CSI Phase Fingerprinting for Indoor Localization With a Deep Learning Approach

Xuyu Wang, *Student Member, IEEE*, Lingjun Gao, *Student Member, IEEE*, and Shiwen Mao, *Senior Member, IEEE*

***Abstract*—With the increasing demand of location-based services, indoor localization based on fingerprinting has become an increasingly important technique due to its high accuracy and low hardware requirement. In this paper, we propose PhaseFi, a fingerprinting system for indoor localization with calibrated channel state information (CSI) phase information. In PhaseFi, the raw phase information is first extracted from the multiple antennas and multiple subcarriers of the IEEE 802.11n network interface card by accessing the modified device driver. Then a linear transformation is applied to extract the calibrated phase information, which we prove to have a bounded variance. For the offline stage, we design a deep network with three hidden layers to train the calibrated phase data, and employ the weights of the deep network to represent fingerprints. A greedy learning algorithm is incorporated to train the weights layer-by-layer to reduce computational complexity, where a subnetwork between two consecutive layers forms a restricted Boltzmann machine. In the online stage, we use a probabilistic method based on the radial basis function for online location estimation. The proposed PhaseFi scheme is implemented and validated with extensive experiments in two representation indoor environments. It is shown to outperform three benchmark schemes based on CSI or received signal strength in both scenarios.**

***Index Terms*—Channel state information (CSI), deep learning, fingerprinting, indoor localization, phase calibration.**

I. Introduction

THE proliferation of mobile terminals such as smartphones, tablets, and laptops has stimulated enormous interests in indoor localization and location-based services [1]–[3]. As one of the popular schemes for indoor localization, a fingerprinting-based approach first establishes a database with thorough measurements of the field and then infers the real-time location by comparing the new measurements with database data. It requires no additional infrastructure support and is thus amenable for indoor deployment.

Manuscript received January 15, 2016; revised March 18, 2016 and April 5, 2016; accepted April 24, 2016. Date of publication April 27, 2016; date of current version January 10, 2017. This work was supported in part by the U.S. National Science Foundation under Grant CNS-1247955 and in part by the Wireless Engineering Research and Education Center at Auburn University. This work was presented in part at the IEEE GLOBECOM 2015, San Diego, CA, USA.

X. Wang and S. Mao are with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201 USA (e-mail: xzw0029@tigermail.auburn.edu; smao@ieee.org).

L. Gao was with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201 USA. He is now with DataYes Inc., Shanghai 200122, China (e-mail: lzg0014@tigermail.auburn.edu).

Digital Object Identifier 10.1109/JIOT.2016.2558659

In the training stage, a database is established by collecting and preprocessing survey data for training positions in the target area. Traditional fingerprinting schemes store all samples of survey data in the database. Such an approach may lead to low efficiency and localization accuracy since the survey data is usually noisy. Machine learning methods, such as K-nearest-neighbor [4], neural networks [5], and support vector machines [5], have been adopted to extract and store the main features (or learning parameters) of survey data instead, and can thus achieve better localization performance. Although these machine learning methods can make better use of survey data than traditional techniques, they belong to the class of shallow learning methods with their limit on fully exploiting the survey data.

In the online testing stage, a mobile terminal collects test data and infers its location by comparing the test data with that stored in the database. The location inference method can be either deterministic or probabilistic. With deterministic methods, the location of the mobile device is estimated by searching each training point in the database to find the most closely matched one with the test data. Alternatively, a probabilistic method is to identify several close points with a maximum likelihood (ML) probability, and to calculate the estimated location as a weighted average of the candidate locations. The probabilistic approach can usually achieve better location accuracy than deterministic methods [6].

Many existing indoor fingerprinting systems utilize WiFi received signal strength (RSS) values as fingerprints due to the simplicity and low hardware requirements. For example, Radar is the first fingerprinting system based on RSS with a deterministic method for location estimation [7]. Later, Horus utilizes a probabilistic method for indoor localization with RSS values [6], which achieves better localization accuracy than Radar. Such RSS-based methods have two main disadvantages. First, RSS values are highly random and its correlation with propagation distance is loose due to shadowing fading and multipath effects. Second, RSS values are coarse information obtained by averaging the amplitudes of all incoming signals, and the rich channel information from different subcarriers is not used. Thus, localization based on RSS values may lead to poor localization performance.

By modifying the device driver, we can now obtain channel state information (CSI) from some advanced WiFi network interface cards (NICs), such as the Intel WiFi link (IWL) 5300 NIC [8], [9]. CSI values provide subcarrier-level channel measurements, which can be helpful for indoor fingerprinting.

For example, FIFS [10] utilizes the weighted average CSI values over multiple antennas to improve the performance of RSS-based method for indoor fingerprinting. Another work, DeepFi [11] learns a large amount of CSI amplitude data from three antennas for indoor localization based on a deep network. These schemes only consider the amplitude of CSI, and the CSI phase information is ignored, which is largely due to the randomness and unavailability of the raw phase information. To the best of our knowledge, CSI-MIMO [12] incorporates both magnitude and phase information of CSI from each subcarrier for fingerprinting, but the phase information is not calibrated. In fact, the calibrated phase information obtained with a linear transformation is successfully used for line-of-sight (LOS) identification with WiFi [13] and passive human movement detection [14]. These two interesting works motivate us to explore calibrated CSI phase information for indoor fingerprinting.

In this paper, we present PhaseFi, an indoor fingerprinting system based on calibrated phase information of CSI. In PhaseFi, the raw phase information is first extracted from the CSI values from the 30 subcarriers of each of the three antennas of the IWL 5300 NIC (i.e., 90 in total), by accessing the modified device driver. Then, by implementing a linear transformation to remove the phase offset, we obtain the calibrated phase information, which is shown in our measurement study to be considerably more accurate than raw phases. We also provide a phase calibration algorithm and prove an upper bound on the variance of the calibrated phase, which clearly indicates its stability feature.

In the offline stage, unlike traditional shallow learning methods, we design a deep network with three hidden layers to train the calibrated phase data, and use weights to represent fingerprints, which can fully exploit the characteristic of the calibrated phase data. We also develop a greedy learning algorithm to train the weights in a layer-by-layer manner to effectively reduce the computational complexity. With this training approach, a subnetwork between two consecutive layers forms a restricted Boltzmann machine (RBM), which is solved by a contrastive divergence with one step iteration (CD-1) algorithm for suboptimal solutions. Once the fingerprint database is established, the online stage uses a Bayes method based on the radial basis function (RBF) for location estimation.

We implement the PhaseFi system with a laptop computer and an access point (AP), and conduct extensive experiments to validate the performance of the PhaseFi system under two representative indoor environments, including a living room in a house and a computer laboratory that is cluttered with metal tables and computers. We find that PhaseFi outperforms three benchmark schemes that are either based on CSI or RSS in both scenarios.

In summary, the main contributions in this paper include the following.

1) We propose to use CSI phase information for indoor fingerprinting. Specifically, we theoretically prove and experimentally validate the feasibility of utilizing the calibrated CSI phase information for indoor localization. To the best of our knowledge, this is the first work to leverage the calibrated CSI phase information for indoor fingerprinting.
2) We design a deep network with three hidden layers to train the calibrated phase data, and utilize the weights of the deep network to represent fingerprints. We also develop a greedy learning algorithm to effectively reduce the computational overhead for training. Furthermore, we present a Bayes method based on RBF for probabilistic location estimation.
3) We implement the PhaseFi system with commodity WiFi device and demonstrate its performance in two representative indoor environments. Experimental results show that PhaseFi outperforms several existing RSSI and CSI-based schemes at only slightly increased execution time. PhaseFi satisfies the real-time localization requirement for indoor localization.

This paper is organized as follows. The preliminaries and phase sanitization are introduced in Section II. We present PhaseFi in Section III and our experimental study in Section IV. Section VI concludes this paper.

