

Gain Without Pain: Enabling Fingerprinting-based Indoor Localization using Tracking Scanners

Hamada Rizk
hamada.rizk@ejust.edu.eg
Osaka University, Japan
E-JUST, Alexandria, Egypt
Tanta University, Tanta, Egypt

Moustafa Youssef
moustafa@alexu.edu.eg
Alexandria University
Alexandria, Egypt

Hirozumi Yamaguchi
h-yamagu@ist.osaka-u.ac.jp
Osaka University
Osaka, Japan

Teruo Higashino
higashino@ist.osaka-u.ac.jp
Osaka University
Osaka, Japan

ABSTRACT

Robust and accurate indoor localization has been the goal of several research efforts over the past decade. Towards achieving this goal, WiFi fingerprinting-based indoor localization systems have been proposed. However, fingerprinting involves significant effort; especially when done at high density; and needs to be repeated with any change in the deployment area. While a number of recent systems have been introduced to reduce the calibration effort, these still trade overhead with accuracy.

In this paper, we present *LiPhi*: an accurate system for enabling fingerprinting-based indoor localization systems without the associated data collection overhead. This is achieved by leveraging the sensing capability of transportable laser range scanners (LRSs) to automatically label WiFi signal scans, which can subsequently be used to build (and maintain) localization models. As part of its design, *LiPhi* has modules to associate WiFi scans with the unlabeled traces obtained from as few as one LRS as well as provisions to train a robust deep learning model.

Evaluation of *LiPhi* using Android phones in two realistic testbeds shows that it can match the performance of manual fingerprinting techniques under the same deployment conditions without the overhead associated with the traditional fingerprinting process. In addition, *LiPhi* improves upon the median localization accuracy obtained from crowdsourcing-based and fingerprinting-based systems by 181% and 297% respectively, when tested with data collected a few months later.

CCS CONCEPTS

• **Networks** → **Location based services**; • **Human-centered computing** → *Ubiquitous and mobile computing*;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGSPATIAL '20, November 3–6, 2020, Seattle, WA, USA

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-8019-5/20/11...\$15.00

<https://doi.org/10.1145/3397536.3422207>

ACM Reference Format:

Hamada Rizk, Hirozumi Yamaguchi, Moustafa Youssef, and Teruo Higashino. 2020. Gain Without Pain: Enabling Fingerprinting-based Indoor Localization using Tracking Scanners. In *28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '20)*, November 3–6, 2020, Seattle, WA, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3397536.3422207>

1 INTRODUCTION

Recent years have witnessed the advent of indoor localization systems harnessing the capabilities of smartphones [36]. At the core of indoor localization systems, WiFi-based systems are built on the ubiquitous deployment of WiFi technology.

Towards making WiFi-based localization possible at the intended accuracy level, fingerprint-based approaches are proposed as they have been shown to be capable of providing accurate, fine-grained positioning [1, 10, 11, 25, 32, 33, 37, 39]. Fingerprinting works in two phases: offline and online. During the offline phase, the received signal strengths (i.e. RSS), from the access points (APs) installed in the area of interest, are recorded by a cell phone carried by the site surveyor. This is done at specific reference points. The surveyor has to remain at the exact points over a relatively long time and tag these recorded data manually through the use of a data collection application. The collected fingerprint is then leveraged to build a localization model that can be either probabilistic methods, e.g. [37], or machine learning-based, e.g. [1]. During the online phase, this model can be subsequently be used to obtain real time estimate for the user's location.

The data collection required for fingerprinting is time-consuming, vulnerable to environmental change, laborious, and unscalable; especially in large testbeds. To tackle this problem, several techniques have been proposed including using robots, additional sensors, computer vision, crowd-sensing, propagation models, and/or data interpolation techniques [7, 9, 14, 27]. These approaches either do not account for the effect of humans, or reduce the ubiquity and/or accuracy of the localization system, or may have other limitations (e.g., require a complex calibration process, raise privacy concerns, require interaction from the users, etc).

In this paper, we propose *LiPhi*: a system for seamlessly enabling fingerprinting-based indoor localization **without the associated data collection overhead** inherent in the traditional fingerprinting method. The idea is to opportunistically leverage



Figure 1: Our transportable LRS-based tracking system.

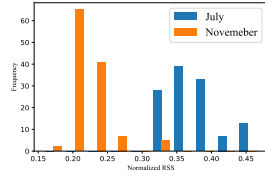


Figure 2: RSS distributions collected in two different months.

transportable Laser-Range Scanners (LRSs or LiDARs) in a user-transparent way to tag WiFi scans collected during the normal movement of building users without human intervention. *LiPhi* can also construct the required fingerprint database and build the localization model with as few as only one LRS. In addition, the used LRSs can be temporarily deployed and then reused in other buildings (Figure 1), significantly reducing the overhead and cost of deploying a WiFi localization system.

Nevertheless, *LiPhi* needs to make provision for a number of challenges including handling instantaneous changes of the WiFi signals which could yield spurious points in the estimated traces, matching **anonymous** LRS traces (i.e. sequence of point labels) to the WiFi scans collected by the identified user, and constructing robust deep learning localization models. For this, we introduce a novel iterative user trace refinement approach that uses temporal and spatial location smoothing and outlier detection to ensure the validity of the initial trace shape. This estimated user trace is matched to the available LRS traces collected at the same time with the lowest cumulative point-wise distance. These automatically-labelled user traces constitute a large (i.e. dense) fingerprint database which enables the effective training of deep learning model. Additionally, *LiPhi* includes provisions to ensure the generalization and robustness of the trained model against overfitting.

We implemented *LiPhi* on Android devices in two different testbeds. Our results show that *LiPhi* outperforms the state-of-the-art indoor positioning techniques in both testbeds by at least 181%. This accuracy is obtained without any fingerprinting overhead or user intervention along with robustness to temporal variations of the signals and infrastructure.

The rest of this paper is structured as follows: Section 2 gives a background on laser range finders and discusses practical issues facing WiFi-based fingerprinting localization. In Section 3, we provide an overview of *LiPhi*. Section 4 presents in detail the methodology proposed by *LiPhi*. In Section 5, we describe the data collection process and provide the detailed evaluation of the system. In Section 6 we discuss the researches carried out in literature and most relevant to *LiPhi*. Finally, we conclude the paper in Section 7.

2 BACKGROUND AND MOTIVATION

In this section, we start by a background on laser range scanners. Then, we discuss issues that need to be addressed by traditional fingerprinting techniques.

