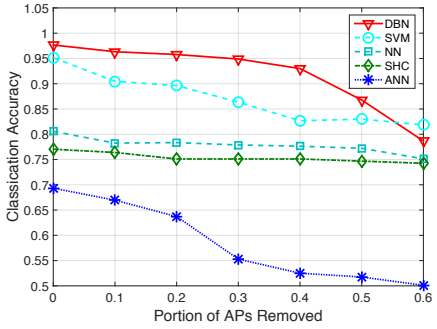

Figure 9: Accuracy versus proportion of APs removed at each target (HTC One X in the business building).

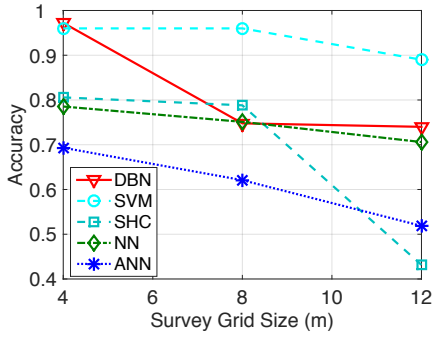

Figure 10: Accuracy versus survey grid size in the business building (HTC One X).

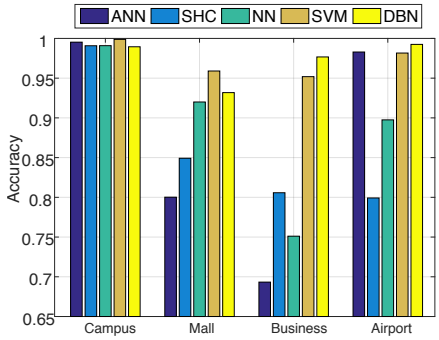

Figure 11: Accuracy of all algorithms in different sites (HTC One X).

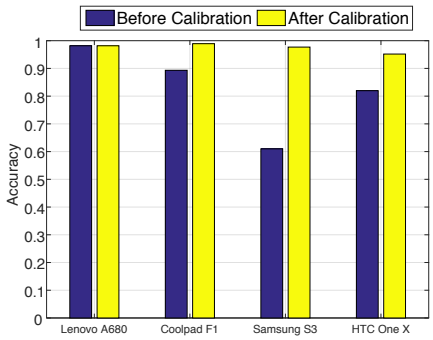

Figure 12: SVM accuracy versus different devices in the business building (RSSI fingerprinting with Lenovo A680).

We also utilize TPR and TNR for the indoor/outdoor detection. In this scenario, "positive" corresponds to indoor fingerprints, while "negative" corresponds to outdoor fingerprints.

Experimental Results & Deployment Recommendations

Note that the APs are deployed by different parties and we do not have exact knowledge of their locations and power. In the airport, overall 1,800 APs are detected. In the university campus, overall 1,702 APs are detected. In business building overall 424 APs are detected. In the shopping mall, overall 977 APs are detected. Due to the different wall partitions and AP coordination, different number of APs are detected at each RP in Figure 7. Note that in our Wi-Fi fingerprints, we do not rule out the noisy cases when there are crowds nearby. Figure 8 shows the cumulative probability of signal noise (standard deviation σ_n^l of f_n^l at each RP). We may observe that there is larger noise within the university campus and the airport than the business building due to the larger data size and more people access. From above, we expect that the site difference in AP detection and signal noise may lead to different area classification results.

Figure 9 shows the area identification accuracy versus proportion of AP removal at each target. We utilize the business building for comparison, as we have shown in the introduction that it is a typical challenging site for area classification (results in other sites are qualitatively similar). We simulate the effect of new wall partitioning or crowd blocking. As shown in Figure 9, all of their accuracy degrades with removal of APs. Loss of some important APs (strong or unique APs) leads to high signal similarity across floors. We may observe that AP loss and random noise are two important factors in the mapping error of Figure 1.