II. Preliminaries and Phase Sanitization

A. CSI

In modern digital wireless communication systems, orthogonal frequency division multiplexing (OFDM) is widely used (e.g., in WiFi standards such as the IEEE 802.11a/g/n) to combat frequency selective fading in multipath propagation environments. As shown in Fig. 1, at the OFDM transmitter, data is encoded and mapped into multiple orthogonal subcarriers and then transmitted over the subcarriers. With inverse fast Fourier transform (IFFT), the subcarriers are converted from the frequency domain to the time domain. To reduce the intersymbol interference (ISI), the cyclic prefix (CP) is added in the time domain. Then, in-phase and quadrature (I–Q) modulation is used for transmission in the multipath channel. The digital data is converted into analog data with the digital to analog converter (DAC). Finally, the analog signal is up-converted and amplified by the high power amplifier (HPA). At the OFDM receiver, the signal is down-converted to the baseband. The automatic gain controller (AGC) can compensate for the signal amplitude attenuation. The inverse process of that at the transmitter is implemented for recovering the data at the receiver.

By modifying the device driver of off-the-shelf NICs, i.e., Intel's IWL 5300, we are able to obtain CSI as fine-grained physical layer (PHY) information, which represents the subcarrier-level channel measurements. In addition, CSI describes the channel properties experienced by the packet. For example, a wireless signal in propagation may undergo considerable impairments due to shadowing, multipath propagation, and distortion, which are reflected in the CSI.

The WiFi channel at the 2.4 GHz band can be considered as a narrowband flat fading channel for OFDM systems. The channel model is defined as

$$\vec{Y} = \mathrm{CSI} \cdot \vec{X} + \vec{N} \tag{1}$$

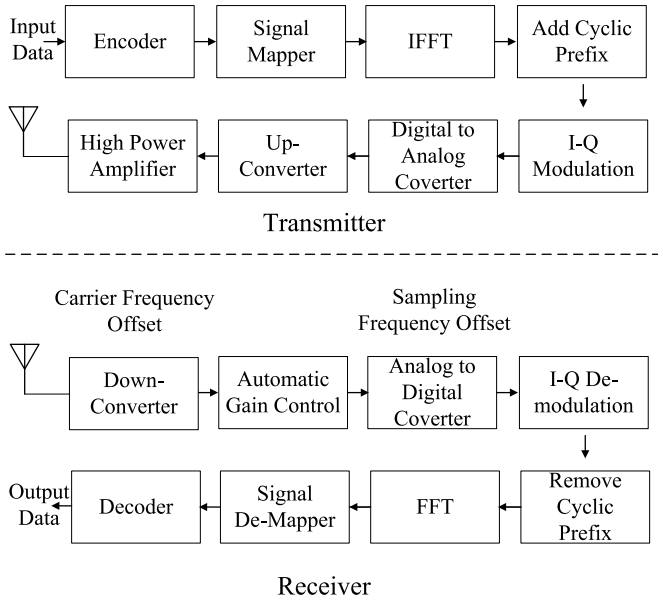


Fig. 1. Block diagram of an OFDM transceiver.

where $\vec{Y}$ and $\vec{X}$ denote the received and transmitted signal vectors, respectively, $\vec{N}$ is the additive white Gaussian noise, and CSI represents the channel's frequency response.

Although an 802.11a/g/n receiver implements an OFDM system with 56 subcarriers, the IWL 5300 NIC exports 30 out of the 56 subcarriers via the device driver for each of its three antennas. The channel frequency response CSI_i of subcarrier i is a complex value, as follows:

$$\text{CSI}_i = |\text{CSI}_i| \exp\{j\angle \text{CSI}_i\} \tag{2}$$

where $|\text{CSI}_i|$ and $\angle \text{CSI}_i$ are the amplitude response and the phase response of subcarrier i, respectively. In this paper, the proposed PhaseFi framework is based on the phases of the 30 subcarriers in the OFDM system, as discussed in the following.

B. Phase Sanitization

Although the phase of CSI is available from the IWL 5300 NIC, they have not been exploited for indoor localization yet. The problem is mainly due to the hardware imperfection, which leads to measured phase errors. In fact, there are two main causes for the above errors for the system in Fig. 1. The first one is carrier frequency offset (CFO) generated by the down-converter for receiver signal, because the central frequencies between the receiver and the transmitter cannot be perfectly synchronized. The other one is the sampling frequency offset (SFO) generated by the ADC, because of nonsynchronized clocks. Moreover, for SFO, the measured phase errors are different for different subcarriers. Thus, the raw phase information is of limited use for indoor localization.

In this paper, we propose a simple yet effective approach to mitigate the random phase offsets by implementing a linear transformation. Let $\angle\widehat{\text{CSI}}_i$ denote the measured phase of subcarrier i. It can be written as

$$\angle\widehat{\text{CSI}}_i = \angle \text{CSI}_i + 2\pi \frac{m_i}{N} \Delta t + \beta + Z \tag{3}$$

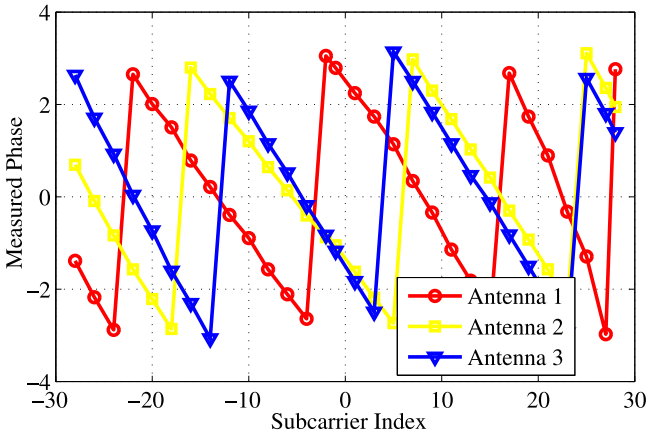


Fig. 2. Measured phase values for three different antennas.

where $\angle \text{CSI}_i$ is the genuine phase, Δt is the time lag due to SFO, m_i is the subcarrier index of the ith subcarrier, N is the fast Fourier transform (FFT) size, β is the unknown phase offset due to CFO, and Z is the measurement noise. We can obtain the subcarrier indices m_i for $i = 1$ to 30, and the FFT size N from the IEEE 802.11n specification [9]. In fact, because of the unknown Δt and β, it is impossible to obtain the genuine phase information. However, considering the phase across the total frequency band, we can implement a linear transformation on the raw phases to remove the Δt and β terms [14].

Let k and b denote the slope of phase and the offset across the entire frequency band, respectively. It is noticed that the phase error $2\pi(m_i/N)\Delta t + \beta$ is a linear function of the subcarrier index m_i. We can estimate the slope of phase k and the offset b with the following expressions:

$$k = \frac{\angle\widehat{\text{CSI}}_{30} - \angle\widehat{\text{CSI}}_1}{m_{30} - m_1} \tag{4}$$

$$b = \frac{1}{30}\sum_{i=1}^{30} \angle\widehat{\text{CSI}}_i. \tag{5}$$

Subtracting $km_i + b$ from the raw phase $\angle\widehat{\text{CSI}}_i$, we can obtain the calibrated phase $\angle\widetilde{\text{CSI}}_i$, which is given by

$$\angle\widetilde{\text{CSI}}_i = \angle\widehat{\text{CSI}}_i - km_i - b. \tag{6}$$

Although the above expression (6) can be used for calibrating phase information, the measured phase is folded due to the recurrence characteristic of phase. Thus, we need to transform the measured phase into the true value. In Fig. 2, we plot the measured phase values of CSI for the three antennas at the receiver. It is noticed that the measured phase of each of the three antennas is folded with the increase of subcarrier index and the range of the phase is $[-\pi\ \pi]$. In order to obtain the true measured phase, the folded phase can be recovered by subtracting multiple 2π. Thus, we propose a new phase calibration algorithm in Algorithm 1. In lines 8–13 of the algorithm, the measured phase is compensated for multiple 2π's by judging whether the measured phase change between the adjacent subcarriers is greater than the given threshold such

Algorithm 1: Phase Calibration

1 **Input:** measured phase values M_P of 30 subcarriers;
2 **Output:** calibrated phase values C_P of 30 subcarriers;
3 Set T_P as a vector as the same size of M_P;
4 Set m as a vector from -28 to 28;
5 Set diff $= 0$;
6 Set $\eta = \pi$;
7 Set $T_P(1)$=$M_P(1)$;
8 **for** $i = 2 : 30$ **do**
9 | **if** $M_P(i) - M_P(i-1) > \eta$ **then**
10 | | diff $=$ diff $+ 1$;
11 | **end**
12 | $T_P(i) = M_P(i) -$ diff $* 2 * \pi$;
13 **end**
14 Compute $k = \frac{T_P(30)-T_P(1)}{m(30)-m(1)}$;
15 Compute $b =$ sum$\{T_P\}/30$;
16 **for** $i = 1 : 30$ **do**
17 | $C_P(i) = T_P(i) - k * m(i) - b$;
18 **end**