2.1 Laser Range Scanners

Laser-Range Scanners (LRSs) are devices that detect surrounding objects using eye-safe lasers. A laser beam scans the scene in one

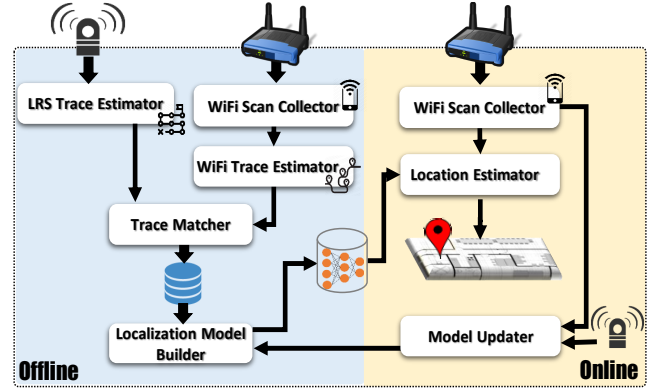


Figure 3: *LiPhi* system architecture.

or two dimensions and can obtain accurate distance at each angle with **sub-decimeter** errors and with high frequency (e.g., 20 scans per seconds at each angle). Since the LRS units become less expensive and more popular, they can be used more in large indoor environments such as malls and museums. Our team has developed a LiDAR-based tracking system (Figure 1) and has deployed it in a shopping mall for the purpose of developing *LiPhi*. As shown from the figure, it is battery-operated with a place-and-play feature, using Raspberry Pi 3 and an LTE module. Therefore, the setup overhead cost of the LiDAR-based tracking system is negligible (which includes placing the LiDAR in a specific location and marking its location on the map). Using LiDARs enables the collection of large amounts of dense data which facilitates the effective application of accurate, though data-hungry solutions (e.g. deep learning).

2.2 Fingerprint Construction Overhead

In this section, we quantify the site survey overhead (in terms of time) which is incurred by a typical manual fingerprinting process. Given n reference points in the area of interest and assuming the time for collecting the information at each point is τ minutes, and a constant time for moving to new point and tagging the location the surveyor is standing at (setup time, denoted by α); then the site survey overhead is computed as $n \times (\tau + \alpha)$. For instance, to construct a fingerprint database for a shopping mall at 1000 points with $\tau = 5$ mins and $\alpha = 1$ mins, this would require 100 hours in data collection, highlighting the massive overhead inherent in manual fingerprinting. Furthermore, this time cost has to be paid with every change in the testbed (e.g. moving an AP to a new location). *In contrast, LiPhi requires zero-extra overhead as fingerprint database is constructed opportunistically and transparently from the users of the building in their normal movements in the environment. Additionally, the used LiDAR-based tracking system covers large area and is easily transportable with negligible overhead.*

2.3 Temporal Variations

The RSS values from an arbitrary AP at a given location have two types of variations: *Short-duration variations* and *Long-duration variations*. *Short-duration variations* are due to occasional environment variations, e.g. user's movements, and lead to drastic variations in the RSS measurements received from some APs over a short time. These types of fluctuations usually lead to location estimates

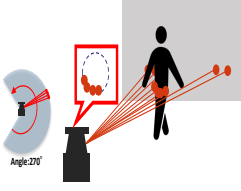


Figure 4: Example of LRS human tracking.

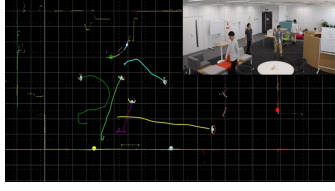


Figure 5: Visualized human traces as detected by the LRS-based Trace Estimator module.

which remarkably deviate from their preceding or subsequent estimations and can be considered as outliers. On the other hand, *Long-duration variations* refer to attenuation of the signals due to lasting changes in the environment which permanently affect the quality of the fingerprint database on which the localization model is built. Figure 2 shows the two RSS distributions of an arbitrary AP at the same location in two different months (July and November) which depicts how the distribution changes over time. This change can be seen as an inadvertent covariate shift [28] and leads to a remarkable drop in the accuracy of the trained localization model. This effect is empirically quantified in Section 5. To tackle this issue, the fingerprint database would need to be rebuilt, which would involve the associated overhead cost estimated in the previous section. By leveraging the laser range scanners, *LiPhi* can keep the fingerprint up-to-date, significantly reducing the temporal variations effects with zero extra overhead.

On the other hand, the installed APs and/or their parameters (e.g. transmission power) may change over time due to maintenance/replacement or addition of new APs. These changes lead to variation in the APs density in the area of interest. For example, during the development of this work, the APs in the building of our Lab have been maintained/replaced leading to a severe decrease of the originally installed APs by 30%. This in turn, negatively affects the already-trained traditional localization model as it loses significant information while it cannot consider the new APs. *This highlights the importance of continuously updating the fingerprint as in LiPhi.*

3 SYSTEM OVERVIEW AND MATHEMATICAL MODEL

3.1 System Overview

Figure 3 shows the architecture of the proposed system. *LiPhi* has two phases: an offline phase and online tracking phase. In the offline phase, *LiPhi* aims to *automatically* construct a fingerprint database and a deep localization model. Towards achieving these goals, the system designer initializes the offline stage by feeding the floorplan layout of the environment with APs' locations to the system¹. For constructing the required fingerprint database, the building users scan the WiFi measurements from the deployed APs in the area of interest using the **WiFi Scan Collector** module in a crowdsourcing manner. This module is an application installed on the user's phone. Note that these scans are collected without any

manual intervention from the users. Hence it does not require user feedback as in previous systems, e.g. [12]. However, the location of the collected scans can be coarsely estimated by the **WiFi Trace Estimator** module based on a propagation model as verified in [6]. Simultaneously, the deployed laser range scanners detect moving objects (i.e. users) in the considered environment. The LRSs scans are further processed by **LRS Trace Estimator** module to obtain the sequence of pedestrian positions forming a trace. Then, the **Trace Matcher** module is responsible for matching and correcting the estimated WiFi-based trace by the location tags of most similar LRS-based trace along the available walking paths. As a result, a fingerprint database is constructed of timestamped WiFi scans and annotated by its corresponding LRS-based labels. This database is leveraged by the **Localization Model Builder** to train a deep neural network, which is used later in the online phase.

During the online phase, the user carrying her phone at unknown location scans for WiFi information from the detectable APs in the area of interest. These scans are then forwarded to the *LiPhi* server. The **Location Estimator** module feeds the data to the localization model constructed in the offline phase to estimate the current user location.

3.2 Mathematical Model

Without a loss of generality, we assume the user's phone is tracked in a 2D indoor environment $\mathbb{L}$ containing m access points and q LRS devices. The user is at unknown location $l \in \mathbb{L}$ carrying a device receiving scans for the nearby APs. Let an arbitrary WiFi scan be represented as $x_i = \{x_{i1}, \dots, x_{in}\}$, where $n \leq m$ and the j^{th} entry is the RSS measurement from the j^{th} AP in the i^{th} scan. In the offline phase, the problem is formally-expressed as: given a signal strength vector $x_i = \{x_{i1}, \dots, x_{in}\}$, we seek to find its corresponding fingerprint location l_i . Suitably-labeled scans will be used in constructing a fingerprint database which can then be used in training a localization model. Noting the fact that the RSS received by a user's phone from a transmitting AP is negatively correlated to the distance between the phone and the AP. Thus, given the coordinates of a subset $n_z \in n$ of APs with known locations (called reference APs), the user location can be *coarsely* estimated. The set of LRSs installed in the area of interest can be used to anonymously and accurately estimate the user trajectory without determining the user identity. The LRS-based trace is a sequence of locations $l = \{l_1, l_2, \dots, l_k\}$ of length k which is matched to the sequence of WiFi-based estimated locations $r = \{r_1, r_2, \dots, r_k\}$ in order to refine the WiFi-based estimates (i.e. tags). On the other hand, the problem in the online phase becomes: given a RSS vector $x_i = \{x_{i1}, \dots, x_{in}\}$, we aim at finding the location l_i that maximizes the probability $P(l_i|x_i)$. For that, the deep localization model that is trained in the offline phase can be harnessed.

