DeepFi: Deep learning for indoor fingerprinting using channel state information

Conference Paper · March 2015

DOI: 10.1109/WCNC.2015.7127718

Xuyu Wang
Auburn University

Santosh Pandey
Cisco Systems, Inc

DeepFi: Deep Learning for Indoor Fingerprinting Using Channel State Information

Xuyu Wang†, Lingjun Gao†, Shiwen Mao†, and Santosh Pandey‡
†Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201
‡Cisco Systems, Inc., 170 West Tasman Dr., San Jose, CA 95134
Email: {xzw0029, lzg0014}@auburn.edu, smao@ieee.org, sanpande@cisco.com

***Abstract*—With the fast growing demand of location-based services in indoor environments, indoor positioning based on fingerprinting has attracted a lot of interest due to its high accuracy. In this paper, we present a novel deep learning based indoor fingerprinting system using Channel State Information (CSI), which is termed DeepFi. Based on three hypotheses on CSI, the DeepFi system architecture includes an off-line training phase and an on-line localization phase. In the off-line training phase, deep learning is utilized to train all the weights as fingerprints. Moreover, a greedy learning algorithm is used to train all the weights layer-by-layer to reduce complexity. In the on-line localization phase, we use a probabilistic method based on the radial basis function to obtain the estimated location. Experimental results are presented to confirm that DeepFi can effectively reduce location error compared with three existing methods in two representative indoor environments.**

I. Introduction

With the proliferation of mobile devices, indoor localization has become an increasingly important problem. Unlike outdoor localization, such as the Global Positioning System (GPS), that has line-of-sight (LOS) signals, indoor localization faces a challenging radio propagation environment, including multipath effect, shadowing, channel fading and time delay [1], **[2]**. In addition to the high accuracy requirement, an indoor positioning system should also have short estimation process time and low complexity for mobile devices. To this end, fingerprinting-based indoor localization becomes an effective method to satisfy these requirements, where an enormous amount of measurements are essential to build a database before real-time position estimation.

Fingerprinting localization usually consists of two fundamental phases: (i) the off-line phase, which is also called the training phase, and (ii) the on-line phase, which is also called the test phase [3]. The training phase is for database construction, when survey data related to the position marks is collected and pre-processed. In the on-line phase, a mobile device records real time data and tests it using the database. The test output is then used to estimate the position of the mobile device, by searching each training point to find the most closely matched one as the target location. Besides such nearest estimation method, an alternative matching algorithm is to identify several close points each with a maximum likelihood probability, and to calculate the estimated position as the weighted average of the candidate positions.

In the off-line training stage, machine learning methods can be used to train fingerprints instead of storing all the received signal strength (RSS). Such machine learning methods not only reduce the computational complexity, but also obtain the core features in the RSS for better localization performance. K-nearest-neighbor, neural networks, and support vector machine as popular machine learning methods that have been applied for fingerprinting based indoor localization. K-nearest-neighbor uses the weighted average of K nearest locations to determine an unknown location with the inverse of the Euclidian distance between the observed RSS measurement and its K nearest training samples [1]. A limitation of K-nearest-neighbor is that it needs to store all the RSS training values. Neural networks utilizes the back-propagation algorithm to train weights, but it only considers one hidden layer and needs label data as a supervised learning [4]. Support vector machine uses kernel function to solve the randomness and incompleteness of the RSS values, which has high computing complexity [5].

Many existing indoor localization systems use RSS values as fingerprints due to its simplicity and low hardware requirements. For example, the Horus system uses a probabilistic method for location estimation with RSS values [6]. Such RSS based methods have two disadvantages. First, RSS values usually have a high variability over time for a fixed location, due to the multipath effects in indoor environments. Such high variability can introduce large location error even for a stationary device. Second, RSS values are coarse information, which does not exploit the subcarriers in an orthogonal frequency-division multiplexing (OFDM) for richer multipath information. Nowadays, it is possible to obtain channel state information (CSI) from some advanced WiFi network interface cards (NIC), which can be used as fingerprints to improve the performance of indoor localization [7], [8]. For instance, the authors in [9] propose an FIFS system that uses the weighted average CSI values over multiple antennas to improve the performance of RSS-based method. In addition, the authors in [10] present a PinLoc system that also exploits CSI information, which considers 1×1 m^2 spot for training data, thus leading to high war-driving.