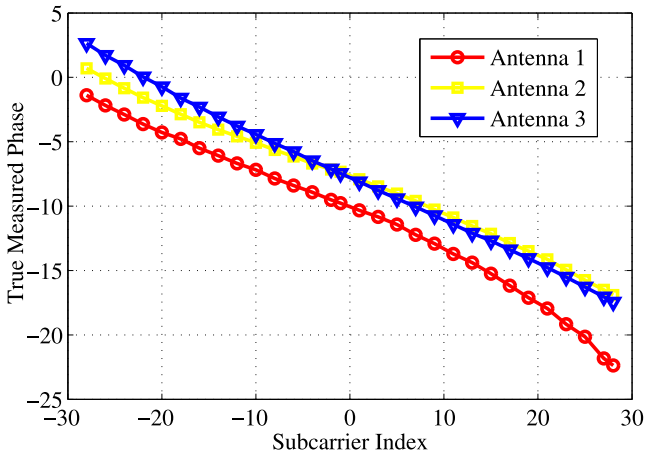


Fig. 3. True measured phase values for three different antennas.

as π. In lines 14–18, the calibrated phase is obtained based on the above phase calibration analysis.

Fig. 3 presents the true measured phase values for three different antennas. We can see that with the increase of subcarrier index, the true measured phase gradually decreases for all the three different antennas. Fig. 4 shows the calibrated phase values for three different antennas. It is noticed that the range of the calibrated phase becomes much smaller than the measured phase for three antennas. On the other hand, we present an upper bound on the variance of the calibrated phase in the following theorem.

Theorem 1: When the indices of 30 subcarriers are symmetric (i.e., ranging from -28 to 28 as in IEEE 802.11n) and the true phases of the 30 subcarriers are independent identically distributed, an upper bound of the variance of the calibrated phase is given by

$$\mathrm{Var}\big(\angle\widetilde{\mathrm{CSI}}_i\big) \le \frac{23}{15}\mathrm{Var}(\angle\mathrm{CSI}_i). \tag{7}$$

Proof: According to (3), we can compute the slope of the phase $k = ((\angle\mathrm{CSI}_{30} - \angle\mathrm{CSI}_1)/m_{30} - m_1) + (2\pi/N)\Delta t$,

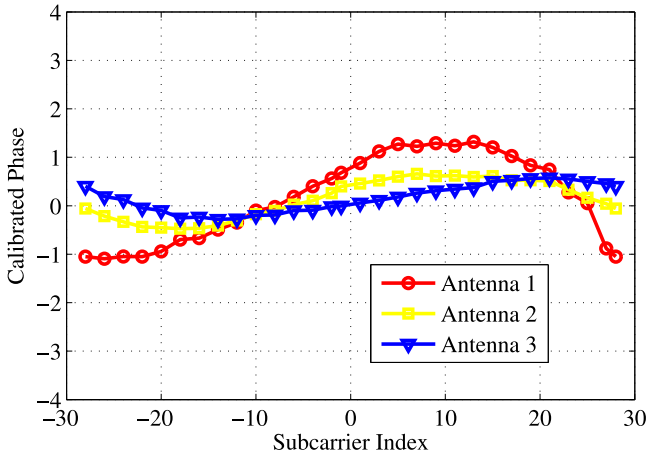


Fig. 4. Calibrated phase values for three different antennas.

and the offset across the total frequency band $b = (1/30)\sum_{i=1}^{30}\angle\mathrm{CSI}_i + (2\pi\Delta t/30N)\sum_{i=1}^{30} m_i + \beta + Z$. Since the indices of the 30 subcarriers are symmetric for IEEE 802.11n [14], we have $\sum_{i=1}^{30} m_i = 0$. It follows that $b = (1/30)\sum_{i=1}^{30}\angle\mathrm{CSI}_i + \beta + Z$. Substituting the slope of the phase, k, the offset, b, and the measured phase of subcarrier i, $\angle\widehat{\mathrm{CSI}}_i$, into (6), the calibrated phase is given by

$$\angle\widetilde{\mathrm{CSI}}_i = \angle\mathrm{CSI}_i - \frac{\angle\mathrm{CSI}_{30} - \angle\mathrm{CSI}_1}{m_{30} - m_1}m_i - \frac{1}{30}\sum_{i=1}^{30}\angle\mathrm{CSI}_i.$$

Note that the calibrated phase is a linear combination of the true phases, with the random offset β and time lag Δt removed. Since the true phases of the 30 subcarriers are independent identically distributed, the variance of the calibrated phase is $\mathrm{Var}(\angle\widetilde{\mathrm{CSI}}_i) = \mathrm{Var}(\angle\mathrm{CSI}_i) + (m_i^2/((m_{30} - m_1)^2))(\mathrm{Var}(\angle\mathrm{CSI}_{30}) + \mathrm{Var}(\angle\mathrm{CSI}_1)) + \mathrm{Var}((1/30)\sum_{i=1}^{30}\angle\mathrm{CSI}_i)$. Since the subcarrier indices are symmetric, we have $m_i \le m_{30}$ and $m_{30} = -m_1$, and it follows that $(m_i^2/(m_{30} - m_1)^2) \le (m_{30}^2/(2m_{30})^2) = (1/4)$. Furthermore, since the true phases of the 30 subcarriers are independent identically distributed, we have $\mathrm{Var}((1/30)\sum_{i=1}^{30}\angle\mathrm{CSI}_i) = (1/30)\mathrm{Var}(\angle\mathrm{CSI}_i)$ and $\mathrm{Var}(\angle\mathrm{CSI}_i) = \mathrm{Var}(\angle\mathrm{CSI}_{30}) = \mathrm{Var}(\angle\mathrm{CSI}_1)$. We thus have $\mathrm{Var}(\angle\widetilde{\mathrm{CSI}}_i) \le (23/15)\mathrm{Var}(\angle\mathrm{CSI}_i)$, which completes the proof. ■

Theorem 1 provides an upper bound on the variance of the calibrated phase, and indicates that the calibrated phase is relatively more stable. In Fig. 5, we plot the raw phases (as blue crosses) and the calibrated phases (as red dots) in the polar coordinate system for 100 CSI data units for the eighth subcarrier in the first antenna of the IWL 5300 NIC. It can be easily seen that the raw phases scatter randomly over all feasible angles. This is why it is not useful for indoor localization. However, the calibrated phases, after the proposed linear transformation, all concentrate into a sector between 330° and 0°. Thus, the proposed linear transform does remove the phase offset.

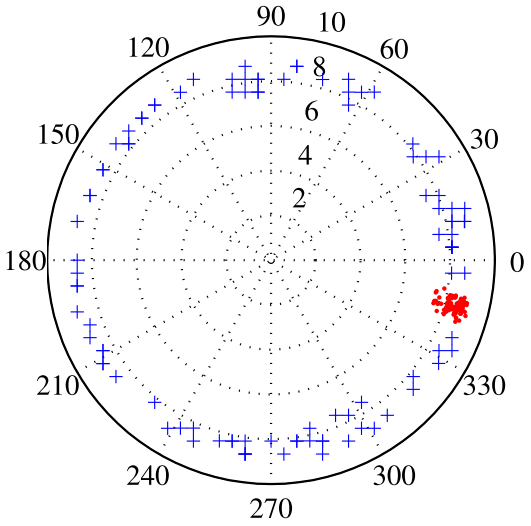


Fig. 5. Raw phase and calibrated phase measurements.