In the next section, we discuss the details of how *LiPhi* builds an fingerprint database and permits robust localization in continuous space.

4 THE LIPHI SYSTEM

We present the details of the offline phase which include the automatic fingerprint construction and training a localization model in addition to how this model is queried in the online phase to provide fine-grained location estimates.

¹This information can be obtained from the building CAD information or automatically determined using crowdsourcing techniques as in [3].

4.1 LRS-based Trace Estimator

In this paper, we leverage simple, low-cost, commercial off-the-shelf 2D scanners. An LRS unit used in our experiment can scan 1080 points horizontally, at an angle of 270 degrees within 25ms. Without loss of generality, we adopt the tracking method in [35] for trajectory estimation based on LRSs. Each LRS unit horizontally scans the environment, and *LiPhi* calculates the difference between the current distance and the background distance (e.g., walls and fixed objects). Then *LiPhi* detects a group of such points that correspond to a single person's waist (ellipse) and estimates the location of the user as the center of the arc-shape (Figure 4). This method forms the user trace by connecting the subsequent nearby estimated locations as shown in Figure 5. It is worth mentioning that *LiPhi* can operate with only a single LRS by covering the whole area of interest incrementally over time by moving the LRS to different sub-areas.

4.2 WiFi-based Trace Estimator

The goal of this module is to obtain a coarse-grained estimate of the unknown phone's location given the corresponding WiFi RSS information. The guiding principle utilized here is that the measured RSSs from different APs heard by a particular user device can ideally represent the distance between the AP and that device. In other words, the stronger the RSS received from an AP, the closer the device should be to this AP [6]. Building on this observation, we define two sub-modules: *Spatial Rule Generator* and *WiFi Scan Annotator* to enable a transparent estimation of the user location in real-time.

4.2.1 Spatial Rule Generator. This module aims to define rules (referred to as spatial rules) that characterize every location in the environment based on its distance to the **reference APs** (i.e. APs whose locations are known). These rules represent the expected RSS relations between every possible pair of reference APs at each location. To enable the effective calculation of spatial rules, *LiPhi* discretizes the area by creating a grid which is virtually superimposed on the layout of the area of interest and the center of each grid cell is referred to as a reference point. The density of the grid cells is a configurable parameter which could be selected to balance between computational overhead and accuracy (Section 5.2.1). As a result, several reference points (i.e. grid cell points) will be spatially defined spanning the area of interest. Given an arbitrary defined point in the area, the Euclidean distances between that point and every detectable reference AP is calculated. Then, for each point, the spatial pairwise rules are derived based on the distance between the point in question and every possible pair of reference APs. In particular, assume there are two reference APs *A* and *B* with distances d_A and d_B from the point, respectively. A spatial rule can be defined as $d_B < d_A$ if *B* is closer to that point than *A*. It is expected that this rule should apply to the signal strengths detected by the mobile device at that point. That is, the RSS level from *B* should be expectedly higher than the RSS level from *A* forming an RSS rule $x_B > x_A$ (Figure 6). For m APs present in the area of interest, a total of $\binom{m}{2}$ pairwise spatial rules are defined for each point. Note that considering relative relations between different reference APs rather than the absolute RSS values lead to device-invariant location estimates.

4.2.2 WiFi Scan Annotator. The goal of this module is to estimate the user's most probable location, which is then used to annotate the corresponding WiFi scan. This can be done by locating the user's mobile device based on the measured RSS signals received from the reference APs during the normal movement of the users of the building (i.e. crowdsourcing). Subsequently, the location of user's device is estimated using the pre-calculated spatial rules of each point. During the resolution process, short-term variations in the wireless channel (e.g arising from link blockages between the device and the reference APs) may lead to erroneous matches. To combat this, this module employs a scoring approach, outlier removal and smoothing techniques, described as follows.

The process of finding the user's location is actually resolved to a number of best-matching reference points, as selected by their matching score. More specifically, all reference points are initialized with a score of zero (i.e. $s = 0$). Matches between the given RSS rules and the rules for a given reference point increase the score for that reference point by 1 for each successful match. Naturally, this implies that a number of reference points (if not all) will have nonzero matching scores. Therefore, the points with the top 10% scores are selected and their centroid (i.e as a spatial location) is reported as the user's location with an assigned confidence score equaling the highest score amongst the selected points.

To further boost the robustness against the occasional environmental dynamics, *LiPhi* leverages multiple consecutive location estimates in its operation, which constitutes a window of v locations. In particular, given a window of v locations (i.e. consecutive estimates in time), *LiPhi* removes the anomalous location estimates that may deviate from the majority of scans in that window. As a result, the predicted consecutive locations are ensured to be correlated. To do that, it calculates the distance between each estimated location and the remaining $v - 1$ reported locations, a value we term inter-location distance and define as [19]:

$$d_i = \sum_{j \neq i} \frac{d(r_i, r_j)}{v - 1} \quad (1)$$

where $d(r_i, r_j)$ is the Euclidean distance between locations r_i and r_j . Then, we calculate the average d_{avg} of inter-location distance of all locations. This module rejects the location estimates that do not satisfy $d_i \leq d_{avg}$. Finally, a robust estimate of the user location r^* is calculated as the center of mass (i.e. weighted average) of the remaining u locations as:

$$r^* = \frac{\sum_{i=1}^u s_i r_i}{\sum_{i=1}^u s_i} \quad (2)$$

where r_i is the i^{th} reported location after avoiding outliers and s_i is its corresponding confidence score. In calculating this average, the weight used for a given location is its corresponding normalized matching score.

4.3 Trace Matcher

We perform trace matching to refine the labels obtained by WiFi location query resolution. LRS location estimates and WiFi predicted labels are synchronized using the same clock. Therefore, *LiPhi* defines a matching window over which WiFi predicted labels are matched to the LRS-based estimated locations along the walking path. Matching over windows helps the system work well even

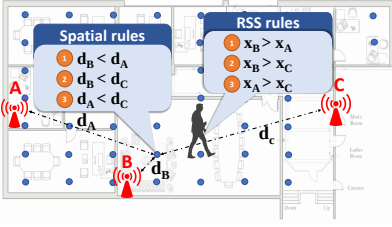


Figure 6: An example of the virtual grid, the spatial and RSS rules.

with **low LRS coverage** as the matching window could be longer, thereby leveraging the LRS estimates at the start and the end of the trace. Even though the location estimated from WiFi at each timestep is not perfectly accurate, the general shape of the trace can uniquely identify the correct walking path in the area of interest reported by the LRS units, especially after rejection of anomalous location estimates. This can be seen from Figure 7. *LiPhi* starts with maximum uncertainty about the correct trace and reduces ambiguity incrementally over time by buffering estimations in a window of length k . The system continuously matches the timestamped WiFi-based labels inside that buffer to the corresponding labels generated by LRS traces. When a trace ambiguity is detected, *LiPhi* starts looking into the buffer to find a trace $c = [\ell_1, \dots, \ell_k]$ such that:

$$c^* = \arg \min_{c \in C} \sum_{i=1}^k \|\ell_i - r_i\| \quad (3)$$

where $\|\ell_i - r_i\|$ is the Euclidean distance between LRS-derived estimate and the WiFi-derived estimate at the i^{th} timestep stored in a buffer of length k . Thereafter, all the buffered WiFi scans are annotated with corresponding timestamped location from the LRS-derived labels.