We can observe in Figure 9 that SHC suffers from missing APs as SHC largely relies on strong RSSI to differentiate the floors. SVM and DBN outperform other algorithms under all kinds of AP removal. Through optimization, the support vectors in SVM can preserve the differentiation of floors after AP removal. Multiple stacked RBMs in DBN introduce deep structure and extract good features from the fingerprints, making it less sensitive to AP loss. In the business building, due to high similarity and noisy signals, ANN suffers from overfitting in the random noise of training samples. Therefore, it cannot accurately classify the targets there.

Figure 10 shows the area identification accuracy versus survey grid size. We remove some of the RPs according to their interval (from 4 m to 8 m) in order to form different grid size. As shown in Figure 10, all algorithms degrade in classification accuracy when the survey grid size increases. Accuracy of SHC degrades under large grid size, because it greatly relies on the completeness of fingerprints. Some strong signal measurements are lost under RP removal. Overall, SVM performs with better accuracy than other algorithms. It is mainly because SVM still retrieves support vectors from the remaining signals

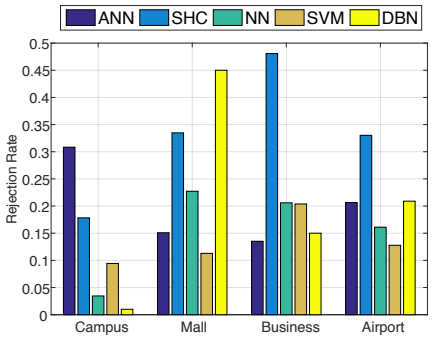

Figure 13: Rejection rate of different algorithms in all four sites (HTC One X).

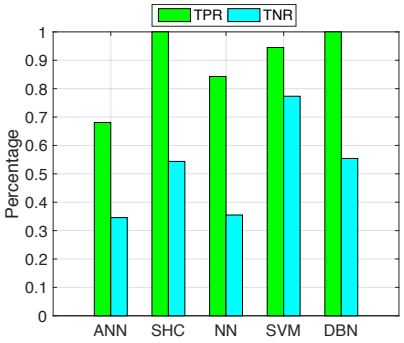

Figure 14: TNR and TPR of rejection schemes in different algorithms at the business building (HTC One X).

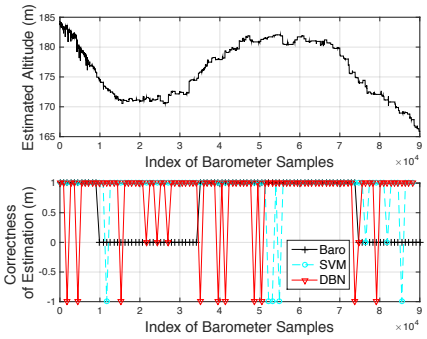

Figure 15: Comparison of barometer, RSSI-based SVM and DBN in the floor identification (the university campus).

after fingerprint reduction, which is less sensitive to training sample number. DBN significantly decreases in classification accuracy as the trained belief network largely relies on the given fingerprint data in order to represent the fingerprint map. If the size of fingerprint is significantly reduced (say, given sparse survey), its classification accuracy may decrease.

Figure 11 summarizes the area identification accuracy using different algorithms at these indoor sites. We observe that they all achieve high accuracy in the university campus, mainly because there are many wall partitions in the campus building to differentiate the RF signals, which matches the observations in [6, 25, 43]. However, some algorithms achieve lower accuracy in the business building and other spacious sites because of the noisy signals there. SVM achieves robust performance in all sites, as it finds the support vectors which can effectively differentiate the RSSI vectors in the sites. Capturing the relationship between the floors and signals, SVM is more accurate than other algorithms in the campus and the mall. The deep structure of DBN is adaptive to fewer RF fingerprints and achieves higher detection accuracy in the business building and the airport than SVM.