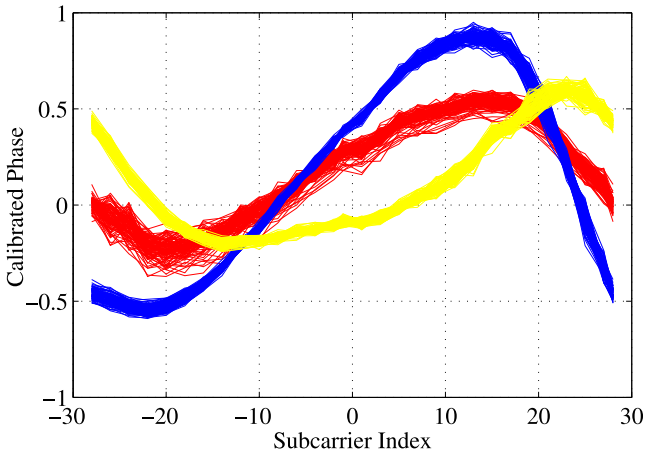


Fig. 6. Calibrated phase values for three different locations.

On the other hand, another characteristic of CSI phase is the great variability at different locations. Fig. 6 plots the calibrated phase for 100 packet receptions from three different positions, from which we can observe that calibrated phases are different for three locations. The calibrated phase not only is more stable in one given location, but also varies in different locations, which can be very useful for indoor fingerprinting.

III. PhaseFi System

A. System Architecture

The architecture of PhaseFi is presented in Fig. 7. In our design, PhaseFi requires one mobile device equipped with an IWL 5300 NIC, which can read CSI data from the slight modified device driver. The IWL 5300 NIC has three antennas, each of which receives from 30 subcarriers. Thus, we can collect 90 CSI data units for one packet reception. Since all the subcarriers are utilized, PhaseFi can effectively improve the diversity of training samples in deep learning, and is thus effective in exploiting the location features for building the fingerprint database. Then, the calibrated phases are obtained by

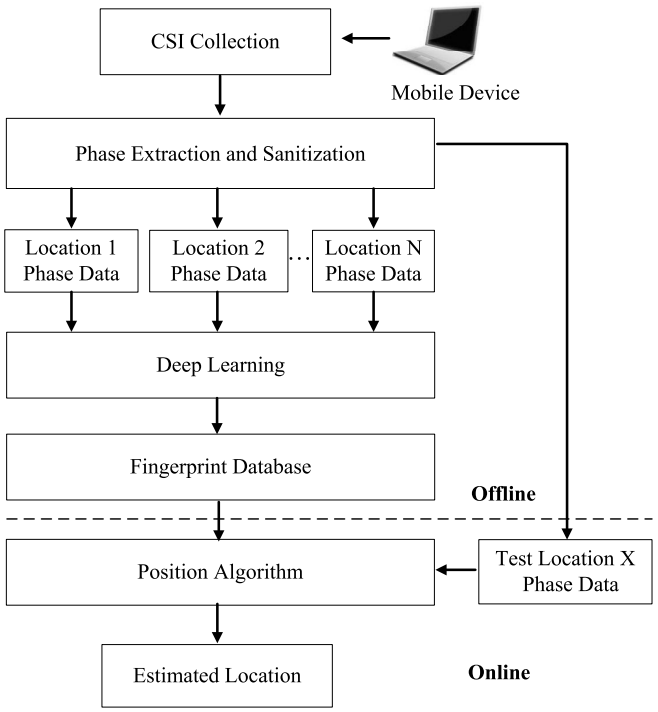


Fig. 7. Architecture of the proposed PhaseFi system.

implementing the proposed linear transformation on the raw phases extracted from CSI data. PhaseFi considers the phase data for indoor fingerprinting for two reasons. First, when a signal encounters obstacle blockages, the amplitude of the signal will be strongly weakened, but the phase of the signal with the periodical change over the propagation distance is relatively more robust. Second, the calibrated phase information is relatively more stable for a given position.

The calibrated phases are then used for both offline training and online testing. In the offline training stage, PhaseFi employs a deep network with three hidden layers to train the calibrated phases. It incorporates deep learning to generate feature-based fingerprints. This approach is different from the traditional methods that directly store the measurement data as fingerprints, which are easily influenced by the complex indoor propagation environment. In addition, a large number of weights in the deep network are used as feature-based fingerprints, which effectively represent the characteristics of the calibrated phases for each position. We create the fingerprint database by training the weights of the deep networks with calibrated phases for different positions. In the online test stage, a probabilistic data fusion approach is used to estimate the mobile device location based on the fingerprint database and the new calibrated phase data from the mobile device.

B. Offline Training

In the offline training stage, PhaseFi incorporates deep learning to train weights and then stores them as the feature-based fingerprint database. The training procedure consists of three stages: 1) pretraining; 2) unrolling; and 3) fine-tuning [15] as shown in Fig. 8. In the pretraining stage, we

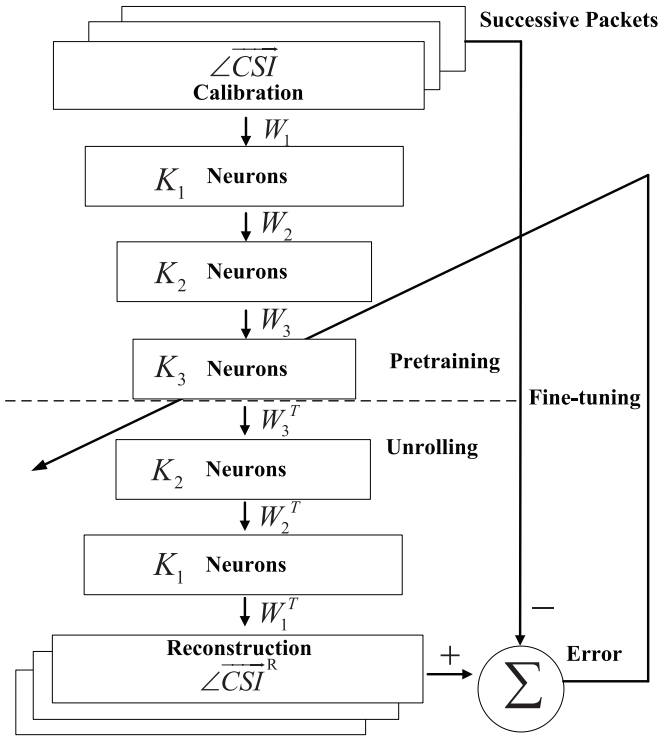


Fig. 8. Training procedure of PhaseFi.

use a deep network with one input layer (with K_0 inputs) and three hidden layers (each with K_i nodes, $k = 1, 2, 3$).

Let h^i denote the hidden variable with K_i nodes at layer i, $i = 1, 2, 3$, and h^0 denote the calibrated phase data.[1] In addition, let W_1, W_2, and W_3 be the weights between the calibrated phase data and the first hidden layer, the first and second hidden layer, and the second and third hidden layer, respectively.

Let $\Pr(h^0, h^1, h^2, h^3)$ denote the probabilistic generative model for the deep network with one input layer and three hidden layers. To obtain the optimal weights in the pretraining stage, we need to maximize the marginal distribution of the calibrated phase data for the deep network, which is formulated by

$$\max_{\{W_1,W_2,W_3\}} \sum_{h^1}\sum_{h^2}\sum_{h^3} \Pr\left(h^0, h^1, h^2, h^3\right). \tag{8}$$