4.4 Localization Model Constructor

This module is responsible for building a localization model using the WiFi fingerprints constructed in the previous module.

4.4.1 The Network Architecture. The architecture of the deep neural network is shown in Figure 8. *LiPhi* employs a fully-connected architecture due to its hierarchical representational ability which allows the learning of complex patterns [21]. Deep networks, in general, consist of three types of layers: input, hidden and output. The input layer of the proposed network consists of the RSS readings received from all APs detected in the area of interest. The RSS of unheard APs in an input scan is set to $-100dBm$. Therefore the size of input vector of the localization model will remain fixed between different scans. *LiPhi* incorporates a deep network, consisting of stacked nonlinear hidden layers. We experimented with different activation functions and found that hyperbolic tangent function (tanh) yields the best performance. The selected network employs a regression function so that the user's location is expressed in continuous space. Therefore, the output layer consists of two neurons corresponding to the 2D spatial coordinates (x, y) . A linear activation function is adopted in the output layer for making a prediction. The selected model is trained using the Adaptive Moment Estimation (Adam) optimizer with the mean square error (MSE) between

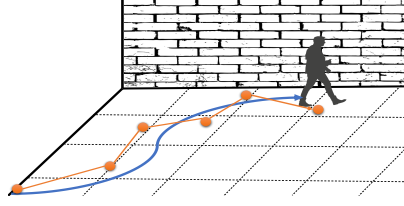


Figure 7: Example of the LRS estimated path (blue) and the RSS estimated trace (orange).

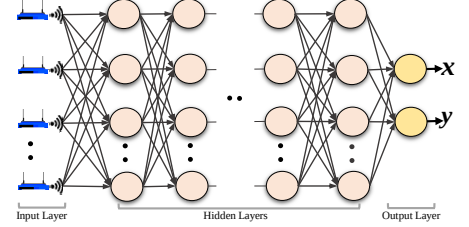


Figure 8: The neural network architecture of the *LiPhi* system.

the estimated output locations and the true output locations as the chosen loss function.

4.4.2 Ensuring Model Robustness. A fully-connected architecture is adopted due to its ability to capture the inter-relationship between different APs present in the same scan. On the other hand, it builds undesirable co-dependence between the model's inputs (i.e. different APs) which may lead to deterioration in the *LiPhi* performance with the high APs' fluctuation. To increase the model robustness against over-fitting and **variation of APs' density** and also accelerate the training process, *LiPhi* employs data augmentation and batch normalization techniques. The number of APs detected at a fixed location varies with time as discussed in Section 2.3. *LiPhi* handles this variation of APs' density problem by randomly switching-off some of the APs in the training samples and generating new synthetic samples reflecting this behaviour. This has the effect of making the training process noisy, forcing the model to learn and generalize from incomplete yet plausible input vectors. Specifically, *LiPhi* generates a binary mask that can be multiplied by the RSS input vector to selectively drop certain APs in a scan. This trick helps the model generalize and take account of such practical phenomena, and also provides the required data for efficient training of the model as discussed in Section 5.2.6. Additionally, device heterogeneity and temporal variations (discussed in Section 2.3) shift the signals' magnitude with some variance [16, 38]. Therefore, we propose an augmentation approach called *noise-shifting* to emulate this behavior. It involves the addition of white Gaussian noise to the different entries of the RSS vectors to handle this issue. Noise-shifting yields improved performance relative to the base case. This is due to the variation added to the underlying data distribution for every AP which further enhances the generalization ability of the deep model.

The number of epochs is automatically selected to avoid under- and overfitting, using early stopping. This is practically achieved by training the network until the performance starts to degrade, and subsequently adopting the model which achieved the best performance during the training process.

Finally, batch normalization is adopted to increase the stability of the neural network and reduce the required time for convergence. It also allows each layer of a network to learn by itself a little bit more independently of other layers. This is done by normalizing the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation.

4.4.3 Model Updater. This part of the module is **optional** and can operate only when the localization model becomes outdated due to

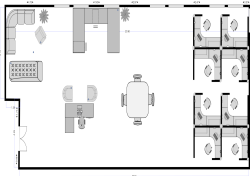


Figure 9: Office testbed.

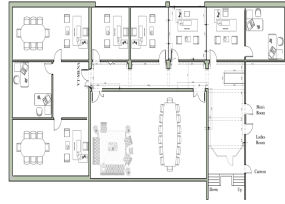


Figure 10: Floor testbed.

long-term variations, e.g. changes in the building structure, furniture placement, the available APs, etc. *LiPhi* does this in two steps 1) updating the fingerprint database 2) generating a new model. In such cases, we can temporarily add an the transportable LRSs to the area of interest for collecting a new fingerprint database by the systems' users and the labels are transparently assigned in the same way as discussed previously. This database is then used to train the deep model. In order to minimize training time and reduce the data needed as compared to the model creation time, transfer learning is leveraged. Specifically, *LiPhi* adopts fine-tuning in order to update the learned RSS-to-location mapping by learning a new mapping when the RSS level changes or the number of the considered APs decrease over time.

4.5 Online Phase

4.5.1 Location Predictor. After deploying the system, this phase aims to localize the user in real-time, using signals received from the nearby APs in the area of interest. The received RSSs of each scan are processed to form an input vector which contains an entry corresponding to each AP in the environment as mentioned previously. This vector is then fed to the trained localization model to estimate the current user location (x, y) .

5 EVALUATION

In this section, we evaluate the performance of *LiPhi* in two real-world indoor testbeds whose details are presented in Table 1. The first one (denoted as Office) is a big office of $240m^2$ area at a service building in our university (Figure 9). It is a cluttered indoor environment which contains desks, white boards and bookcases. The second one (denoted as Floor), shown in Figure 10, is a larger indoor testbed spanning a whole floor in another university campus with a $629m^2$ area containing several labs with different sizes and furniture placements, meeting room, offices as well as corridors.

Firstly, we describe how the data is collected and the software used. Next, we study the effect of the different system parameters on *LiPhi*'s accuracy. Finally, we compare the performance of *LiPhi* to three state-of-the-art localization systems.