In this paper, we propose a deep learning based fingerprinting scheme to mitigate the above problems of existing machine learning based methods. The deep learning based scheme can fully explore the feature of the wireless channel data and obtain the optimal weights as fingerprints. It also incorporates a greedy learning algorithm to reduce computational complexity, which has been successfully applied in image processing

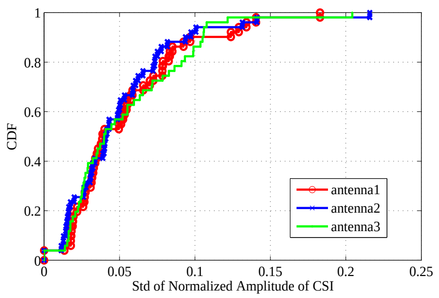

Fig. 1. Standard deviation of normalized amplitudes of CSI values.

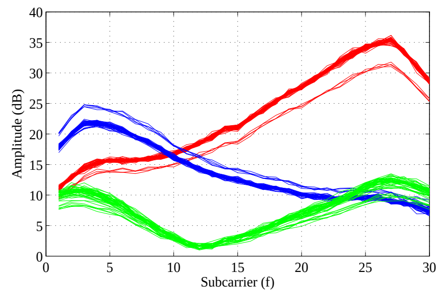

Fig. 2. Amplitudes of channel frequency responses of 50 packets at three different positions.

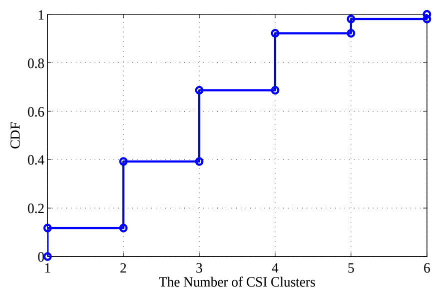

Fig. 3. CDF of the number of channel frequency responses at 50 different locations.

and voice recognition [11]. The proposed scheme is based on channel state information (CSI) to obtain more information about the wireless channels than RSS based schemes. The proposed scheme is also different from the existing CSI based schemes, in that it incorporates 90 magnitudes of CSI values collected from three antennas to train the weights of a deep network with deep learning. As a result, our method does not require to sample a large number of positions.

In particular, we present DeepFi, a deep learning based indoor fingerprinting scheme using CSI. We first introduce the background of CSI and offer three hypotheses based on CSI. We then present the DeepFi system architecture, which includes an off-line training phase and an on-line localization phase. In the training phase, CSI information for all the subcarriers from three antennas are collected from accessing the device driver and are analyzed with a deep network with four hidden layers. We propose to use the weights in the deep network to represent fingerprints, and to incorporate a greedy learning algorithm to reduce training complexity. In the on-line localization phase, a probabilistic data fusion method based on radial basis function is developed for online location estimation. The proposed DeepFi scheme is validated under extensive experiments in two representative indoor environments, i.e., an living room environment and a computer laboratory environment. DeepFi is shown to outperform several existing RSSI and CSI based schemes in both experiments.

The remainder of this paper is organized as follows. The background and hypotheses are presented in Section II. In Section III, we offer the DeepFi system. Experimental results are discussed in Section IV. Section V concludes this paper.

II. Background and Hypotheses

A. Channel State Information

Thanks to the advanced NICs, such as Intel's IWL 5300, it is now much easier to conduct channel state measurements than in the recent past when we have to detect hardware records for physical layer (PHY) information. Now CSI can be retrieved from a laptop by accessing the NIC's device drive. CSI records the channel variation experienced during propagation, which can be used to analyze the properties of wireless signals. Transmitted from a source, a wireless signal may experience abundant impairments caused by, e.g., the multipath effect, fading, shadowing, and delay distortion. Without CSI, it is hard to reveal the channel characteristics with only the signal power.

Let X and Y denote as the transmitted and received vectors. We have

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

where vector N is the additive white Gaussian noise. CSI stands for the channel frequency response. which can be estimated by the expression CSI $= Y/X$.