Due to the complex model structure with multiple hidden layers and a large number of nodes in the deep network, it is challenging to obtain the optimal weights using the calibrated phase data with the ML method. In PhaseFi, we develop a greedy learning algorithm to train the weights layer-by-layer by using a stack of RBMs to reduce complexity [16]. For the layer i RBM model, $i = 1, 2, 3$, the joint distribution $\Pr(h^{i-1}, h^i)$ is expressed by an RBM as a bipartite undirected graphical model [16], which is given by

$$\Pr\left(h^{i-1}, h^i\right) = \frac{\exp\left(-\mathbb{E}\left(h^{i-1}, h^i\right)\right)}{\sum_{h^{i-1}}\sum_{h^i}\exp\left(-\mathbb{E}\left(h^{i-1}, h^i\right)\right)} \tag{9}$$

where $\mathbb{E}(h^{i-1}, h^i)$ represents the free energy between layer $i-1$ and layer i. $\mathbb{E}(h^{i-1}, h^i)$ is defined as

$$\mathbb{E}\left(h^{i-1}, h^i\right) = -b^{i-1}h^{i-1} - b^i h^i - h^{i-1}W_i h^i \tag{10}$$

where b^{i-1} and b^i are the biases for units of layer $i-1$ and units of layer i, respectively. In fact, since it is difficult to find the joint distribution $\Pr(h^{i-1}, h^i)$, we use the CD-1 algorithm to approximate it as follows:

$$\begin{cases} \Pr\left(h^{i-1}|h^i\right) = \prod_{j=1}^{K_{i-1}} \Pr\left(h_j^{i-1}|h^i\right) \\ \Pr\left(h^i|h^{i-1}\right) = \prod_{j=1}^{K_i} \Pr\left(h_j^i|h^{i-1}\right) \end{cases} \tag{11}$$

where $\Pr(h_j^{i-1}|h^i)$ and $\Pr(h_j^i|h^{i-1})$ are described by sigmoid belief network, that are

$$\begin{cases} \Pr\left(h_j^{i-1}|h^i\right) = \left[1 + \exp\left(-b_j^{i-1} - \sum_{t=1}^{K_i} W_i^{j,t} h_t^i\right)\right]^{-1} \\ \Pr\left(h_j^i|h^{i-1}\right) = \left[1 + \exp\left(-b_j^i - \sum_{t=1}^{K_{i-1}} W_i^{j,t} h_t^{i-1}\right)\right]^{-1}. \end{cases} \tag{12}$$

We use the greedy algorithm to estimate the parameters of all weights for a stack of RBMs. First, given the calibrated phase data, the parameters $\{b^0, b^1, W_1\}$ of the first layer RBM are estimated by using CD-1 method. Then we freeze the parameters $\{b^0, W_1\}$ of the first layer, and sample from the conditional probability $\Pr(h^1|h^0)$ to train the parameters $\{b^1, b^2, W_2\}$ of the second layer RBM. Next, the parameters $\{b^0, b^1, W_1, W_2\}$ of the first and second layers are frozen, and then we sample from the conditional probability $\Pr(h^2|h^1)$ to train the parameters $\{b^2, b^3, W_3\}$ of the third layer RBM.

To update the weights in each RBM, the CD-1 method is adopted to approximate them. For the layer i RBM model, first, $\hat{h}^{i-1}$ is estimated by sampling from the conditional probability $\Pr(h^{i-1}|h^i)$. Then $\hat{h}^i$ is obtained by sampling from the conditional probability $\Pr(h^i|\hat{h}^{i-1})$. Finally, the parameters are updated as follows:

$$\begin{cases} \Delta W_i = \epsilon\left(h^{i-1}h^i - \hat{h}^{i-1}\hat{h}^i\right) \\ \Delta b^i = \epsilon\left(h^i - \hat{h}^i\right) \\ \Delta b^{i-1} = \epsilon\left(h^{i-1} - \hat{h}^{i-1}\right) \end{cases} \tag{13}$$

where ϵ is the step size.

Once the pretraining stage is completed, we obtain the near-optimal weights for the deep network. Then, in the unrolling stage, the reconstructed calibrated phase data are obtained by unrolling the deep network with forward propagation. Finally, we use the back-propagation algorithm to train all weights in the deep network by computing the error between the input calibrated phase data and the reconstructed calibrated phase data. In addition, the error can be used to iteratively optimize the weights layer-by-layer based on the back-propagation algorithm. This stage is called fine-tuning. After minimizing the error, the optimal weights are stored in the fingerprint database.

The pseudocode for weight training with multiple received packets is presented in Algorithm 2. We first receive n packet

[1] For PhaseFi, we employ three hidden layers to train and test CSI calibrated phase data. This was based on our experimental study, which shows that the three hidden layer structure can achieve near real-time online localization performance as well as low localization errors.

Algorithm 2: Weights Training

1 **Input:** n packet receptions each with 90 CSI calibrated phase values for each of the N training locations;
2 **Output:** N groups of fingerprints each consisting of six weight matrices;
3 **for** $j = 1 : N$ **do**
4 // pretraining;
5 **for** $i = 1 : 3$ **do**
6 initialize $W^i = 0$, $b^i = 0$;
7 **for** $k = 1 : maxepoch$ **do**
8 **for** $t = 1 : n$ **do**
9 $h^0 = v(t)$;
10 Compute $\Pr(h^i|h^{i-1})$ based on the sigmoid with input h^{i-1};
11 Sample h^i from $\Pr(h^i|h^{i-1})$;
12 Compute $\Pr(h^{i-1}|h^i)$ based on the sigmoid with input h^i;
13 Sample $\hat{h}^{i-1}$ from $\Pr(h^{i-1}|h^i)$;
14 Compute $\Pr(h^i|\hat{h}^{i-1})$ based on the sigmoid with input $\hat{h}^{i-1}$;
15 Sample $\hat{h}^i$ from $\Pr(h^i|\hat{h}^{i-1})$;
16 $W_i = W_i + \alpha(h^{i-1}h^i - \hat{h}^{i-1}\hat{h}^i)$;
17 $b^i = b^i + \alpha(h^i - \hat{h}^i)$;
18 $b^{i-1} = b^{i-1} + \alpha(h^{i-1} - \hat{h}^{i-1})$;
19 **end**
20 **end**
21 **end**
22 //unrolling;
23 **for** $i = 1 : 3$ **do**
24 Compute $\Pr(h^i|h^{i-1})$ based on the sigmoid with input h^{i-1};
25 Sample h^i from $\Pr(h^i|h^{i-1})$;
26 **end**
27 Set $\hat{h}^i = h^i$;
28 **for** $i = 3 : 1$ **do**
29 Compute $\Pr(\hat{h}^{i-1}|\hat{h}^i)$ based on the sigmoid with input $\hat{h}^i$;
30 Sample $\hat{h}^{i-1}$ from $\Pr(\hat{h}^{i-1}|\hat{h}^i)$;
31 **end**
32 //fine-tuning;
33 Obtain the error between input data h^0 and reconstructed data $\hat{h}^0$;
34 Update the six weights using the error with back-propagation;
35 **end**

for each of the N training positions, each of which has 90 CSI calibrated phase data units as input data. Let $v(t)$ be the input data from packet t. The output of the training includes N groups of fingerpirnts, each of which owns six weight matrices. Moreover, a deep network for each of the N training locations should be trained. The training phase consists of three steps: 1) pretraining; 2) unrolling; and 3) fine-tuning. For pretraining, the greedy learning algorithm is used to train the deep network with three hidden layers. The weight matrix are initialized first, and are then iteratively updated with the CD-1 method for obtaining initial weights, where m packets are learned and iteratively generate output as input of the next hidden layer (lines 4–21).

After weights training is finished, the input data will be unrolled to get the reconstructed data. First, we utilize the input data to compute $\Pr(h^i|h^{i-1})$ based on the sigmoid with input h^{i-1} to get the coding output h^3, which is a reduced dimension data (lines 23–26). Then, by computing $\Pr(\hat{h}^{i-1}|\hat{h}^i)$ based on the sigmoid with input $\hat{h}^i$, the reconstructed data $\hat{h}^0$ is sampled, where the weights of the deep network are only transposed, thus reducing the time complexity of weights training (lines 27–31). Once the reconstructed data $\hat{h}^0$ is obtained, a supervised learning method based on back-propagation algorithm is used for the deep network as in the fine-tuning phase. Thus, we compute the error between the input data $v = h^0$ and reconstructed data $\hat{h}^0$ to successively update the weight matrix (lines 33 and 34).