5.1 Data Collection Setup and Tools

The data is collected with an Android application designed specially for this task. This application continuously scans for the nearby APs in the area of interest and records the information of each one including: the current time, the MAC address (ID) and the corresponding signal strength (i.e. timestamp, ID, RSS). The scanning rate is set to 1 scan per second. Even though we have a total of 136 and 52 APs detected in the Floor and the Office testbeds respectively; we use five and eight reference APs respectively since those

Table 1: Summary of the testbeds considered in evaluating *LiPhi*.

Criteria	Office	Floor
Area (m^2)	10×24	17×37
Number of virtual fingerprint points	128	576
Building Material	Brick	Brick & Wood
Total number of APs	52	136
Number of reference APs	5	8
Number of LRSs	4	6
Sampling rate (scan/sec)	1	1
LRS height (m)	1.4	1.4

Table 2: Default parameter values used in evaluation.

Parameter	Range	Default
Smoothing Window length	1-20	5
Virtual grid spacing (m)	0.1-5	1m
Number of DNN layers	1-100	7
Learning rate	0-1	0.001
Number of hidden units per layers	20-1000	100

are the ones with known a priori locations. LRS units are uniformly-distributed over the area of interest and they cover around six rooms of the Floor environment². Four LRSs are deployed along the periphery of the walls in the Office testbed. All LRSs are installed at the same height of 1.40m, and user traces can be detected as visualized in Figure 5.

Test points were collected on a uniform grid with a 1m spacing using the traditional fingerprinting approach **for evaluation only**. Note that *LiPhi* does not require any calibration or collection of data in the traditional fingerprinting manner to build the fingerprint database. The data was collected using several Android phones including Samsung Note 8, HTC One X9, Motorola Moto G5, among others. This is done with a view to capturing the device-variant characteristics of the WiFi measurements. Additionally, the test scans are collected on different months in order to show how the system will perform in the presence of environmental changes over time. We implemented our deep localization model using the Keras learning library which is a high-level neural network API running on top of the Google TensorFlow framework.

5.2 Effect of Changing *LiPhi* Parameters

In this section, we evaluate the effect of the different parameters and factors that affect *LiPhi* performance. In the following subsections, we show the effects of changing these parameters only on the Floor testbed for clarity of presentation. We report the optimal obtained parameters in Table 2. On the other hand, we report how *LiPhi* performs in both testbeds in Subsection 5.2.7.

5.2.1 Effect of Virtual Grid Spacing. Figure 11 shows the effect of changing the virtual cell spacing (i.e. reference point density) on the overall accuracy of the system and the corresponding time required by the *WiFi-based Trace Estimator* module to provide a location estimate. The run-time is calculated using a Lenovo Thinkpad X1 laptop running a 2.2GHz Intel i7-8750H processor with 64 GB RAM.

²This can be done incrementally with a single LRS unit only.

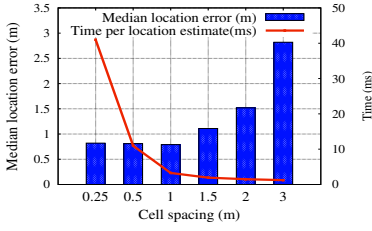


Figure 11: Effect of the grid cell spacing.

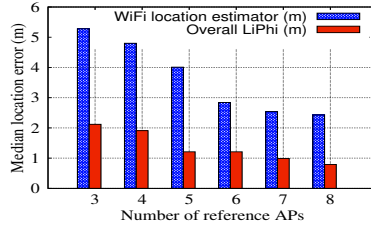


Figure 12: Effect of changing the number of reference APs.

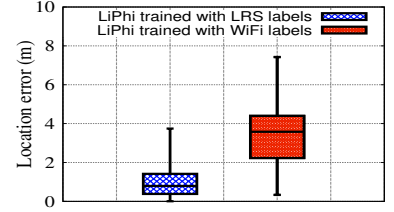


Figure 13: Effect of using LRS in labeling.

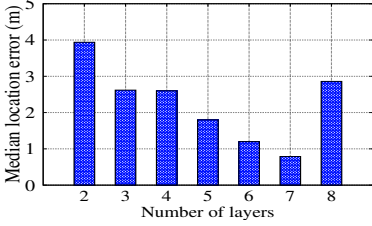


Figure 14: Effect of changing the number of layers on *LiPhi* accuracy.

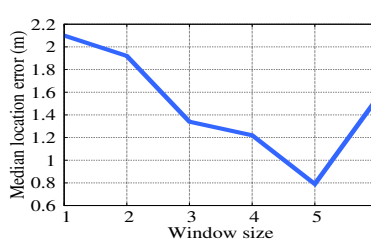


Figure 15: Effect of the smoothing window size on *LiPhi* accuracy.

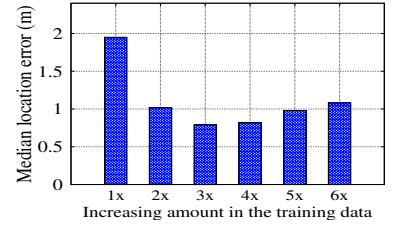


Figure 16: Effect of data augmentation on enhancing *LiPhi* accuracy.

The figure shows that, as expected, a smaller spacing between reference points yields high localization accuracy with a negligible delay in the calculation time. Nevertheless, this is performed only during the offline phase and *LiPhi* does not affect the real-time performance of the system. Additionally, the figure shows that a reference point spacing of up to 1m is enough to maintain the high accuracy of *LiPhi*.

5.2.2 Effect of Varying Density of Reference APs. Figure 12 shows the effect of changing the APs density on the system accuracy. For this, we uniformly and incrementally removed APs from the eight total reference APs present in the area of interest. The figure shows the accuracy of the WiFi trace estimator module degrades as the number of available APs is reduced. On the other hand, even with as low as five reference APs, *LiPhi* maintains a steady localization error of around 1m. This high accuracy with a relatively low number of APs can be explained as a result of two processes: Firstly, the location resetting using the accurate LRS trace estimator. Secondly, the used data augmentation techniques help the model to maintain its localization accuracy even with low AP densities. This highlights the robustness of *LiPhi*.

5.2.3 Effect of LRS-based Labeling. In this section, we study the influence of training the localization model of *LiPhi* using WiFi scans labeled by LRS as compared to labeling using the *WiFi-based Trace Estimator* module only. Figure 13 shows boxplots of the localization error of the system in both cases. The figure depicts that, as expected, leveraging LRS gives a drastic improvement in median error (351.8%), compared to the case of relying only on the coarse-grained WiFi-based labels. This can be attributed to the LRSs' refinement of the training data which significantly enhances the learning of the localization model and justifies the impact of using the place-and-play LRSs on the *LiPhi* system.

5.2.4 Number of Layers in the Network. Deep learning is designed to provide a hierarchical learning ability which can be achieved through cascading different layers. Therefore the number of layers of the deep network is one of the effective hyperparameters to boost the system performance. Figure 14 shows the effect of changing the number of layers on *LiPhi* accuracy. Empirically, the figure shows that increasing the number of layers increases the accuracy. This can be justified as the deeper models have more parameters and better learning ability. The figure also shows that, beyond an optimal value of 7 layers, the model tends to overfit the training data leading to a drop in accuracy.