Figure 12 shows the area identification accuracy of probabilistic SVM using different devices in the business building before and after our online device calibration. We can observe much higher localization accuracy among different mobile devices after proposed RSSI calibration. It is mainly because the proposed online calibration successfully adjusts the RSSIs among different devices, enhancing the scalability in practical deployment. Note that after RSSI calibration, difference in accuracy is mainly because different smartphones may detect different number of APs. We have also applied our calibration scheme over other schemes, and the improvement is qualitatively similar. For brevity we do not repeat them here.

Figure 13 shows the rejection rate (*RR*) of different algorithms at these experimental sites. We may observe higher rejection rates in the shopping mall and the business building, as the fingerprint signals are similar across areas in the building, making the LBS less responsive to user query. Compared with other sites, as there are more wall partitioning in the campus, we can observe a lower rejection rate for SHC, NN and SVM. Overall, DBN and ANN have higher rejection rate in the mall and the business building. It is because the over-fitted neural network trained from the input fingerprints can be sensitive to noise in these two sites.

We compare average classification time (online) of the algorithms for each query. We focus on the performance in the business building with HTC One X as other results are qualitatively similar. Note that NN is much slower than other four algorithms (around 2.45 s per target) in this site due to large search space (3,021 RPs). SHC takes 0.2115 s for each target. SHC is faster as it only considers the APs detected at each target. SVM takes 0.0383 s as the training process retrieves vector variables $[\omega, b]$ to represent differentiation between signals. ANN takes 0.2493 ms for one target, while DBN uses average 8.10 ms for online classification. In terms of offline training time, both NN and SHC do not require extra model learning. SVM takes 2.376 s and ANN takes 1.884 s for all RF fingerprints in the business building. DBN takes overall 268.73 s due to its deep structures. We can see that DBN takes much longer training time than other schemes. For practical deployment, we highly recommend reducing the computation through approaches like parallel programming to save training time and also the implementation efficiency.

ANN and DBN are much faster than NN, SHC and SVM. It is because the trained models using ANN and DBN are more compact (only the weights are stored), leading to higher efficiency in computation. In our experiment, the DBN computation is heavier than ANN as DBN has a deeper structure with more layers and layer units. We also measure the power consumption using above different schemes in online testing based on PowerTutor [54]. On average the floor localization application with NN takes 364 mW, SHC takes 317 mW and SVM consumes 167 mW. With relatively lower consumption, ANN takes 107 mW and DBN takes 127 mW. Therefore, in order to save energy for mobiles, SVM, ANN and DBN are preferred.

Figure 14 shows the rejection TPR and TRN of different algorithms in the business building using HTC One X. We may observe that for each algorithm TNR is smaller than TPR. It is mainly because in deployment, the rejection scheme should ensure TPR with higher priority. It is mainly because once “floor jump" error happens, the user experience of LBS significantly

degrades. Note that a slightly sensitive rejection design may simply lead to negligible delay in location response. After the previous rejection, the smartphone may collect another RSSI vector in around 0.5 s, and then it conducts another floor classification in less than 0.5 s. Therefore, the user in fact does not experience much delay in the real deployment.

We have further conducted floor classification in our university campus using SVM, DBN and the basic barometer-altitude transformation [5]. We define 1 as correct decision, 0 as incorrect one and -1 as signal rejection for SVM and DBN. We have collected RSSI vectors and barometer readings in around 13 hours (around $90,000$ barometer readings). Figure 15 shows that without continuous and timely calibration the barometer experiences significantly different readings at a single location. Hence the pure barometer without external calibration (compensation) can only achieve 51.57% accuracy in our floor determination, which also matches the observations in [31, 50, 55]. On the other hand, SVM and DBN achieve better detection accuracy with RSSI. Specifically, SVM achieves 92.7% identification accuracy and DBN achieves around 86.25% accuracy.

To summarize, the advantage of the barometer over the RF signals is the low power consumption (around 10 mW) and the high accuracy given compensation. RF fingerprint-based floor detection is more robust than the barometer under dynamic thermal change. In practical point of views, deploying barometer is more suitable in measuring altitude offset rather than the absolute floor height, or fusing with other wireless signals for more general floor classification.

We have also tested each of the one-class classification algorithms for indoor/outdoor detection, which is shown in Figure 16. We can observe that the overall indoor/outdoor classification accuracy of SOM, SVDD, MPM and PCA are 84.97%, 81.35%, 88.97% and 95.69%, respectively. From above results, we may observe a better accuracy using PCA for indoor/outdoor detection. It is mainly because PCA reduces the dimensions in fingerprints and meanwhile some noisy dimensions are also filtered, leading to better classification accuracy. SOM relies on a neural network for feature space mapping, which suffers from overfitting of training data. How to combine these classifiers for detection accuracy improvement will be studied in our future work.

Furthermore, for SOM, SVDD, MPM and PCA, the TPR in one-class classification is 88.54%, 84.80%, 99.73% and 99.06%, respectively. The corresponding TNR is 80.88%, 85.72%, 82.45% and 95.60%, respectively. We observe that TPR is often higher than TNR in most schemes. It is because the indoor RSSI vectors often include strong signals due to more APs indoors, leading to easier classification. Some outdoor RSSI vectors near the boundary occasionally measure strong indoor APs and hence lead to misclassication for outdoor signals. We have also tested the IO-detector [56] in our university campus. IO-detector jointly leverages the light sensors, magnetometer and cell tower for indoor/outdoor detection. It achieves 86.76% detection accuracy (TPR is 99.9% and TNR is 71.93%) in our experimental site. We observe that when deploying in the glass-roof area the cell and light sensor

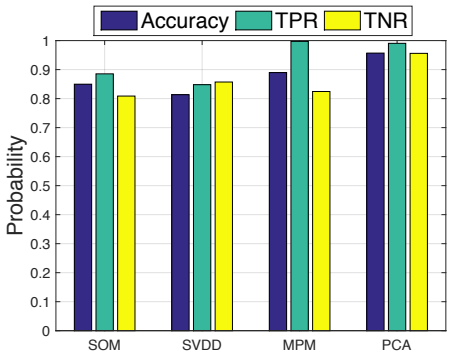

Figure 16: Accuracy/TPR/TNR comparison of SOM, SVDD, MPM and PCA in the inside/outside-region detection (campus).

readings are not differentiable between indoor/outdoor and hence the TNR is not high. We can observe that RF fingerprint-based inside/outside-region detection is more adaptive to such environment with similar light intensity and cell signals. Further sensor fusion with light sensors or cell signals can be applied with higher expense of energy consumption.

We have also compared the offline training and online running time of one-class classification algorithms. The average training time for SOM, SVDD, MPM and PCA is 106.0 ms, 7.3 ms, 0.276 ms and 2.6 ms, respectively. The corresponding average testing time for a query is 0.0410 ms, 0.0962 ms, 0.1061 ms and 0.0860 ms, respectively. SOM and PCA are both fast in classification as they first map the target RSSI vectors to lower dimensions for classification and lead to less computation. To summarize, our results show PCA data description can be a more accurate and efficient way for indoor/outdoor detection.

CONCLUSION

Area classification can be leveraged for context awareness and to improve the computation efficiency of LBS systems. Despite its benefits, there have not been systematic deployment studies over accurate and efficient area classification algorithms (inside/outside-region detection and area identification) in spacious indoor sites like airports or shopping malls.

In this paper, we study large-scale fingerprint-based area classification through extensive qualitative and quantitative studies. We study the existing challenges in finding correct indoor areas (including floors, indoor/outdoor and buildings) and efficient deployment of different schemes. Extensive experimental studies in several spacious sites have further validated our system designs and insights for area-aware classification in spacious sites. Our extensive experimental comparison studies show the practicability and deployability of efficient area determination models, proper rejection schemes of unclassifiable signals, inside/outside-region detection with the one-class classification and adaptive online device calibration.

ACKNOWLEDGMENT

We thank the paper shepherd and the anonymous reviewers for their constructive comments. This work was supported, in part, by Hong Kong Research Grant Council (RGC) General Research Fund (610713).

REFERENCES

1. Charu C. Aggarwal. 2013. *Outlier Analysis*. Springer, New York.
2. F. Alsehly, T. Arslan, and Z. Sevak. 2011. Indoor positioning with floor determination in multi story buildings. In *Proc. IPIN*. IEEE, 1–7.
3. Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. 2009. SurroundSense: Mobile Phone Localization via Ambience Fingerprinting. In *Proc. ACM MobiCom*. 261–272.
4. Paramvir Bahl and Venkata N Padmanabhan. 2000. RADAR: An in-building RF-based user location and tracking system. In *Proc. IEEE INFOCOM*. 775–784.
5. Mário N. Berberan-Santos, Evgeny N. Bodunov, and Lionello Pogliani. 1997. On the barometric formula. *American Journal of Physics* 65, 5 (1997), 404–412.
6. Preeti Bhargava, Shivsubramani Krishnamoorthy, Aditya Karkada Nakshathri, Matthew Mah, and Ashok Agrawala. 2013. Locus: An indoor localization, tracking and navigation system for multi-story buildings using heuristics derived from Wi-Fi signal strength. In *Proc. MobiQuitous*. Springer, 212–223.
7. Christopher M Bishop. 1995. *Neural networks for pattern recognition*. Oxford University Press.
8. Christopher M Bishop and others. 2006. *Pattern Recognition and Machine Learning*. Vol. 1. Springer New York.
9. Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Trans. Intelligent Systems & Technology (TIST)* 2, 3 (2011), 27.
10. S. H. Fang and T. Lin. 2012. Principal Component Localization in Indoor WLAN Environments. *IEEE Trans. Mobile Computing* 11, 1 (Jan 2012), 100–110.
11. Laurent E. Ghaoui, Michael I. Jordan, and Gert R. Lanckriet. 2003. Robust Novelty Detection with Single-Class MPM. In *Advances in Neural Information Processing Systems 15 (NIPS)*. MIT Press, 929–936.
12. Tao Gu, Hung Keng Pung, and Da Qing Zhang. 2004. A middleware for building context-aware mobile services. In *Proc. IEEE VTC-Spring*, Vol. 5. 2656–2660 Vol.5.
13. Dongsoo Han, Sukhoon Jung, Minkyu Lee, and Giwan Yoon. 2014. Building a Practical Wi-Fi-Based Indoor Navigation System. *IEEE Pervasive Computing* 13, 2 (Apr. 2014), 72–79.
14. Suining He and S.-H. Gary Chan. 2016a. Tilejunction: Mitigating Signal Noise for Fingerprint-Based Indoor Localization. *IEEE Trans. Mobile Computing* 15, 6 (June 2016), 1554–1568.
15. Suining He and S.-H. Gary Chan. 2016b. Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons. *IEEE Communications Surveys Tutorials* 18, 1 (First quarter 2016), 466–490.
16. Suining He, S.-H. Gary Chan, Lei Yu, and Ning Liu. 2015a. Calibration-free Fusion of Step Counter and Wireless Fingerprints for Indoor Localization. In *Proc. ACM UbiComp*. 897–908.
17. Suining He, S.-H. Gary Chan, Lei Yu, and Ning Liu. 2015b. Fusing noisy fingerprints with distance bounds for indoor localization. In *Proc. IEEE INFOCOM*. 2506–2514.
18. Suining He, Tianyang Hu, and S.-H. Gary Chan. 2015c. Contour-based Trilateration for Indoor Fingerprinting Localization. In *Proc. ACM SenSys*. 225–238.
19. Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* 18, 7 (July 2006), 1527–1554.
20. Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. *Recommender Systems: An Introduction*. Cambridge University Press.
21. Teuvo Kohonen. 1998. The self-organizing map. *Neurocomputing* 21, 1-3 (1998), 1 – 6.
22. Christos Laoudias, Demetrios Zeinalipour-Yazti, and Christos G Panayiotou. 2013. Crowdsourced indoor localization for diverse devices through radiomap fusion. In *Proc. IPIN*. 1–7.
23. Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. 2007. An Empirical Evaluation of Deep Architectures on Problems with Many Factors of Variation. In *Proc. ICML*. ACM, New York, NY, USA, 473–480.
24. Liqun Li, Guobin Shen, Chunshui Zhao, Thomas Moscibroda, Jyh-Han Lin, and Feng Zhao. 2014. Experiencing and handling the diversity in data density and environmental locality in an indoor positioning service. In *Proc. ACM MobiCom*. 459–470.
25. Heng-Xiu Liu, Bo-An Chen, Po-Hsuan Tseng, Kai-Ten Feng, and Tian-Sheng Wang. 2015. Map-Aware Indoor Area Estimation with Shortest Path Based on RSS Fingerprinting. In *Proc. IEEE VTC Spring*. 1–5.
26. Chengwen Luo, Hande Hong, and Mun Choon Chan. 2014. PiLoc: A self-calibrating participatory indoor localization system. In *Proc. ACM/IEEE IPSN*. 143–154.
27. 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 *Proc. ACM/IEEE IPSN*. 178–189.
28. A.K.M. Mahtab Hossain, Yunye Jin, Wee-Seng Soh, and Hien Nguyen Van. 2013. SSD: A Robust RF Location Fingerprint Addressing Mobile Devices' Heterogeneity. *IEEE Trans. Mobile Computing* 12, 1 (Jan. 2013), 65–77.
29. Larry M. Manevitz and Malik Yousef. 2002. One-class SVMs for Document Classification. *J. Mach. Learn. Res.* 2 (March 2002), 139–154.

30. F. Montorsi, F. Pancaldi, and G. M. Vitetta. 2014. Map-Aware Models for Indoor Wireless Localization Systems: An Experimental Study. *IEEE Trans. Wireless Communications* 13, 5 (May 2014), 2850–2862.

31. Kartik Muralidharan, Azeem Javed Khan, Archan Misra, Rajesh Krishna Balan, and Sharad Agarwal. 2014. Barometric phone sensors: More hype than hope!. In *Proc. ACM HotMobile*. 12:1–12:6.

32. Pekka Orponen. 1994. Computational Complexity of Neural Networks: A Survey. *Nordic J. of Computing* 1, 1 (March 1994), 94–110.

33. Ignazio Pillai, Giorgio Fumera, and Fabio Roli. 2011. A Classification Approach with a Reject Option for Multi-label Problems. In *Image Analysis and Processing-ICIAP 2011*. Lecture Notes in Computer Science, Vol. 6978. Springer Berlin Heidelberg, 98–107.

34. Valentin Radu, Panagiota Katsikouli, Rik Sarkar, and Mahesh K. Marina. 2014. A Semi-supervised Learning Approach for Robust Indoor-outdoor Detection with Smartphones. In *Proc. ACM SenSys*. 280–294.

35. Lenin Ravindranath, Calvin Newport, Hari Balakrishnan, and Samuel Madden. 2011. Improving Wireless Network Performance Using Sensor Hints. In *Proc. NSDI*. 281–294.

36. Bernhard Schölkopf, Robert C Williamson, Alex J. Smola, John Shawe-Taylor, and John C. Platt. 2000. Support Vector Method for Novelty Detection. In *Advances in Neural Information Processing Systems 12 (NIPS)*. MIT Press, 582–588.

37. N. Serrano, A. Savakis, and Jiebo Luo. 2002. A computationally efficient approach to indoor/outdoor scene classification. In *Proc. International Conference on Pattern Recognition*, Vol. 4. 146–149.

38. Xingfa Shen, Yueshen Chen, Jianhui Zhang, Landi Wang, Guojun Dai, and Tian He. 2015. BarFi: Barometer-aided WiFi Floor Localization Using Crowdsourcing. In *Proc. IEEE MASS*. 416–424.

39. Stephen P. Tarzia, Peter A. Dinda, Robert P. Dick, and Gokhan Memik. 2011. Indoor Localization Without Infrastructure Using the Acoustic Background Spectrum. In *Proc. ACM MobiSys*. 155–168.

40. D.M.J. Tax. 2015. DDtools, the Data Description Toolbox for Matlab. (June 2015). version 2.1.2.

41. David Martinus Johannes Tax. 2001. *One-class classification*. TU Delft, Delft University of Technology.

42. Yu-Chih Tung and Kang G. Shin. 2015. EchoTag: Accurate Infrastructure-Free Indoor Location Tagging with Smartphones. In *Proc. ACM MobiCom*. 525–536.

43. Alex Varshavsky, Anthony LaMarca, Jeffrey Hightower, and Eyal de Lara. 2007. The SkyLoc floor localization system. In *Proc. IEEE PerCom*. 125–134.

44. He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. 2012. No Need to War-drive: Unsupervised Indoor Localization. In *Proc. ACM MobiSys*. 197–210.

45. Zhuoling Xiao, Hongkai Wen, Andrew Markham, and Niki Trigoni. 2014. Lightweight Map Matching for Indoor Localisation Using Conditional Random Fields. In *Proc. ACM/IEEE IPSN*. 131–142.

46. Zhice Yang, Zeyu Wang, Jiansong Zhang, Chenyu Huang, and Qian Zhang. 2015. Wearables Can Afford: Light-weight Indoor Positioning with Visible Light. In *Proc. ACM MobiSys*. 317–330.

47. Haibo Ye, Tao Gu, Xianping Tao, and Jian Lu. 2014a. B-Loc: Scalable Floor Localization Using Barometer on Smartphone. In *Proc. IEEE MASS*. 127–135.

48. H. Ye, T. Gu, X. Tao, and J. Lu. 2014b. Crowdsourced smartphone sensing for localization in metro trains. In *Proc. IEEE WoWMoM*. 1–9.

49. Haibo Ye, Tao Gu, Xianping Tao, and Jian Lu. 2014c. F-Loc: Floor localization via crowdsourcing. In *Proc. IEEE ICPADS*. 47–54.

50. Haibo Ye, Tao Gu, Xianping Tao, and Jian Lu. 2014d. SBC: Scalable Smartphone Barometer Calibration Through Crowdsourcing. In *Proc. MobiQuitous*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 60–69.

51. Haibo Ye, Tao Gu, Xiaorui Zhu, Jinwei Xu, Xianping Tao, Jian Lu, and Ning Jin. 2012. FTrack: Infrastructure-free floor localization via mobile phone sensing. In *Proc. IEEE PerCom*. 2–10.

52. Moustafa Youssef. 2015. Towards Truly Ubiquitous Indoor Localization on a Worldwide Scale. In *Proc. SIGSPATIAL (GIS '15)*. Article 12, 4 pages.

53. Moustafa Youssef and Ashok Agrawala. 2008. The Horus location determination system. *Wireless Networks* 14, 3 (2008), 357–374.

54. Lide Zhang, Birjodh Tiwana, Zhiyun Qian, Zhaoguang Wang, Robert P. Dick, Zhuoqing Morley Mao, and Lei Yang. 2010. Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones. In *Proc. IEEE/ACM/IFIP CODES/ISSS*. 105–114.

55. F. Zhao, H. Luo, X. Zhao, Z. Pang, and H. Park. 2015. HYFI: Hybrid Floor Identification Based on Wireless Fingerprinting and Barometric Pressure. *IEEE Transactions on Industrial Informatics* (2015).

56. Pengfei Zhou, Yuanqing Zheng, Zhenjiang Li, Mo Li, and Guobin Shen. 2012. IODetector: A Generic Service for Indoor Outdoor Detection. In *Proc. ACM SenSys*. 113–126.