C. Position Algorithm

In the online test stage, a probabilistic method is developed to estimate the location of the mobile device based on the fingerprint database and new calibrated phase data. We compute the posteriori probability $\Pr(l_i|h^0)$ based on Bayes' law, which is given by

$$\Pr\left(l_i|h^0\right) = \frac{\Pr(l_i)\Pr\left(h^0|l_i\right)}{\sum_{i=1}^{N}\Pr(l_i)\Pr\left(h^0|l_i\right)} \tag{14}$$

where N is the number of reference locations, l_i is reference location i in the fingerprint database, $\Pr(l_i)$ is the prior probability that the mobile device is determined to locate at the reference location l_i. We assume that $\Pr(l_i)$ follows an uniformly distribution, and then the posteriori probability $\Pr(l_i|h^0)$ can be simplified as follows:

$$\Pr\left(l_i|h^0\right) = \frac{\Pr\left(h^0|l_i\right)}{\sum_{i=1}^{N}\Pr\left(h^0|l_i\right)}. \tag{15}$$

Based on the deep network model, we consider $\Pr(h^0|l_i)$ as the RBF in the form of a Gaussian function to measure the degree of similarity between the reconstructed calibrated phase data $\hat{h}^0$ and the input calibrated phase data h^0, which is given by

$$\Pr\left(h^0|l_i\right) = \exp\left(-\frac{1}{\lambda\sigma}\left\|h^0 - \hat{h}^0\right\|\right) \tag{16}$$

where σ is the variance and λ is the parameter of the variance of the input calibrated phase data. Finally, the position of the mobile device can be computed as a weighted average of all the reference locations, as

$$\hat{l} = \sum_{i=1}^{N}\Pr\left(l_i|h^0\right)l_i. \tag{17}$$

IV. Experimental Validation

A. Experiment Methodology

We examine the performance of PhaseFi with extensive experiments. In our experiments, a TP Link router serves as AP and the mobile device is a Dell laptop equipped with an IWL 5300 NIC. We also modify the NIC's device driver to read CSI values that are recorded in the hardware in the form of CSI for each packet reception. The phase data are extracted from the CSI and calibrated for training and testing.

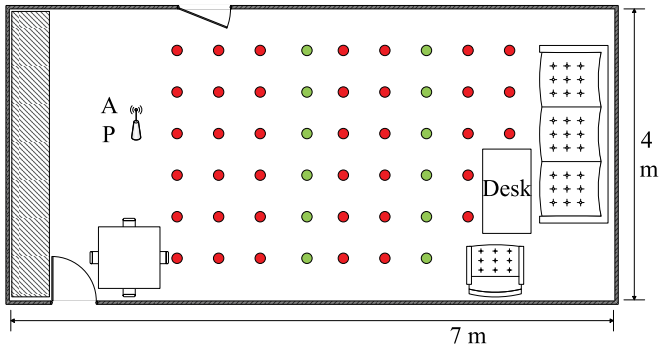


Fig. 9. Layout of the living room for training/test positions.

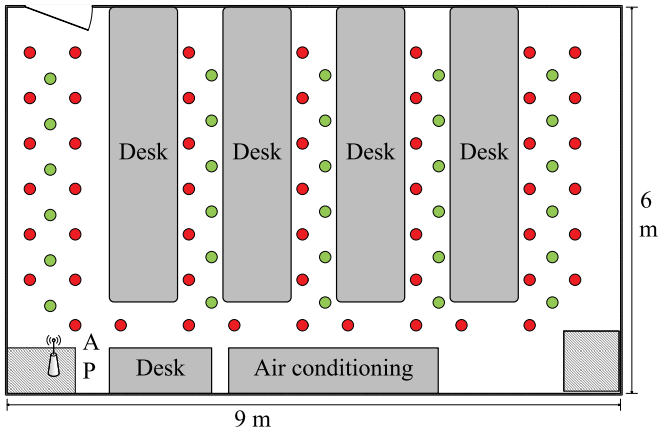


Fig. 10. Layout of the laboratory for training/test positions.

At the AP, the router needs to respond to a mobile device for the localization service. Thus, the Ping command is employed to implement the request and response process between the laptop and the router. The laptop Pings the router, and then the router returns packets to the laptop. In our localization experiment, we write a Java program to implement continuous Pings at a rate of 20 times per second. There are two reasons to choose this rate. First, if we run Ping at a lower rate, no enough packets will be available to determine a mobile device position. The rate of 20 times per second is proper for the online phase in PhaseFi. Second, if too many Pings are run, it is difficult for the laptop to process the received packets with the short time. Also, because we need to continuously determine the mobile device position, it may cause packet loss and buffer overflow. Moreover, once the IWL 5300 NIC receives a packet, the CSI value will be recorded in the hardware in the form of CSI per packet reception. PhaseFi can obtain 90 CSI values and calibrate them for each packet reception, which are all used for weights training or for determining the mobile device position.

In this section, we validate the performance of PhaseFi in two representative indoor environments as follows. First, we conduct experiments in a 4×7 m^2 living room where there are no outstanding obstacles around the center so that most of the measured locations can have LOS receptions. Fig. 9 shows the layout of the living room as well as the training/test points. The AP is placed at one end (rather than the center) of the living room on the floor to avoid isotropy. We set 38 points as training points (in red) and 12 points as test points (in green). In addition, we collect CSI data for 400 packet receptions for each training point, and 20 packet receptions for each test point. A deep network with structure 90 inputs, $K_1 = 60$, $K_2 = 30$, and $K_3 = 15$ is used for the living room environment.

Second, we chose a computer laboratory in Broun Hall in the campus of Auburn University. In this 6×9 m^2 laboratory, there are PCs and many desks crowded in the room such that most of the LOS paths are blocked, thus leading to a complex radio propagation environment. Fig. 10 shows the layout of the laboratory, where we select 50 training points and 30 test points. There is only one AP that is placed on the left bottom corner. To obtain integrated characteristics of the subcarriers, we read CSI data for 800 packet receptions for each training point, and 20 packet receptions for each test point. The structure of the deep network in the laboratory environment is the same as that in the living room environment.

TABLE I
MEAN ERRORS AND EXECUTION TIME (LIVING ROOM)

Algorithm	*Mean error (m)*	*Std. dev. (m)*	*Mean exe. time (s)*
PhaseFi	1.0800	0.4046	0.3780
FIFS	1.2436	0.5705	0.2362
Horus	1.5449	0.7024	0.2297
ML	2.1615	1.0416	0.2290

TABLE II
MEAN ERRORS AND EXECUTION TIME (LABORATORY)

Algorithm	*Mean error (m)*	*Std. dev. (m)*	*Mean exe. time (s)*
PhaseFi	2.0134	1.0139	0.3770
FIFS	2.3304	1.0219	0.2439
Horus	2.5996	1.4573	0.2214
ML	2.8478	1.5545	0.2220

For comparison purpose, we implement three existing methods, including FIFS [10], Horus [6], and ML [17]. FIFS and Horus are introduced in Section I. In ML, based on RSS measurements, only one reference location with maximum posterior probability is considered as the estimated result. For a fair comparison, all schemes use the same measured data set to estimate the position of the mobile device.

The performance metric for the comparison of localization algorithms is the mean sum error $\mathcal{E}$. Consider the estimated location of an unknown user i is $(\hat{x}_i, \hat{y}_i)$ and the actual position of the user is (x_i, y_i). For M estimated locations, the mean sum error is computed as

$$\mathcal{E} = \frac{1}{M} \sum_{i=1}^{M} \sqrt{\left(\hat{x}_i - x_i\right)^2 + \left(\hat{y}_i - y_i\right)^2}. \tag{18}$$

B. Localization Performance

Tables I and II present the mean location errors, the standard deviations, and the average execution time of the living room and the laboratory experiments, respectively. In the living room environment, we find PhaseFi to achieve a mean location error of 1.08 m and a standard deviation of 0.4046 m for the 12 test points. In the laboratory scenario, the error is higher

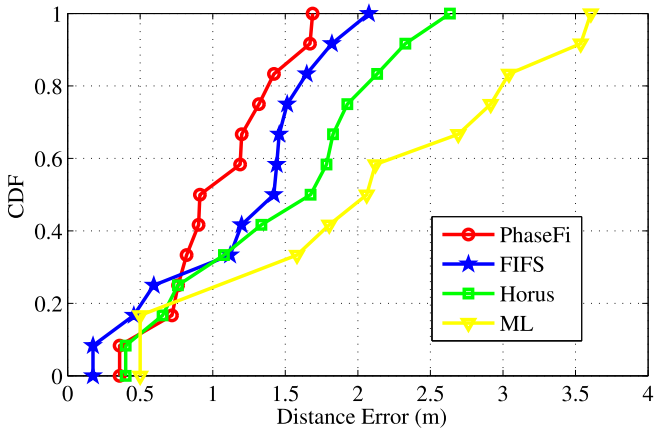


Fig. 11. CDF of localization errors of the living room experiments.