5.2.5 WiFi Estimator Smoothing Window. Figure 15 shows the effect of varying the number of the WiFi scans utilized in estimating a location by the WiFi-based estimator module (Section 4.2). The figure shows that the more scans are fed to the module, the better localization accuracy until it reaches an optimal value at $v = 5$ beyond which a degradation occurs. This can be justified to two opposing factors. (1) Increasing v results in more information for location smoothing and outlier avoidance. (2) However, as v increases, more time is spent to collect these samples which may lead to locating the user in a preceding location (i.e. latency in response). A balance is achieved at a window size of 5 scans which leads to the best performance.

5.2.6 Effect of Data Augmentation. Figure 16 shows the effect of using data augmentation in increasing the amount of data and therefore the localization performance. The figure shows that data augmentation improves the *LiPhi* performance compared to augmentation-free training by 59%. The figure also confirms that the more training samples generated by the augmentation technique, the better the performance to cope with real world deployments by implicitly simulating the inherent variation of the noisy wireless channel.

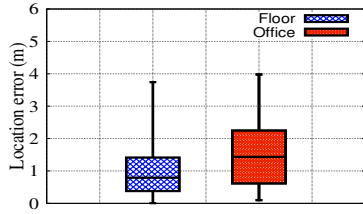


Figure 17: Performance of *LiPhi* in two different testbeds.

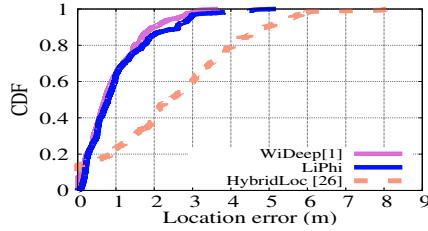


Figure 18: CDF of localization error with a fresh fingerprint.

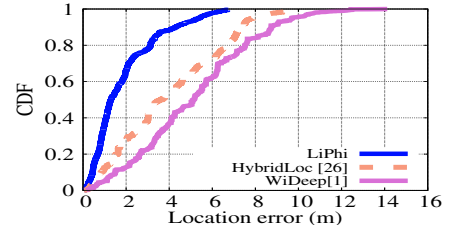


Figure 19: CDF of localization error after four months.

Table 3: Summary of the localization error percentiles of different techniques.

Mode	Technique	25 th Percentile	50 th Percentile	75 th Percentile	90 th Percentile
Fresh fingerprint	<i>LiPhi</i>	0.38m	0.79m	1.41m	2.62m
	HybridLoc [26]	1.15m (-203%)	2.41m (-205%)	3.82m (-171%)	4.89m (-87%)
	WiDeep [1]	0.38m (0%)	0.72m (9%)	1.40m (1%)	1.99m (24%)
Four months later	<i>LiPhi</i>	0.80m	1.29m	2.49m	4.28m
	HybridLoc [26]	1.77m (-121%)	3.63m (-181%)	6.23m (-150%)	7.35m (-72%)
	WiDeep [1]	3.21m (-301%)	5.12m (-297%)	7.12m (-186%)	8.75m (-104%)

Beyond using three times multiple of the original data, *LiPhi* performance tends to saturate then deteriorate as the noisy data becomes dominant relative to the non-synthetic/original samples.

5.2.7 Performance in Different Testbeds. In this section, we evaluate how the system would perform in two different testbeds: the Floor testbed and the Office testbed (Figure ??). The former is larger ($629m^2$) with many rooms; therefore, it requires a wider coverage by LRSs while the latter is smaller ($240m^2$) an open area without inner walls. In the Floor testbed, six LRSs are leveraged to cover the whole area of twelve rooms incrementally, and the Office testbed has four LRSs, covering the entire area of interest. Figure 17 shows that *LiPhi* obtains better performance in the Floor testbed as compared to the Office testbed. This can be justified due to two reasons: (1) The number of considered APs (input vector) in Floor testbed is more (136 and 52 on the Floor and the Office testbeds, respectively), which favors its performance as the model learns more information; (2) WiFi signatures are more location discriminative in the Floor testbed due to the presence of walls and the richer multi-path environment.

5.3 Comparative Evaluation

In this section, we compare the accuracy of *LiPhi* to one traditional fingerprinting technique which builds a denoising autoencoder for localization (WiDeep [1]), and another probabilistic WiFi-based localization technique that automatically constructs the fingerprint database using labels obtained from BLE-devices (iBeacons), HybridLoc [26]. For fair comparison, all techniques are deployed in the same environment and trained on the same data (i.e. RSS data) collected from a total of 136 APs. Additionally, a default of 10 iBeacons are installed in the environment which are required for HybridLoc’s operation, as reported in [26]. Figure 18 and Figure 19 show the CDF of distance error for the three techniques with temporal variation. Figure 18 shows the accuracy of the three systems when

tested with a fresh fingerprint. Specifically, *LiPhi* and WiDeep [1] are matched in performance even though WiDeep [1] is trained with manual fingerprinted data while *LiPhi* is trained with automatically constructed fingerprints. This can be attributed due to the accurate labeling obtained from the LRS-based trace estimator module. HybridLoc [26] obtains the lowest performance as BLE-based tracking provides coarse-grained labeling which is not enough for annotating WiFi scans. Additionally, it uses a probabilistic method which do not benefit from the deep learning methods advantages.

Figure 19 shows how all systems would perform when tested four months later. The figure illustrates that our *LiPhi* system achieves improvements in localization performance by 297% and 181%, under this condition, as compared to WiDeep and HybridLoc; respectively. This can be explained by the combination of the data augmentation methods (random dropping and additive noise) in the training data and the adoption of different regularization techniques which gives *LiPhi* greater flexibility and generalization ability than the other systems. Additionally, *LiPhi* and HybridLoc [26] obtained better performance as compared to WiDeep [1] as they have provision to update the fingerprint database and the localization model. Despite the high overhead spent to collect data for WiDeep [1] as a traditional fingerprinting technique, its accuracy cannot be maintained without re-doing the arduous calibration process. In summary - as shown in Table 3 - *LiPhi* is robust to variation over time surpassing the other techniques. This highlights the promise of *LiPhi* in enabling an accurate localization models with zero calibration overhead.

6 RELATED WORK

In this section, we discuss the most relevant literature.

6.1 Fingerprinting Systems

Fingerprinting systems [4, 37] present the most popular localization technique due to their high accuracy. In particular, the system in [4]

employs deterministic matching using K-nearest neighbor (KNN), so that the unknown user location is assigned to the fingerprint location closest to the average RSS signature of that location. However, deterministic techniques cannot handle the inherent noise and variations in the WiFi signal. On the other hand, probabilistic techniques such as [37] have better adaptability to noise as noise is usually modeled as an uncertainty phenomenon. In this case, the recorded fingerprints are the RSS histogram of each AP at each reference location and the user location is estimated based on Bayesian inference. In these techniques, the signals from different APs are considered to be independent to avoid the curse of dimensionality problem. This leads to a loss of useful information which leads to coarse-grained localization accuracy.

On the other hand, cameras are used to improve WiFi fingerprinting-based indoor positioning in [14]. However, this solution usually requires complicated calibration process to adjust for camera scaling and perspective and may necessitate the presence of many permanently-installed cameras to cover the whole area. Unlike camera-based solutions, a LiDAR-based solution is transportable. So it can seamlessly be placed and work without any tedious calibration. Additionally, camera-based solutions are not suitable for environments where privacy is at a premium, which is not the case in LiDARs.

Recently, different deep learning-based localization systems, e.g. [1, 17, 18, 20, 22–24, 33, 34] have shown better localization performance due to their ability to learn complex patterns and automatically extract discriminative features. Several deep learning architectures have been proposed in indoor positioning including: Restricted Boltzman Machines in DeepFi [33], a deep convolutional neural network for CSI-based localization in [34] and stacked denoising autoencoders for each fingerprint reference point in [1]. The commonality between these techniques is that they depend on traditional fingerprinting and do not have provisions to reduce the data collection overhead. This is a major problem in deep learning-based systems as they require large amounts of data to be properly trained, which directly translates to extra fingerprinting overhead in order to satisfy this requirement.

In contrast, LiPhi builds a deep-learning based localization model relying on a fingerprint database transparently constructed without explicit user participation. Additionally, LiPhi has provisions to boost the model robustness to noise.

6.2 Crowdsourcing Systems

Another line of research is proposed to mitigate the calibration overhead required for constructing fingerprint databases using crowdsourcing. This can be done explicitly without user intervention [13] or implicitly with user intervention [2, 15, 31]. The system in [13] increases the fingerprint coverage by periodically asking the user to provide her current location. Although in theory this method can provide accurate fingerprints, it is annoying to users is not a practical solution. Hence, the systems in [2, 15, 31] implicitly estimate the user location coarsely using dead-reckoning based on the users' smartphones' inertial sensors. The corresponding WiFi scan is then associated with a fingerprint. Thereafter, the location estimation can be opportunistically refined using sensor-based landmarks or map-matching. However, inertial sensors in smartphones are noisy,

leading to an increasing error over time and missed opportunities to correct estimations. To avoid the noisy inertial sensors, the system in [26] proposes a method to tag WiFi scans with locations obtained from BLE-enabled high-end smart phones. Therefore, the method requires the area to be well covered with BLE beacons (e.g. iBeacons) which should be sensed by high-end phones to estimate the user location and therefore build a WiFi fingerprint database. While this method is feasible, its application requires the site surveyors to be equipped with high-end phones and the area to be well covered by BLE beacons. Therefore, it cannot be considered a ubiquitous solution for every environment.

LiPhi, on the contrary, requires neither user intervention nor high-end devices with permanently installed devices. It only uses temporarily installed LRSs for constructing the fingerprint database.

6.3 LRS-based Systems

Several systems have been proposed leveraging LRSs in many indoor applications. In [5], an indoor navigational system for a robot is built based on WiFi for localization in addition to an embedded LRS for enhancing the position estimates and avoiding obstacles. *LiPhi*, On the contrary, does not assume that every user's phone is equipped with LRSs. Another research direction aims to track pedestrians in indoor environments based on LRS as proposed in [8, 30]. Although LiDAR is a promising technology for accurate user tracking, it cannot identify the tracked person. To handle this issue, the system in [29] leverages mobile phone inertial sensors to estimate the user trajectory using dead-reckoning (DR) approach. Then, the system matches the LRS-based trajectory to the DR-based trajectory in order to identify the user. However, depending on the noisy on-board inertial sensors in the consumers' phones leads to a large position errors and random estimated trajectories which cannot be easily matched. Additionally, the system leverages LRSs for tracking purposes and requires inertial sensors for identification purposes, the latter which exist only in high-end smart phones. This therefore requires extreme deployment expenses and limits its ubiquitous adoption.

In contrast, LiPhi does not requires noisy inertial sensors nor permanently deployed LRSs. LiPhi only uses LRSs temporarily during fingerprint database construction/maintenance, leading to extreme saving in expenses. Additionally, it provides an accurate WiFi localization system with similar accuracy to traditional (i.e manual) fingerprinting techniques with virtually zero data collection overhead.

7 CONCLUSION

We presented *LiPhi*, a robust system that can automatically construct (and update) a fingerprint database for indoor localization systems without requiring traditional data collection overhead. The system leverages crowd-sourced WiFi signals to roughly estimate the user location, whose accuracy is then improved using a LRS-derived estimate. The resulting fine-grained location estimates are used to tag WiFi scans which are then used to train a deep localization model. We also presented data augmentation techniques to improve the localization model's robustness against temporal signal variations.

We evaluated *LiPhi*, in two different challenging environments that represent a full floor and a large Office in the university, using

different Android devices. The results show that *LiPhi* provides a localization accuracy better than the state-of-the-art crowdsourcing-based and fingerprinting-based systems by 297% and 181%, when tested with data collected a few months later; respectively.

ACKNOWLEDGMENT

Part of this work was carried out in part under the Society 5.0 Realization Research Support Project funded by Ministry of Education and the Cooperative Research Project Program of RIEC, Tohoku University.

REFERENCES

- [1] Moustafa Abbas, Moustafa Elhamshary, Hamada Rizk, Marwan Torki, and Moustafa Youssef. 2019. WiDeep: WiFi-based Accurate and Robust Indoor Localization System using Deep Learning. In *Proceedings of the International Conference on Pervasive Computing and Communications (PerCom)*. IEEE.
- [2] Heba Abdelnasser, Reham Mohamed, Ahmed Elgohary, Moustafa Farid Alzantot, He Wang, Souvik Sen, Romit Roy Choudhury, and Moustafa Youssef. 2016. SemanticSLAM: Using environment landmarks for unsupervised indoor localization. *IEEE Transactions on Mobile Computing* 15, 7 (2016), 1770–1782.
- [3] Moustafa Alzantot and Moustafa Youssef. 2012. CrowdInside: automatic construction of indoor floorplans. In *the 20th International Conference on Advances in Geographic Information Systems*. ACM, 99–108.
- [4] Paramvir Bahl and Venkata N Padmanabhan. 2000. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of the Nineteenth International Conference on Computer and Communications Societies*, Vol. 2. IEEE, 775–784.
- [5] Joydeep Biswas and Manuela Veloso. 2010. Wifi localization and navigation for autonomous indoor mobile robots. In *Proceedings of the International conference on robotics and automation*. IEEE, 4379–4384.
- [6] Rizanne Elbakly and Moustafa Youssef. 2016. A robust zero-calibration RF-based localization system for realistic environments. In *Proceedings of the 13th International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 1–9.
- [7] Ahmed Eleryan, Mohamed Elsabagh, and Moustafa Youssef. 2011. Synthetic generation of radio maps for device-free passive localization. In *Proceedings of the International conference on Global Telecommunications Conference (GLOBECOM)*. IEEE, 1–5.
- [8] Ajo Fod, Andrew Howard, and MAJ Mataric. 2002. A laser-based people tracker. In *Proceedings of the International Conference on Robotics and Automation*, Vol. 3. IEEE, 3024–3029.
- [9] Yiming Ji, Saad Biaz, Santosh Pandey, and Prathima Agrawal. 2006. ARIADNE: a dynamic indoor signal map construction and localization system. In *Mobile systems, applications and services*. ACM, 151–164.
- [10] Christos Laoudias, Paul Kemppi, and Christos G Panayiotou. 2009. Localization using radial basis function networks and signal strength fingerprints in WLAN. In *Proceedings of the International Conference on Global Communications*. IEEE, 1–6.
- [11] Noureddine Lasla, Mohamed F Younis, Abdelraouf Ouadjaout, and Nadjib Badache. 2015. An effective area-based localization algorithm for wireless networks. *IEEE Trans. Comput.* 64, 8 (2015), 2103–2118.
- [12] Jun-geun Park, Ben Charrow, Dorothy Curtis, Jonathan Battat, Einat Minkov, Jamey Hicks, Seth Teller, and Jonathan Ledlie. 2010. Growing an Organic Indoor Location System. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*. Association for Computing Machinery, New York, NY, USA, 271–284. <https://doi.org/10.1145/1814433.1814461>
- [13] Jun-geun Park, Ben Charrow, Dorothy Curtis, Jonathan Battat, Einat Minkov, Jamey Hicks, Seth Teller, and Jonathan Ledlie. 2010. Growing an organic indoor location system. In *the 8th international conference on Mobile systems, applications, and services*. ACM, 271–284.
- [14] Laura Radaelli, Yael Moses, and Christian S Jensen. 2014. Using cameras to improve wi-fi based indoor positioning. In *International symposium on web and wireless geographical information systems*. Springer, 166–183.
- [15] Anshul Rai, Krishna Kant Chintalapudi, Venkata N Padmanabhan, and Rijurekha Sen. 2012. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 293–304.
- [16] Hamada Rizk. 2019. Device-Invariant Cellular-Based Indoor Localization System Using Deep Learning. In *The ACM MobiSys 2019 on Rising Stars Forum*. ACM, 19–23.
- [17] Hamada Rizk. 2019. SoloCell: Efficient Indoor Localization Based on Limited Cell Network Information And Minimal Fingerprinting. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 604–605.
- [18] Hamada Rizk, Moustafa Abbas, and Moustafa Youssef. 2020. OmniCells: Cross-Device Cellular-based Indoor Location Tracking Using Deep Neural Networks. In *18th Annual IEEE Conference on Pervasive Computing and Communications (PerCom)*. IEEE.
- [19] Hamada Rizk, Sherin Elgokhy, and Amany Sarhan. 2015. A hybrid outlier detection algorithm based on partitioning clustering and density measures. In *the 10th International Conference on Computer Engineering & Systems (ICCES)*. IEEE, 175–181.
- [20] Hamada Rizk, Ahmed Shokry, and Moustafa Youssef. 2019. Effectiveness of Data Augmentation in Cellular-based Localization Using Deep Learning. In *Proceedings of the International Conference on Wireless Communications and Networking (WCNC)*. IEEE.
- [21] Hamada Rizk, Marwan Torki, and Moustafa Youssef. 2018. CellinDeep: Robust and Accurate Cellular-based Indoor Localization via Deep Learning. *IEEE Sensors Journal* (2018).
- [22] Hamada Rizk, Hirozumi Yamaguchi, Teruo Higashino, and Moustafa Youssef. 2020. A Ubiquitous and Accurate Floor Estimation System Using Deep Representational Learning. In *Proceedings of the 28th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM.
- [23] Hamada Rizk and Moustafa Youssef. 2019. Increasing Coverage of Indoor Localization Systems for IEEE112 Support. In *2nd MENA Regional International Telecommunication Society Conference (ITS)*.
- [24] Hamada Rizk and Moustafa Youssef. 2019. MonoDCell: A Ubiquitous and Low-Overhead Deep Learning-based Indoor Localization with Limited Cellular Information. In *Proceedings of the 27th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM.
- [25] Jang-Ping Sheu, Pei-Chun Chen, and Chih-Shun Hsu. 2008. A distributed localization scheme for wireless sensor networks with improved grid-scan and vector-based refinement. *IEEE transactions on mobile computing* 7, 9 (2008), 1110–1123.
- [26] Ahmed Shokry, Moustafa Elhamshary, and Moustafa Youssef. 2017. The Tale of Two Localization Technologies: Enabling Accurate Low-Overhead WiFi-based Localization for Low-end Phones. In *the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM.
- [27] Biljana Risteska Stojkoska, Jordan Palikrusher, Kire Trivodaliev, and Slobodan Kalajdziski. 2017. Indoor localization of unmanned aerial vehicles based on RSSI. In *Proceedings of the International Conference on Smart Technologies*. IEEE, 120–125.
- [28] Masashi Sugiyama and Motoaki Kawanabe. 2012. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press.
- [29] Takumi Takafuji, Kazuhisa Fujita, Takamasa Higuchi, Akihito Hiromori, Hirozumi Yamaguchi, and Teruo Higashino. 2014. Indoor localization utilizing tracking scanners and motion sensors. In *Proceedings of the 11th IEEE International Conference on Ubiquitous Intelligence and Computing*. IEEE, 112–119.
- [30] Yusuke Wada, Takamasa Higuchi, Hirozumi Yamaguchi, and Teruo Higashino. 2013. Accurate positioning of mobile phones in a crowd using laser range scanners. In *Proceedings of the 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 430–435.
- [31] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. 2012. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 197–210.
- [32] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. 2015. DeepFi: Deep learning for indoor fingerprinting using channel state information. In *Proceedings of the International Conference on Wireless Communications and Networking*. IEEE, 1666–1671.
- [33] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. 2017. CSI-based fingerprinting for indoor localization: A deep learning approach. *IEEE Transactions on Vehicular Technology* 66, 1 (2017), 763–776.
- [34] Xuyu Wang, Xiangyu Wang, and Shiwen Mao. 2017. CiFi: Deep convolutional neural networks for indoor localization with 5 GHz Wi-Fi. In *Proceedings of the International Conference on Communications (ICC)*. IEEE, 1–6.
- [35] Hirozumi Yamaguchi, Akihito Hiromori, and Teruo Higashino. 2018. A human tracking and sensing platform for enabling smart city applications. In *Proceedings of the International Conference on Distributed Computing and Networking*. ACM.
- [36] Moustafa Youssef. 2015. Towards truly ubiquitous indoor localization on a world-wide scale. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 12.
- [37] Moustafa Youssef and Ashok Agrawala. 2005. The Horus WLAN location determination system. In *the 3rd international conference on Mobile systems, applications, and services*. ACM, 205–218.
- [38] Liye Zhang, Lin Ma, Yubin Xu, and Cheng Li. 2017. Linear regression algorithm against device diversity for indoor WLAN localization system. In *Proceedings of the International Global Communications Conference*. IEEE, 1–6.
- [39] Wei Zhang, Kan Liu, Weidong Zhang, Youmei Zhang, and Jason Gu. 2016. Deep neural networks for wireless localization in indoor and outdoor environments. *Neurocomputing* 194 (2016), 279–287.