The WiFi channel at the 2.4 GHz band can be considered as a narrowband flat fading channel. Intel WiFi Link 5300 implements an OFDM system with 48 subcarriers, 30 out of which can be read as CSI information via the device driver. The channel frequency response $\mathbf{CSI}_i$ of the i_{th} subcarrier is a complex value, which is defined by

$$\mathbf{CSI}_i = |\mathbf{CSI}_i| \exp\left(j \sin \angle \mathbf{CSI}_i\right). \tag{2}$$

where $|\mathbf{CSI}_i|$ and $\angle\mathbf{CSI}_i$ are the amplitude response and the phase response of the i_{th} subcarrier, respectively. In this paper, the proposed DeepFi framework is based on these 30 subcarriers (or, CSI values) in the OFDM system, which can reveal completely different properties than RSSI.

B. Hypotheses

We next propose three hypotheses on the CSI values, which are verified later with the statistical measurement results.

$\boxed{\text{Hypotheses 1}}$ CSI values maintain stability at a stationary location but exhibit variability between adjacent positions.

CSI values reflect channel properties in the frequency domain and exhibit great stability over time. Fig. 1 plots the CDF of the standard deviation of CSI amplitudes, collected from three antennas when receiving 50 packets. It shows that the standard deviation of CSI amplitudes at 90% positions is 10% of the average value, and therefore CSI values have higher stability than RSSI, which usually has more than 30% normalized standard deviation. The stability of CSI values is also invariant to indoor environment changes. Our measurements last a long period covering both office hours and quiet hours. But no obvious difference in stability is found for different environments. On the contrary, RSS values usually vary greatly due to multipath even at the same position. It can be easily disturbed by human movements.

On the other hand, another characteristic of CSI values is the apparent variability when the estimated position is away from the access point (AP). Fig. 2 plots the subcarriers of CSI for 50 back-to-back packet receptions from three adjacent positions, from which hardly any similar trend can be observed. In order to maintain the correlation between test and training points, we partition the floor into small grids and densely sample training points.

Hypotheses 2 The multipath effect on CSI values reveals various clusters with different attenuation for the subcarriers.

CSI values reflect channel frequency responses with abundant multipath components and channel fading. The indoor environment can be view as a time-varying channel model, and therefore CSI may change slightly over time. Our study of channel frequency responses show that there are several dominant clusters at a position, where each cluster presents a multipath combination. Fig. 3 shows the distribution of the number of clusters according to packets for 50 locations. As shown in Fig. 3, when we define that one cluster has the same CSI subcarrier trend, most of the locations have two or three clusters. We also find that some locations has only one cluster, which usually means that there is less reflection and diffusion; some other locations with five or six clusters may suffer much from the multipath effect.

To consider all possible of the number of clusters, we detected for a long time in each position. Since lots of data are needed to train specific characteristics in deep learning, more packet transmissions will be helpful to reveal the comprehensive properties at each spot. In our experiments, 1000 packets are recorded for training at each location instead of 60 packets in the FIFS system.

Hypotheses 3 The three antennas of Intel WiFi Link 5300 NIC have different features, which improve the diversity of training samples.

Intel WiFi Link 5300 is equipped with three antennas. We find that the channel frequency responses of the three antennas are totally different, even for the same packet reception. In Fig. 7, signals from the three antennas exhibit different properties in CSI subcarriers. In FIFS, CSI from the three antennas are simply accumulated to produce an average value. In contrast, DeepFi aims to utilize their varieties to enhance the training process in deep learning. The 30 subcarriers can be treated as 30 nodes and used as input data of visible variability for deep learning training. Due to the three antennas of the Intel WiFi Link 5300 NIC, there can be 90 nodes in total that can be used as input data for deep learning training. The greatly increased number of nodes for input data can improve the diversity of training samples, thus leading to better performance of localization if reasonable parameters are chosen.

III. The DeepFi System

A. System Architecture

Fig. 4 shows the system architecture of DeepFi. In our design, DeepFi only needs one access point and one mobile device equipped with an Intel WiFi link 5300 NIC. At the mobile device, raw CSI values can be read from the modified chipset firmware for packets received from the AP. The Intel WiFi link 5300 NIC has three antennas, each of which can collect 30 different subcarriers. We can obtain 90 raw CSI values for one packet reception. Unlike FIFS that averages over multiple antennas to reduce the received noise, our system exploits all CSI values from the three antennas for indoor fingerprint to exploit diversity of the MIMO channel. Since it is hard to use the phases of CSI values for localization, we only consider the amplitude responses of CSI values for fingerprinting. On the other hand, since the input values are limited in the range of (0, 1) for effective deep learning, we normalize the amplitudes of the 90 CSI values for different locations, which can be utilized in both the offline training phase and the online localization phase.

In the offline training phase, we can generate feature-based fingerprints, which are greatly different from the traditional methods that are based on clustering. Feature-based fingerprints utilize a large number of weights obtained by deep learning to denote different locations, which effectively describe the characteristics of CSI values. The feature-based fingerprints server can store all weights for different training locations. In the online localization phase, the mobile device can estimate its position based on data fusion, which normalizes the magnitudes of CSI values using weights from different positions to get its estimated location.

B. Weight Training with Deep Learning

Fig. 5 illustrates how to train weights based on deep learning. There are three stages in the procedure, including pretraining, unrolling, and fine-tuning [12]. In the pretraining stage, it is a deep network with four hidden layers, where every hidden layer owns a different number of neurons. In order to reduce the dimension of CSI data, we assume that the number of neurons in a higher hidden layer is more than that in a lower hidden layer. Let K_1, K_2, K_3 and K_4 denote the number of neurons in the first, second, third, and fourth hidden layer, respectively. Based on the above assumption, we have $K_1 > K_2 > K_3 > K_4$. We can set different numbers of neurons in different indoor environments.

In addition, we propose a new approach to represent fingerprints by the weights between two connected layers. Define W_1, W_2, W_3 and W_4 as the weights between the normalized magnitudes of CSI values and the first hidden layer, the first and second hidden layer, the second and third hidden layer, and the third and fourth hidden layer, respectively. The key idea is that by training all weights in the deep network, we can store them as fingerprints to help localization in the online test stage. Moreover, we define h_i as the hidden variable at layer i, where $i = 1, 2, 3, 4$, and let v denote the input data, i.e., the normalized magnitudes of CSI values.

We can represent the deep network with a probabilistic generative model that has four hidden layers, which can be written as

$$\begin{aligned} & \Pr(v, h^1, h^2, h^3, h^4) \\ = & \Pr(v|h^1) \Pr(h^1|h^2) \Pr(h^2|h^3) \Pr(h^3, h^4). \end{aligned} \tag{3}$$

Because the nodes in the deep network is mutually independent, $\Pr(v|h^1)$, $\Pr(h^1|h^2)$, and $\Pr(h^2|h^3)$ can be represented

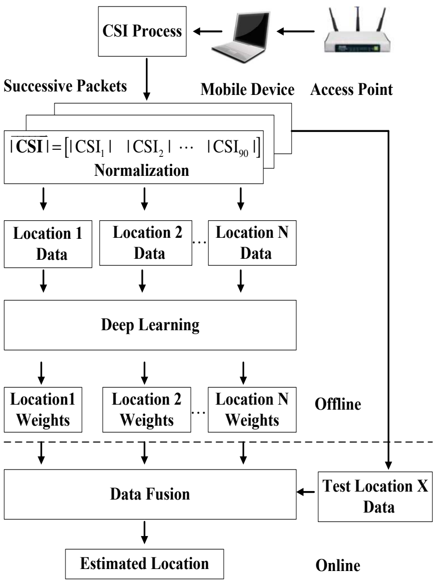

Fig. 4. DeepFi Architecture.

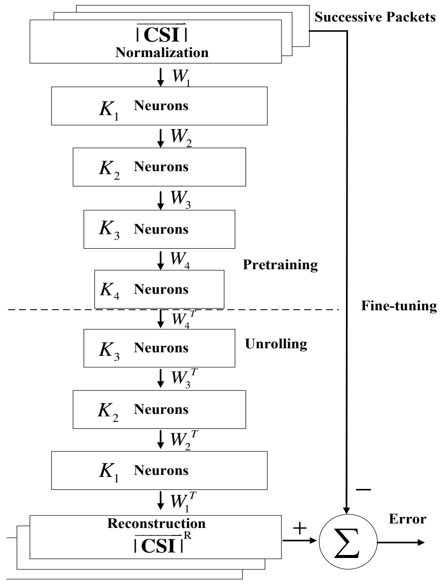

Fig. 5. Weight training with deep learning.

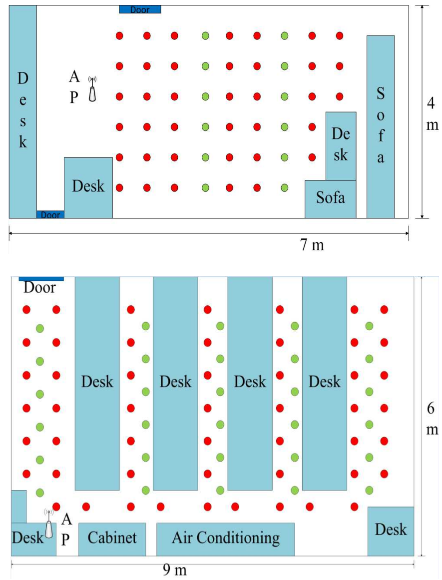

Fig. 6. Layout of the living room (top) and the laboratory (bottom) and training/test positions.

by

$$\begin{cases} \Pr(v|h^1) = \prod_{i=1}^{90} \Pr(v_i|h^1) \\ \Pr(h^1|h^2) = \prod_{i=1}^{K_1} \Pr(h_i^1|h^2) \\ \Pr(h^2|h^3) = \prod_{i=1}^{K_2} \Pr(h_i^2|h^3), \end{cases} \tag{4}$$

where $\Pr(v_i|h^1)$, $\Pr(h_i^1|h^2)$, and $\Pr(h_i^2|h^3)$ are described by the sigmoid belief network in the deep network, as

$$\begin{cases} \Pr(v_i|h^1) = 1/\left(1 + \exp\left(b_i^0 - \sum_{j=1}^{K_1} W_1^{i,j} h_j^1\right)\right) \\ \Pr(h_i^1|h^2) = 1/\left(1 + \exp\left(b_i^1 - \sum_{j=1}^{K_2} W_2^{i,j} h_j^2\right)\right) \\ \Pr(h_i^2|h^3) = 1/\left(1 + \exp\left(b_i^2 - \sum_{j=1}^{K_3} W_3^{i,j} h_j^3\right)\right), \end{cases} \tag{5}$$

where b_i^0, b_i^1 and b_i^2 are the biases for unit i of input data v, unit i of layer 1, and unit i of layer 2, respectively. On the other hand, the joint distribution $\Pr(h^3, h^4)$ is expressed by an Restricted Bolzmann Machine (RBM) as a bipartite undirected graphical model [13], which is given by

$$\Pr(h^3, h^4) = \exp(-\mathbb{E}(h^3, h^4))/Z, \tag{6}$$

where $Z = \sum_{h^3} \sum_{h^4} \exp(-\mathbb{E}(h^3, h^4))$ and $\mathbb{E}(h^3, h^4) = -\sum_{i=1}^{K_3} b_i^3 h_i^3 - \sum_{j=1}^{K_4} b_j^4 h_j^3 - \sum_{i=1}^{K_3} \sum_{j=1}^{K_4} W_4^{i,j} h_i^3 h_j^4$. In fact, since it is difficult to find the joint distribution $\Pr(h^3, h^4)$, we use the contrastive divergence (CD) algorithm to approximate it, which is given by

$$\begin{cases} \Pr(h^3|h^4) = \prod_{i=1}^{K_3} \Pr(h_i^3|h^4) \\ \Pr(h^4|h^3) = \prod_{i=1}^{K_4} \Pr(h_i^4|h^3), \end{cases} \tag{7}$$

where $\Pr(h_i^3|h^4)$, and $\Pr(h_i^4|h^3)$ are described by sigmoid belief network, that are

$$\begin{cases} \Pr(h_i^3|h^4) = 1/\left(1 + \exp\left(b_i^3 - \sum_{j=1}^{K_4} W_4^{i,j} h_j^4\right)\right) \\ \Pr(h_i^4|h^3) = 1/\left(1 + \exp\left(b_i^4 - \sum_{j=1}^{K_3} W_4^{i,j} h_j^3\right)\right). \end{cases} \tag{8}$$

Finally, the marginal distribution of input data for the deep belief network is written by

$$\Pr(v) = \sum_{h^1} \sum_{h^2} \sum_{h^3} \sum_{h^4} \Pr(v, h^1, h^2, h^3, h^4) \tag{9}$$

Due to the complex model structure with the large number of neurons and multiple hidden layers in the deep belief networks, it is difficult to obtain the weights using the given input data with the maximum likelihood method. In DeepFi, we adopt a greedy learning algorithm using a stack of RBMs to train the deep network in a layer-by-layer manner [13]. This greedy algorithm first estimates the parameters $\{b^0, b^1, W_1\}$ of the first layer RBM to model the input data. Then the parameters $\{b^0, W_1\}$ of the first layer are frozen, and we obtain the samples from the conditional probability $\Pr(h^1|v)$ to train the second layer RBM (i.e., to estimate the parameters $\{b^1, b^2, W_2\}$), and so forth. Finally, we can obtain the parameters $\{b^3, b^4, W_4\}$ of the fourth layer RBM with the above greedy learning algorithm. For the layer i RBM model, we use the CD-1 (CD with 1 step iteration) method to update weights W_i. We first get h^i based on the samples from the conditional probability $\Pr(h^i|h^{i-1})$, and then obtain $\hat{h}^{i-1}$ based on the samples from the conditional probability $\Pr(h^{i-1}|h^i)$, and finally we obtain $\hat{h}^i$ using the samples from the conditional probability $\Pr(h^i|\hat{h}^{i-1})$. Thus, we can update the parameters as follows.

$$\begin{cases} W_i = W_i + \alpha(h^{i-1}h^i - \hat{h}^{i-1}\hat{h}^i) \\ b^i = b^i + \alpha(h^i - \hat{h}^i) \\ b^{i-1} = b^{i-1} + \alpha(h^{i-1} - \hat{h}^{i-1}), \end{cases} \tag{10}$$

where α is the step size.

After the pretraining stage, we need to unroll the deep network to obtain the reconstruction data by using input data with forward propagation. Then the error between input data and the reconstructed data can be used to adjust all the weights in different layers with the back-propagation algorithm. This procedure is called fine-tuning. By minimizing the error, we can obtain the optimal weights to represent fingerprints, which are stored to help improve indoor localization in the on-line stage.

C. Location Estimation based on Data Fusion

After off-line training, we need to test it with positions that are different from those used in the training stage. Because the probabilistic methods have better performance of indoor localization than deterministic ones, we use the probability model based on Bayes' law, which is given by

$$\Pr(L_i|v) = \frac{\Pr(L_i)\Pr(v|L_i)}{\sum_i \Pr(L_i)\Pr(v|L_i)}, \tag{11}$$

where L_i is the i-th reference location, $\Pr(L_i|v)$ is the posteriori probability, $\Pr(L_i)$ is the prior probability that the mobile device is determined to be located at reference location i. In addition, we assume that $\Pr(L_i)$ is uniformly distributed. It follows that

$$\Pr(L_i|v) = \frac{\Pr(v|L_i)}{\sum_i \Pr(v|L_i)}. \tag{12}$$

Based on the deep network model, we define $\Pr(v|L_i)$ as the radial basis function (RBF) in the form of a Gaussian function, which is formulated by

$$\Pr(v|L_i) = \exp\left(-\frac{\|v-\hat{v}\|}{\lambda\sigma}\right), \tag{13}$$

where $\hat{v}$ is the reconstruction input data by using deep learning, σ is the variance of the input data, λ is the coefficient of variation (CV) of the input data. Finally, the position of the mobile device can be estimated as a weighted average of all the reference locations, which is given by

$$\hat{L} = \sum_i \Pr(L_i|v)L_i. \tag{14}$$

We will evaluate the proposed DeepFi system for indoor localization in the following section.

IV. Experiment Validation

A. Experiment Methodology

We perform experiments with DeepFi and examine both the training phase and the test phase. Training locations are equally distributed in the entire room and test points are randomly chosen among the training points. A TL router acts as an AP, while a Dell laptop equipped with an Intel WiFi Link 5300 NIC is used as the mobile device in both training and test phases.

We verify the performance of DeepFi in various scenarios and compare the resulting location errors in different environments. We find that in an open room where there are no outstanding obstacles around the center, the performance of localization is better than that in complex environment where there are few LOS paths. In this section, we present the experimental results from two typical indoor localization environments, as given below.

TABLE I
Mean errors for the Living Room and and Laboratory Experiments

	Living Room		Laboratory	
Algorithm	*Mean error (m)*	*Std. dev. (m)*	*Mean error (m)*	*Std. dev. (m)*
DeepFi	0.9425	0.5630	1.8081	1.3432
FIFS	1.2436	0.5705	2.3304	1.0219
Horus	1.5449	0.7024	2.5996	1.4573
ML	2.1615	1.0416	2.8478	1.5545

1) Living Room in a House: The living room we choose is almost empty, so that most of the measured locations can have LOS receptions. In this 4×7 m^2 room, the AP was placed on the floor, and so do all the training and test points. As shown in the top plot in Fig. 6, 50 points are uniformly scattered with half meter spacing in the room. Because only one AP was utilized in our experiment, the AP placed at one end (rather than the center) of the room to avoid isotropy. We arbitrarily set 12 points in two lines as test points and use the remaining points for training (see the top plot in Fig. 6: the training positions are marked in red and the test positions are marked in green). For each location, we collect CSI data for nearly 500 packet receptions in 60 seconds. We choose a deep network with structure $K_1 = 300, K_2 = 150, K_3 = 100$, and $K_4 = 50$ for the living room environment.

2) Computer Laboratory: The other test scenario is a computer laboratory in Broun Hall in the campus of Auburn University. There are many desks and PCs crowded in the 6×9 m^2 room, which block most of the LOS paths and form a complex radio propagation environment. In this laboratory, 50 training points and 30 test points are selected, as shown in the bottom plot in Fig. 6. The mobile device will also be put at these locations on the floor, with LOS paths blocked by the desks and computers. To obtain integrated characteristics of the subcarriers, CSI information for 1000 packet receptions are collected at each location. We set the deep network with structure $K_1 = 500, K_2 = 300, K_3 = 150$, and $K_4 = 50$ for the laboratory environment.

3) Benchmarks: For comparison purpose, we implemented three existing methods, including FIFS [9], Horus [6], and Maximum Likelihood (ML) [14]. FIFS and Horus are introduced in Section I. In ML, the maximum likelihood probability is used for location estimation with RSS, where only one candidate location is used as the estimation result. For a fair comparison, these schemes use the same measured data as in DeepFi to estimate the location of the mobile device.

B. Localization Performance

In this section, we evaluate the performance of the proposed CSI-based deep learning algorithm with statistical results for the two representative scenarios. The mean and standard deviation of the location errors are presented in Table I. We can see that in the living room experiment, the mean distance error is about 0.95 meter for DeepFi with a single AP. In the computer laboratory scenario, where there exists abundant multipath and shadowing effect, the mean error is about 1.8

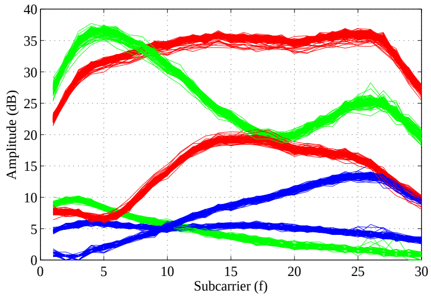

Fig. 7. Amplitudes of channel frequency responses of the three antennas (each is plotted in one color) of 50 packets.

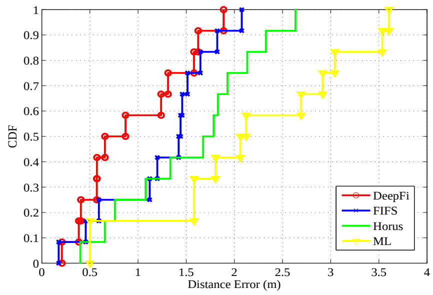

Fig. 8. CDF of localization errors in the living room experiment.

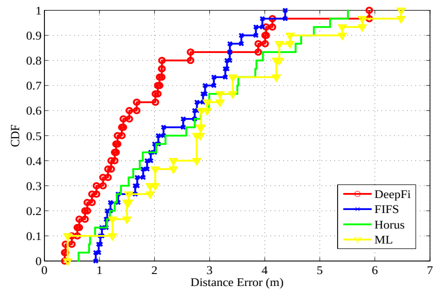

Fig. 9. CDF of localization errors in the laboratory experiment.

meters across 30 test points. DeepFi outperforms FIFS in both scenarios; the latter has a distance error of 1.2 meters in the living room scenario and 2.3 meters in the laboratory scenario. DeepFi achieves a 20% improvement over FIFS, due to the integrated properties of CSI subcarriers from the three antennas. Both CSI fingerprinting schemes, i.e., DeepFi and FIFS, outperform the two RSSI-based fingerprinting schemes, i.e., Horus and ML. The latter achieves an error of at least 2.6 meters in the laboratory experiment.

Fig. 8 presents the cumulative distribution function (CDF) of distance errors with the four methods in the living room experiment. With DeepFi , more than 60% of the test points have an error under 1 meter using a single AP, while FIFS ensures that fewer than 30% of the test points have an error under 1 meter. In addition, most of the test points have distance errors less than 1.5 meters in FIFS, which is similar to DeepFi. On the other hand, both RSSI methods based on coarse information, i.e., Horus and ML, do not perform as well as the CSI-based schemes. There are only 80% of the points have an error under 2 meters.

Fig. 9 plots the CDF of distance errors in the laboratory experiment. In this more complex propagation environment, DeepFi can achieve a 1.7 meters distance error for over 60% of the test points, which is the most accurate among the four method. Because desks obstruct most LOS paths and magnify the multipath effect, the correlation between signal strength and propagation distance is weak. The methods based on propagation properties, i.e., FIFS, Horus, and ML all have degraded performance. In Fig. 9, it is noticed that 70% of the test points have a 3 meters distance error with FIFS and Horus. Unlike FIFS, DeepFi utilizes various CSI subcarriers. It achieves higher accuracy even with just a single AP. It performs well in this NLOS environment because DeepFi shows CSI space property instead of propagation property with the distance.

V. Conclusion

In this paper, we presented DeepFi, a deep learning based indoor fingerprinting scheme that uses CSI information. In DeepFi, CSI information for all the subcarriers from three antennas are collected from accessing the device driver and analyzed with a deep network with four hidden layers. Based on three hypotheses on CSI, we proposed to use the weights in the deep network to represent fingerprints, and incorporated a greedy learning algorithm to train all the weights layer-by-layer to reduce complexity. In addition, a probabilistic data fusion method based on the radial basis function was developed for online location estimation. The proposed DeepFi scheme is validated in two representative indoor environments, and is found to outperform several existing RSSI and CSI based schemes in both experiments.

References

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

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

[3] P. Bahl and V. N. Padmanabhan, “Radar: An in-building RF-based user location and tracking system,” in *Proc. IEEE INFOCOM’00*, Tel Aviv, Israel, Mar. 2000, pp. 775–784.

[4] S. Dayekh, “Cooperative localization in mines using fingerprinting and neural networks,” in *Proc. IEEE WCNC’10*, Sydney, Australia, Apr. 2010, pp. 1–6.

[5] Z. Wu, C. Li, J. Ng, and K. Leung, “Location estimation via support vector regression,” *IEEE Trans. Mobile Comput.*, vol. 6, no. 3, pp. 311–321, Mar. 2007.

[6] M. Youssef and A. Agrawala, “The Horus WLAN location determination system,” in *Proc. ACM MobiSys’05*, Seattle, WA, June 2005, pp. 205–218.

[7] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. Ni, “CSI-based indoor localization,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1300–1309, July 2013.

[8] D. Halperin., W. J. Hu., A. Sheth., and D. Wetherall., “Predictable 802.11 packet delivery from wireless channel measurements,” in *Proc. ACM SIGCOMM’10*, Sept. 2010, pp. 159–170.

[9] J. Xiao, K. Wu., Y. Yi, and L. Ni, “FIFS: Fine-grained indoor fingerprinting system,” in *Proc. IEEE ICCCN’ 12*, Aug. 2012, pp. 1–7.

[10] 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’12*, Jun. 2012, pp. 183–196.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Neural Information and Processing Systems 2012*, Lake Tahoe, NV, Dec. 2012, pp. 1106–1114.

[12] G. Hinton and R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, July 2006.

[13] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” in *Proc. Advances in Neural Information Processing Systems 19 (NIPS’06)*, Cambridge, MA, 2007, pp. 153–160.

[14] M. Brunato and R. Battiti, “Statistical learning theory for location fingerprinting in wireless LANs,” *Elsevier Computer Networks*, vol. 47, no. 6, pp. 825–845, Apr. 2005.