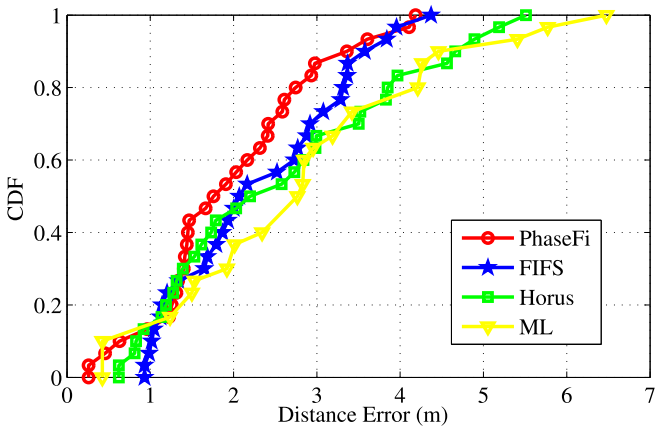


Fig. 12. CDF of localization errors of the laboratory experiments.

due to abundant multipath and shadowing effect. Our system achieves a mean error of 2.0134 m and a standard deviation of 1.0139 m for the 30 test points. For indoor localization accuracy, PhaseFi based on calibrated phases outperforms all the other schemes (i.e., FIFS, Horus, and ML) that are based on amplitudes. PhaseFi also demonstrates robust performance for different locations with the smallest standard deviation. We also examine the computational complexity of the schemes. Although the mean execution time for PhaseFi is higher than the benchmark schemes, the 0.38 s average execution time of PhaseFi for both scenarios still satisfies the real-time requirement for most indoor localization applications. In fact, by optimizing the parameters and reducing the number of nodes in the deep network, the average execution time of PhaseFi can be further reduced.

Fig. 11 shows the cumulative distribution functions (CDFs) of distance errors with the four schemes in the living room scenario. For PhaseFi, more than 50% of the test points have an error under 0.9 m using one AP, while the other schemes guarantee that 30% of the test points have an error under 0.9 m. Moreover, PhaseFi and FIFS have approximate 80% of the test points with mean location errors under 1.5 m, while Horus and ML have the same test points with mean error under 2.0 and 3.0 m, respectively. The CSI-based schemes such as PhaseFi and FIFS can utilize the fined-grained subcarrier information, and are thus more stable than the RSS-based schemes.

Fig. 12 presents the CDFs of distance errors achieved with the four schemes in the laboratory environment. In this more complex propagation environment, for PhaseFi, about 60% of the test points have a distance error under 2 m, while the other schemes have the same portion of test points with error under 2.7 m. We find PhaseFi to be the most accurate among the four schemes, because the phase of the signal periodically changes over the propagation distance, which is more robust than amplitude, especially in cluttered propagation environments. The signal amplitude is usually more vulnerable to transmission impairments, and the correlation between signal strength and propagation distance is usually weak in indoor scenarios. Thus PhaseFi outperforms the three amplitude-based schemes (based on either CSI or RSS).

Fig. 13 shows the mean localization errors versus different number of packets in the laboratory and living room

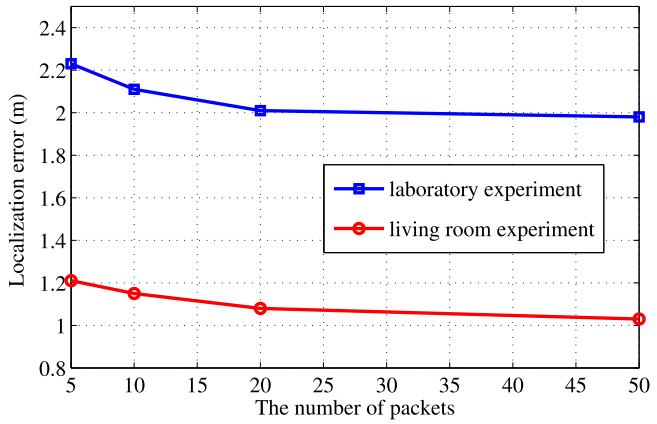


Fig. 13. Mean localization errors for different number of packets in the laboratory and living room experiments.

experiments. In both experiments, the distance error is decreased as more packets are used. In particular, the mean distance error is decreased from 1.21 to 1.03 m in the living room experiment, and from 2.23 to 1.98 m in the laboratory experiment, when the number of packets is increased from 5 to 50. Only small reduction in localization error is achieved when the number of packets is increased for ten times. Thus, we use 20 packets for online test in PhaseFi, which achieves not only a good localization accuracy, but also a low computational complexity for real-time localization applications.

V. Related Works

Indoor localization has attracted considerable interest in the research community with many interesting prior works [18], [19]. For brevity, we only review two categories of indoor localization methods that are closely related to the proposed PhaseFi system in this section, i.e., fingerprinting-based and phase-based systems.

A. Fingerprinting-Based System

Fingerprinting-based localization systems are the mainstream approach for indoor localization [20], [21]. The first work for RSS-based indoor localization methods is RADAR [7], which utilizes K-nearest-neighbor to estimate

mobile user location. To improve localization accuracy, Horus [6] considers probabilistic method for indoor localization based on RSS. On the other hand, CSI values are employed for indoor localization to avoid the instability of RSS values [22]. FIFS [10] and PinLoc [23] systems utilize CSI amplitude values extracted from off-the-shelf IWL 5300 NIC as fingerprints to estimate mobile user position. To improve localization accuracy based on CSI values, DeepFi [11], [24] system leverages a large amount of CSI amplitude data from three antennas for indoor localization based on a deep network with four hidden layers. Although these systems based on RSS and CSI obtain high indoor localization accuracy, they need large amounts of training samples to build the database. Therefore, a new method to reduce the burden of wardriving is crowdsourcing-based database construction, which shares the load of fingerprints obtained by multiple users [25]. Zee [26] employs the inertial sensors and particle filtering to predict users walking trace, and to collect the fingerprints with WiFi data as crowdsourced measurements to do the training step. LiFS [27] utilizes users trajectories to get fingerprint values and then creates the mapping between the fingerprints and the floor plan. Recently, the fundamental limits of fingerprinting-based schemes are analyzed in [28] and fusing noisy fingerprints with a distance bound is also considered in [29].

B. Phase-Based System

Phase-based localization systems consider angle-of-arrival (AOA) or time-difference of-arrival (TDOA) to estimate user locations. For AOA-based methods, indoor localization systems employ multiple antennas to compute the incoming angles and estimate the location of one user by using geometric relationships. By using off-the-shelf Atheros chipsets with three antennas, the CUPID system [30] can obtain CSI to estimate AOA with the mean error about 20 degree based on MUSIC algorithm. Because of a low resolution of three antennas for the CUPID system, it achieves a poor localization performance. To improve localization accuracy, the array-track system [31] employs a rectangular array of 16 antennas to estimate the AOA, and then utilizes spatial smoothing to reduce the effect of multipath on AOA. However, this array-track system with 16 antennas is not suitable for commodity mobile devices. On the other hand, for TDOA-based methods, ToneTrack system [32] employs at least three WARP-based APs to estimate user location, and requires time synchronization for all APs.

The proposed PhaseFi system is the first to use the calibrated phase information for indoor fingerprinting with commodity WiFi devices, by exploiting the diversity of the OFDM channel. It effectively overcomes the problem of the low resolution of antennas array in commodity mobile devices, requires no time synchronization and no extra cost on new infrastructure support.

VI. Conclusion

In this paper, we proposed PhaseFi, a phase fingerprinting system for indoor localization. In the system, the phase information was first extracted and calibrated from the three antennas of the IWL 5300 NIC by accessing the modified device driver. In the offline stage, we designed a deep network with three hidden layers to train the calibrated phase data, and used weights to represent fingerprints. To reduce complexity, a greedy learning algorithm was incorporated to train the weights layer-by-layer, where a subnetwork between two consecutive layers formed an RBM approximately and solved by a CD-1 algorithm. In the online stage, a Bayes method based on RBF was used for location estimation. The proposed PhaseFi scheme was validated in two representative indoor environments, and was shown to outperform three benchmark schemes based on either CSI or RSS in both scenarios.

References

[1] X. Wang, L. Gao, and S. Mao, "PhaseFi: Phase fingerprinting for indoor localization with a deep learning approach," in *Proc. IEEE GLOBECOM*, San Diego, CA, USA, Dec. 2015, pp. 1–6.

[2] X. Wang, S. Mao, S. Pandey, and P. Agrawal, "CA2T: Cooperative antenna arrays technique for pinpoint indoor localization," in *Proc. MobiSPC*, Niagara Falls, ON, Canada, Aug. 2014, pp. 392–399.

[3] X. Wang *et al.*, "Mobility improves LMI-based cooperative indoor localization," in *Proc. IEEE WCNC*, New Orleans, LA, USA, Mar. 2015, pp. 2215–2220.

[4] H. Liu, H. Darabi, P. Banerjee, and L. Jing, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[5] S. Dayekh, S. Affes, N. Kandil, and C. Nerguizian, "Cooperative localization in mines using fingerprinting and neural networks," in *Proc. IEEE WCNC*, Sydney, N.S.W., Australia, Apr. 2010, pp. 1–6.

[6] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. ACM MobiSys*, Seattle, WA, USA, Jun. 2005, pp. 205–218.

[7] P. Bahl and V. N. Padmanabhan, "Radar: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 775–784.

[8] K. Wu *et al.*, "CSI-based indoor localization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1300–1309, Jul. 2013.

[9] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," in *Proc. ACM SIGCOMM*, New Delhi, India, Sep. 2010, pp. 159–170.

[10] J. Xiao, K. Wu, Y. Yi, and L. Ni, "FIFS: Fine-grained indoor fingerprinting system," in *Proc. IEEE ICCCN*, Munich, Germany, Jul./Aug. 2012, pp. 1–7.

[11] X. Wang, L. Gao, S. Mao, and S. Pandey, "DeepFi: Deep learning for indoor fingerprinting using channel state information," in *Proc. IEEE WCNC*, New Orleans, LA, USA, Mar. 2015, pp. 1666–1671.

[12] Y. Chapre, A. Ignjatovic, A. Seneviratne, and S. Jha, "CSI-MIMO: Indoor Wi-Fi fingerprinting system," in *Proc. IEEE LCN*, Edmonton, AB, Canada, Sep. 2014, pp. 202–209.

[13] C. Wu *et al.*, "PhaseU: Real-time LOS identification with WiFi," in *Proc. IEEE INFOCOM*, Hong Kong, Apr./May 2015, pp. 2038–2046.

[14] K. Qian, C. Wu, Z. Yang, Y. Liu, and Z. Zhou, "PADS: Passive detection of moving targets with dynamic speed using PHY layer information," in *Proc. IEEE ICPADS*, Hsinchu, Taiwan, Dec. 2014, pp. 1–8.

[15] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

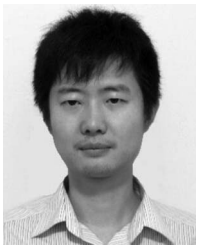
[16] Y. Bengio *et al.*, "Greedy layer-wise training of deep networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, Dec. 2007, pp. 153–160.

[17] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless LANs," *Comput. Netw.*, vol. 47, no. 6, pp. 825–845, Apr. 2005.

[18] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 13–32, Jan. 2009.

[19] S. He and S.-H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 466–490, Jan. 2016.

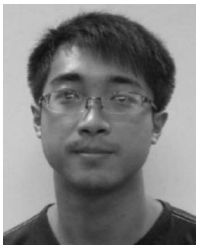
[20] C. Wu *et al.*, "Static power of mobile devices: Self-updating radio maps for wireless indoor localization," in *Proc. IEEE INFOCOM*, Hong Kong, Apr./May 2015, pp. 2497–2505.
[21] D. Liang, Z. Zhang, and M. Peng, "Access point reselection and adaptive cluster splitting-based indoor localization in wireless local area networks," *IEEE Internet Things J.*, vol. 2, no. 6, pp. 573–585, Dec. 2015.
[22] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Comput. Surv.*, vol. 46, no. 2, pp. 25–32, Nov. 2013.
[23] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, "You are facing the Mona Lisa: Spot localization using PHY layer information," in *Proc. ACM MobiSys*, Ambleside, U.K., Jun. 2012, pp. 183–196.
[24] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Veh. Technol.*, to be published, doi: 10.1109/TVT.2016.2545523.
[25] X. Zhang *et al.*, "Robust trajectory estimation for crowdsourcing-based mobile applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1876–1885, Jul. 2014.
[26] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. ACM Mobicom*, Istanbul, Turkey, Aug. 2012, pp. 293–304.
[27] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Proc. ACM Mobicom*, Istanbul, Turkey, Aug. 2012, pp. 269–280.
[28] Y. Wen, X. Tian, X. Wang, and S. Lu, "Fundamental limits of RSS fingerprinting based indoor localization," in *Proc. IEEE INFOCOM*, Hong Kong, Apr./May 2015, pp. 2479–2487.
[29] S. He, S.-H. G. Chan, L. Yu, and N. Liu, "Fusing noisy fingerprints with distance bounds for indoor localization," in *Proc. IEEE INFOCOM*, Hong Kong, Apr./May 2015, pp. 2506–2514.
[30] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding WiFi localization," in *Proc. ACM MobiSys*, Taipei, Taiwan, Jun. 2013, pp. 249–262.
[31] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *Proc. ACM NSDI*, Lombard, IL, USA, Apr. 2013, pp. 71–84.
[32] J. Xiong, K. Sundaresan, and K. Jamieson, "Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization," in *Proc. ACM Mobicom*, Paris, France, Sep. 2015, pp. 537–549.



Xuyu Wang (S'13) received the B.S. degree in electronic information engineering and M.S. degree in signal and information processing from Xidian University, Xi'an, China, in 2009 and 2012, respectively and is currently working toward the Ph.D. degree in electrical and computer engineering at Auburn University, Auburn, AL, USA.

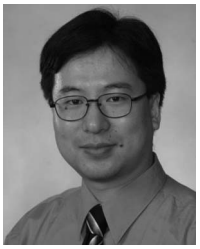
His current research interests include indoor localization, deep learning, wireless communications, software defined radio, and big data.

Mr. Wang was a recipient of the Woltolsz Fellowship at Auburn University. He was a corecipient of the Second Prize of Natural Scientific Award of Ministry of Education, China, in 2013.



Lingjun Gao (S'14) received the B.E. degree in electrical engineering from the Civil Aviation University of China, Tianjing, China, in 2013, and the M.S. degree in electrical and computer engineering from Auburn University, Auburn, AL, USA, in 2015.

He is currently a Data Engineer with DataYes Inc., Shanghai, China. His current research interests include machine learning, indoor localization, and testbed implementation.



Shiwen Mao (S'99–M'04–SM'09) received the Ph.D. degree in electrical and computer engineering from the Polytechnic University, Brooklyn, NY, USA, in 2004.

He is currently the Samuel Ginn Distinguished Professor and the Director of the Wireless Engineering Research and Education Center, Auburn University, Auburn, AL, USA. His current research interests include wireless networks and multimedia communications.

Dr. Mao is a Distinguished Lecturer of the IEEE Vehicular Technology Society. He is on the Editorial Board of IEEE TRANSACTIONS ON MULTIMEDIA, IEEE INTERNET OF THINGS JOURNAL, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and IEEE TRANSACTIONS ON MULTIMEDIA. He serves as an Area TPC Chair of IEEE INFOCOM 2017 and 2016, the Technical Program Vice Chair for Information Systems of IEEE INFOCOM 2015, the Symposium/Track Co-Chair for many conferences, including the IEEE ICC, the IEEE GLOBECOM, and ICCCN, the Steering Committee Voting Member for the IEEE ICME and AdhocNets, and in various roles in the organizing committees of many conferences. He is the Vice Chair of Letters and Member Communications of the IEEE ComSoc Multimedia Communications Technical Committee. He was a recipient of the 2015 IEEE ComSoc TC-CSR Distinguished Service Award, the 2013 IEEE ComSoc MMTC Outstanding Leadership Award, and the National Science Foundation CAREER Award in 2010. He was a corecipient of the IEEE GLOBECOM 2015 Best Paper Award, the IEEE WCNC 2015 Best Paper Award, the IEEE ICC 2013 Best Paper Award, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems.