IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 1, 2019, accepted February 24, 2019, date of publication March 7, 2019, date of current version March 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2903487

Deep Learning-Based Indoor Localization Using Received Signal Strength and Channel State Information

CHAUR-HEH HSIEH[1], JEN-YANG CHEN[2], AND BO-HONG NIEN[2]
[1]Information Engineering College, Yango University, Fuzhou 350015, China
[2]Department of Electronic Engineering, Ming Chuan University, Taoyuan 333, Taiwan

Corresponding author: Jen-Yang Chen (jychen@mail.mcu.edu.tw)

This work was supported in part by the Ministry of Science and Technology of Taiwan under Grant MOST 106-2221-E-130 -013-MY2.

ABSTRACT Indoor localization has received wide attention recently due to the potential use of wide range of intelligent services. This paper presents a deep learning-based approach for indoor localization by utilizing transmission channel quality metrics, including received signal strength (RSS) and channel state information (CSI). We partition a rectangular room plane into two-dimensional blocks. Each block is regarded as a class, and we formulate the localization as a classification problem. Using RSS and CSI, we develop four deep neural networks implemented with multi-layer perceptron (MLP) and one-dimensional convolutional neural network (1D-CNN) to estimate the location of a subject in a room. The experimental results indicate that the 1D-CNN using CSI information achieves excellent localization performance with much lower network complexity.

INDEX TERMS Indoor localization, deep learning, convolutional neural network (CNN), received signal strength (RSS), channel state information (CSI).

I. INTRODUCTION

With the rapid growth of artificial intelligence technology, intelligent application has sprung up in many ways, such as indoor navigation, intelligent robots, internet of things (IoT) applications, and smart architectures (such as smart home, smart cities, smart buildings and smart grids). These applications require the location of a user or a device in an indoor environment. In medicine, industry, disaster management, surveillance, and a number of various other fields, user and device localization also has wide-scale applications [1]. Therefore, developing a kind of localization technology specific to indoor environment is becoming more and more important.

Indoor localization is the process of obtaining device or user location in an indoor environment. It can be roughly divided into two categories. One is analyzing the attenuation phenomenon caused by radio waves crossing objects to estimate the location of objects. The common techniques include Time of Flight (ToF), Time Difference of flight (TDoF), Angle of Arrival (AoA), Time of Arrival (ToA), Time Difference of Arrival (TDoA), Received Signal Strength (RSS), and Channel State Information (CSI) [2]–[5]. The other category is to capture images with a video camera (color camera or infrared camera) and use image processing techniques to estimate the location of the objects [6], [7]. The radio-wave and the video-camera localization methods have their own advantages and disadvantages. At present, radio-wave localization seems to have more advantages than video localization does.

With radio-wave approach, indoor environment is often complex, characterized by non-line-of-sight (NLoS) signal propagation, presence of obstacles, signal fluctuation or noise, environmental changes, etc. Obtaining high localization precision is still challengeable. Various techniques and systems of the radio-wave approach have been reviewed in [1], and [8]–[13]. These techniques and systems can be further classified into device-based and device-free methods. The former configures a wireless transmitter in the object to be located, and builds multiple wireless receivers in the environment. The value of the transmission quality index in the wireless receiver changes when the object is placed in different locations. The use of this feature achieves indoor localization. The device-free method analyzes transmission

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Ali Imran.

quality index in a similar manner. The main difference is that it puts the transmitter and the receiver in the environment without wearing the device on the object to be positioned. The device-free method has some advantages over the device-based method. For instance, the user doesn't need to carry the device, which is troublesome for most people. Another advantage is that the device-free method only needs to build one transmitter and one receiver in the environment, and thus costs less than the device-based method. These may explain why the device-free method has received wide attention in recent years.

Fingerprinting is a popular technique in radio-wave indoor localization. It often requires environmental survey to obtain fingerprints of the environment where the localization system is to be used. The fingerprinting technique consists of two phases: offline training and online localization. During the offline phase, a site survey is conducted to collect radio signals from different access points (APs) at many reference points (RPs) of known locations. Thus, each RP is represented by its fingerprint. All the signal vectors form the fingerprints of the site and are stored in a database. In online phase, the real-time measurements match the offline stored fingerprints to estimate the user location using a distance metric [8], [12], [14]. Various radio-wave technologies have been studied and implemented such as Bluetooth, Zigbee, WiFi, and UWB (Ultra Wideband) [12]. Most of current smart phones, laptops and other portable devices are WiFi enabled. Existing WiFi APs can also be used as RPs for signal collection. Therefore, basic localization systems can be built without the need of additional infrastructure. This makes WiFi one of the most widely-studied localization technologies in the literature [1], [12].

In WiFi-based localization, the fingerprints are usually collected in the form of RSS or CSI. For example, the RADAR system [15] first builds a fingerprint map by gathering the WiFi RSS measurements at different locations in the offline training phase, and then estimates the location by minimizing the Euclidean distance between online RSS measurements and fingerprints in the radio map during the online test phase. The work in [16] considers channel interference and the reliability of the RSS measurement and proposes a pre-processing and a post-processing scheme to improve localization accuracy. Reference [17] presents a novel scheme of fingerprint generation, representation and matching, which significantly outperforms existing approaches. The accuracy can be further improved by using CSI since it provides more accurate channel information. The Splicer [18] measures the CSIs from spectrum band and adjusts the errors of amplitude and phase to further improve localization precision. Reference [19] presents a novel method for reducing localization error. It utilizes multidimensional scaling analysis to calculate the Euclidean distance and time-reversal resonating strength between the target point and the reference points and then employs the K Nearest Neighbor (KNN) algorithm to estimate location. In [20], a probabilistic fingerprint-based technique, using principal component analysis to filter the most relevant feature vectors, is suggested to boost the localization efficiency.

In fingerprinting approach, machine-learning methods can be used to extract core features of signals. The features are regarded as fingerprints and stored in a database. This method is highly different from traditional methods that store RSS or CSI directly in the database. Fingerprint approach not only improves localization accuracy but also reduces computational complexity. A number of machine-learning algorithms, including probabilistic methods, KNN, Artificial Neural networks (ANN), Support Vector Machine (SVM), and Random Forest (RF), are widely used to match online measurements to offline measurements [21], [22].

In recent years, deep learning has been widely explored and achieved great success, especially in computer vision. Several indoor fingerprinting systems, based on deep neural networks, have been proposed [23]–[29]. They reported better performance over the conventional methods. Reference [24] presents a deep learning-based indoor fingerprinting algorithm, called DeepFi, which uses CSI amplitudes from all the subcarriers with a deep autoencoder network. DeepFi adopts a greedy learning algorithm using a stack of RBMs (Restricted Boltzmann Machines) to train the deep autoencoder in a layer-by-layer manner. With the trained network parameters of the autoencoder, a data fusion scheme is then utilized to estimate the location. However, the deep autoencoder is only used to extract the features of the CSI signal. A localization method that first leverages calibrated phase information of CSI signals is presented in [25]. A deep neural network similar to [24] is employed in this method. Reference [26] proposes an autoencoder based Deep Extreme Learning Machine indoor localization method, which utilizes the high level extracted features by autoencoder from collected data by smartphones with internal sensors. Chen *et al.* [27] propose the convolutional neural network (CNN)-based localization algorithm using CSI. The CSI amplitude is organized into a time-frequency matrix, which resembled the image. A two-dimensional convolution neural network (2-D CNN) is then utilized for localization. Unlike [24], the network utilizes end-to-end deep learning. Wang *et al.* [28], [29] propose deep convolutional neural network based scheme for indoor localization. The scheme transforms the AoA data into CSI image and utilizes 2-D CNN to improve localization performance as well. Several more recent CNN-based works adopt 2-D CNN for localization as well [30], [31]. They convert 1-D time-series signal into 2-D matrix in different manners. For example, [30] combines magnetic field and WiFi signal into positioning image. In [31], the RSS signals obtained with K APs and N unique locations are formed into fingerprint images of size $N \times K$.

The methods that convert a 1-D signal to 2-D image lead to two drawbacks: (a) an increase in network complexity, and (b) an overhead of conversion; both increase computational burden of the system. To attack this problem, this paper explores a deep neural network for indoor localization that processes 1-D signals directly. By partitioning a rectangular

room plane into two-dimensional blocks, we formulate the localization as a classification problem. RSS and CSI signals are respectively applied to two deep neural networks, including MLP and 1-D CNN. As a result, four deep neural networks, including MLP-RSS, MLP-CSI, CNN-RSS and CNN-CSI, are developed. The localization performances of the four networks are compared and analyzed by their prediction accuracy and network complexity. A novel localization error metric, referred to as localization error probability, is also proposed to evaluate the localization accuracy. The contributions of this work are summarized as follows:

1) We investigate various end-to-end deep learning methods using RSS and CSI signals for device-free indoor localization, and evaluate their performance in terms of localization accuracy and network complexity.
2) To our best knowledge, this work is the first to propose 1-D CNN for device-free indoor localization, which is proved to achieve comparable performance with much lower network complexity, as compared to the existing 2-D CNN.
3) A novel localization error probability metric is presented to evaluate the localization precision.

The remainder of this paper is organized as follows. Section II gives a preliminary on the basics of RSS and CSI. In Section III, the proposed method is described in details; we first describe how to collect packet data and parsed data to obtain RSS and CSI features; based on two types of features, we design MLP and 1D-CNN deep neural networks separately. The experimental process, selection of network architectures, performance comparison, and localization error analyses are described in Section IV. Finally, the conclusion is drawn in Section V.

II. PRELIMINARY

A. RECEIVED SIGNAL STRENGTH (RSS)

RSS is the measurement of power present in a received radio signal. It depends on the transmitted power and the distance between the transmitter and receiver [32] as

$$RSS(dbm) = 10 \times log \frac{P_r}{P_{ref}}, \tag{1}$$

where P_{ref} is the reference power, and P_r is the received power calculated as

$$P_r = P_t \times G_t \times G_r \times (\frac{\lambda}{4\pi d})^2, \tag{2}$$

where P_t is the transmitted power, G_t denotes the transmitter antenna gain, G_r is the receiver antenna gain, λ is the wavelength of the radio wave, and d represents the distance between the transmitter and receiver.

The above equation proves that the RSS is inversely proportional to squared distance. However, in actual use, it can be affected by many factors, such as multipath effects and diffraction phenomena. Consequently, the RSS signal depicts highly nonlinear and uncertain relation with the distance in a residential room. Thus, modeling the relation of RSS signals with locations is very challenging. However, due to its simplicity and easy implementation, RSS is still popular in indoor localization.

B. CHENNEL STATE INFORMATION (CSI)

Recently, the Multiple Input Multiple Output (MIMO) system has been widely applied in wireless communications for raising the transmission efficiency. It is a method for multiplying the capacity of a radio link using multiple transmitter and receiver antennas to exploit multipath propagation. In MIMO antenna configuration, a high-rate signal is split into multiple lower-rate streams, and each stream is transmitted from a different transmit antenna in the same frequency channel. The MIMO is typically combined with orthogonal frequency-division multiplexing (OFDM) modulation, which is a digital multi-subcarrier modulation method. The CSI obtained at the receiver, is often used to measure the quality of the channel.

CSI describes how a signal propagates from the transmitter to the receiver and represents the combination effect of, for example, scattering, fading, and power decay with distance [10]. The quantitative analysis of signal propagation behavior within a WiFi-covered area can identify different types of disturbances. Thus, CSI has a variety of applications, such as detecting human motion or identifying the location of a subject.

Most radio wave transmission processes are susceptible to multipath effects. Therefore, the received signal is a superposition of multiple path signals. The CSI distinguishes multipath signals by means of Channel Impulse Response (CIR). In a non-time-varying system, CIR is represented as

$$h(\tau) = \sum_{i=1}^{N} a_i e^{-j\theta_i} \delta(\tau - \tau_i) + n(\tau), \tag{3}$$

where a_i, θ_i, τ_i represent the amplitude, phase angle and time delay of the ith transmission path, respectively, N is the total number of paths, $n(\tau)$ is Gaussian noise, and δ is an impulse function. However, in general commercial equipment, CIR cannot be accurately obtained. To overcome this limitation, the channel frequency response (CFR) is often used to model the channel, which consists of amplitude response and phase response in frequency domain. CFR can be obtained by Fast Fourier Transform (FFT), i.e., $H = FFT(h(\tau))$.

Let $x(f_k, t)$ and $y(f_k, t)$ be the frequency domain representations of transmitted and received signals, respectively, with carrier frequency f_k. The two signals are related by

$$y(f_k, t) = H(f_k, t) \times x(f_k, t), \tag{4}$$

where $H(f_k, t)$ is the complex valued CFR for carrier frequency f_k measured at time t. CSI measurements contain these CFR values. The frequency response of the kth channel can be expressed as

$$H(f_k) = |H(f_k)| \, exp\{j \angle H(f_k)\}, \tag{5}$$

where $|H(f_k)|$ is the amplitude response of the kth channel at frequency f_k, and $\angle H(f_k)$ is its phase response.

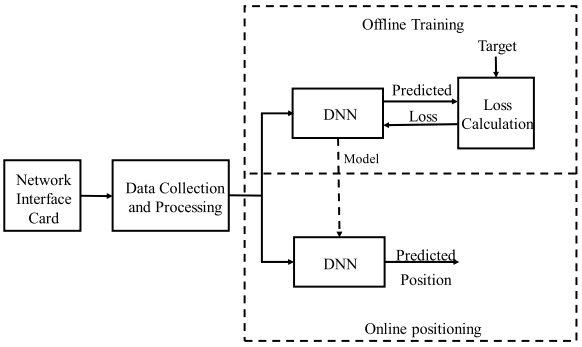

FIGURE 1. Proposed positioning system architecture.

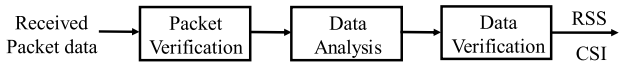

FIGURE 2. Data parsing process.

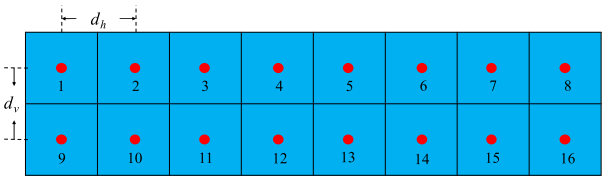

FIGURE 3. Layout of the quantized locations.

III. PROPOSED METHOD

The localization system proposed in this paper collects the RSS and CSI between the wireless AP and the network interface card to estimate the current location of a person. Our system consists of two phases, offline training and online positioning, as shown in Fig.1. During the offline training phase, the system captures the fingerprints (features) of RSS and CSI to train a proposed deep neural network (DNN) in order to obtain a positioning model. During the online positioning, the system receives real-time data from a network interface card and then predicts where the person is located. We also design an efficient packet data parsing tool to extract CSI and RSS from the original packet data received.

A. DATA COLLECTION AND PROCESSING OF RSS AND CSI

To collect RSS and CSI data, we adopt a notebook computer that installs an Intel 5300 network interface card with three receiving antennas and a wireless network router with two antennas. RSS can be collected through a general network interface card. However, for the CSI at the physical layer, in addition to hardware support, it requires software tool to help. The most stable CSI collection tool currently known is the Linux 802.11n CSI Tool [33]. The tool provides drivers and collection library that allow us to collect CSI through the firmware of the Intel 5300. In addition, it provides a series of Matlab-based analysis tools to help researchers develop CSI applications. In this work, we develop an efficient packet data-parsing tool written in C language, which strengthens Linux 802.11n CSI Tool. It is fully automatic and thus lowers labor costs for data collection.

The raw binary data received by the wireless network receiver contains various information of the transmission process, such as the numbers of transmitter antennas and receiver antennas, the CSI, and the RSS. However, to obtain this information from the original data needs certain analysis process. The developed packet data-parsing tool aims to analyze and parse the original data file. The parsing process of this tool consists of packet verification, data analysis, and data verification, as shown in Fig. 2. If the RSS or CSI is lost during the transmission process, it is unable to check from the packet integrity. Thus, the data verification step is required to confirm whether the RSS or CSI is complete. With the parsing procedure, the parsed data is ensured whether it is complete. The developed packet data parsing tool is very computationally efficient and can improve the overall localization performance.

As stated above, the network receiver contains three antennas. Therefore, a received packet contains only three RSS data. Obviously, a single packet in one position is not enough to consider the environmental changes. To solve the problem, we concatenate 15 packets into one training /testing sample. Thus, an RSS sample vector is expressed as

$$\text{RSS} = (|rss_{1,1}|, |rss_{1,2}|, |rss_{1,3}|, |rss_{2,1}|, \ldots, |rss_{15,3}|), \quad (6)$$

where $rss_{i,j}$ denotes the received RSSs for the ith packet from the jth antenna.

The number of CSIs in a packet is the product of the transmitter antenna, the receiver antenna, and the number of channels (sub-carriers). The Intel 5300 network interface card used in this work has 30 channels. Thus, the total number of CSI data per packet is $3 \times 2 \times 30 = 180$. Using Intel 5300 wireless network, we can obtain CFR containing 30 frequency responses defined in (5).

In our work, only amplitude response is utilized for indoor localization. For simplicity, we concatenate the 180 data into a CSI sample vector and denoted as

$$\text{CSI} = (|csi_{1,1,1}|, |csi_{1,1,2}|, \ldots, |csi_{1,1,30}|, \times |csi_{1,2,1}|, \ldots, |csi_{3,2,30}|). \quad (7)$$

B. DEEP NEURAL NETWORK DESIGN

We partition the space of a rectangular room into $M \times N$ (2×8 in our experiment) blocks, as shown in Fig. 3. Each block denotes a quantized location and represents a class. The localization problem therefore becomes a classification one with $M \times N$ classes. The size of the block determines the localization resolution as well as the system complexity. The smaller the block size, the higher the resolution. However, due to the increase of the number of classes, the system complexity will become higher. In this work, we adopt the MLP and 1-D CNN for the supervised classification problem, which are described in the following paragraphs.

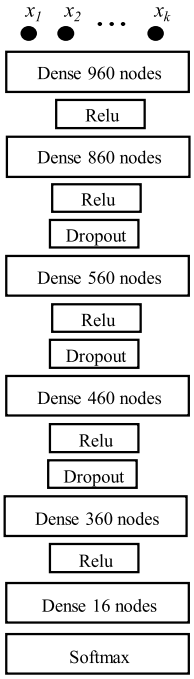

FIGURE 4. Typical architecture of proposed MLP using RSS and CSI.

1) MULTI-LAYER PERCEPTRON

Fig. 4 shows a typical architecture of the proposed MLP using RSS or CSI. In our typical indoor-localization design, 16 locations are preset in a room. The size of the output layer K is thus 16. The size of the input layer depends upon the dimension of RSS or CSI signal, which is 45 for RSS and 180 for CSI. In this architecture, several configurations are implemented and evaluated, which will be discussed in the next section.

The MLP model is obtained by training with the well-known stochastic gradient descent (SGD) algorithm [28]. The SGD employs the backpropagation (BP) scheme which calculates the loss (error) function between the network prediction and the ground truth, and back propagates the error of each layer from output to input, and updates the network parameters θ (weights and biases) iteratively.

To avoid vanishing gradient problem, the ReLu function is adopted as the activation function in the hidden layers. However, the softmax activation function, as defined in (8), is employed in the output layer, which maps the real-value input into prediction probability in the range of [0,1] by

$$\sigma\left(z_j\right) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, \quad j = 1, \ldots K. \tag{8}$$

In this network, we choose cross-entropy as loss function. Furthermore, we adopt L_2 regularization to avoid overfitting [34]. Thus, the learning of DNN is to minimize the loss function $L(t, y, \theta)$ defined in (9) with respect to network parameter set θ, where the second term is for regularization.

$$L(t, y, \theta) = -\sum_{s=1}^{K} t_s log y_s + \lambda \sum_i w_i^2, \tag{9}$$

where t and y denote the target output and the predicted output vectors of the MLP. The cross-entropy indicates the difference between the amount of information contained in t and the amount of information contained in y. If the value of the cross-entropy is small, then the predicted output is close to the target output.

In BP training, the modified gradient descent method, named adaptive moment estimation (*adam*), is employed to update the parameter set of the MLP by [35]

$$\theta_t \leftarrow \theta_{t-1} - \frac{\eta \hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}, \tag{10}$$

where η is a fixed learning step size; ε is a very small constant; $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected first moment estimate and the biased-corrected second moment estimate, which are calculated by

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}, \tag{11}$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \tag{12}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \tag{13}$$

where g_t is the gradient of lost function at time t; β_1 and β_2 are the attenuated constants for the first moment and second moment, respectively.

In addition, the dropout is also employed between the two hidden layers to avoid overfitting [34].

2) 1-D CONVOLUTIONAL NEURAL NETWORK

In general, the CNN is composed of a number of convolutional layers for feature extraction, where each layer is usually followed by a pooling layer which is also followed by one or more fully connected layers for classification (or regression). 2D-CNN has been widely used for image feature extraction and classification problems in the literature [36]–[41].

A typical architecture of the proposed 1-D CNN for indoor localization is shown in Fig.5. We use the convolution blocks to extract hierarchical features from low level to high level. Each convolution block consists of 1-D convolution, batch normalization, and activation function. It is noted that the polling layer is not utilized here as it degrades the performance in our experiment. Batch normalization is applied after convolution and before activation. It provides any layer in a convolution block with inputs that are zero mean/unit variance, which improves the performance and the stability of deep neural networks [42].

The fully connected layer (dense layer) is aimed to construct a classifier based on the input hierarchical features. The number of neurons in this layer is 16. The ReLu function is utilized as activation function in all blocks except the output layer. The SGD algorithm and the cross-entropy loss function are utilized in learning the CNN model. The learning procedure is similar to that of the MLP stated before, and hence the description is omitted here. The optimal kernel size and the feature maps are obtained in Section IV.

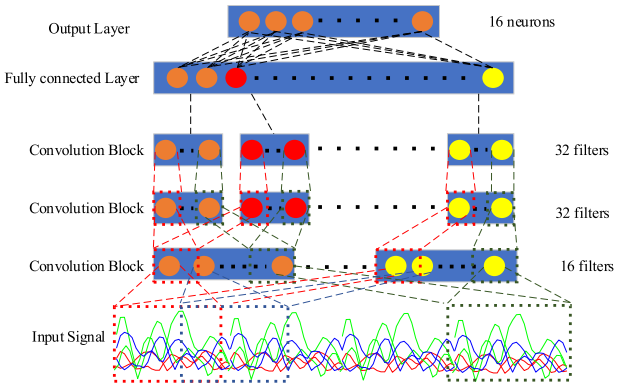

FIGURE 5. Overall structure of our proposed 1-D CNN.

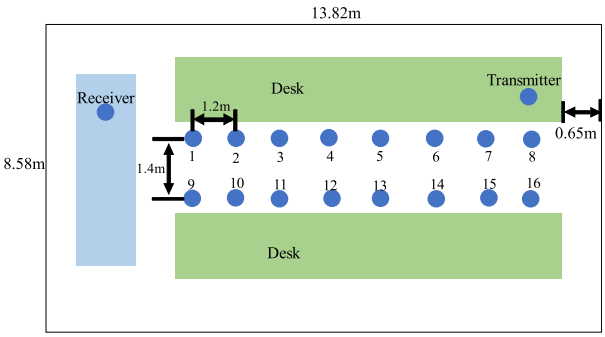

FIGURE 6. Experimental environment map.

IV. NUMERICAL ANALYSIS

A. EXPERIMENTAL ENVIRONMENT AND PROCEDURE

The localization performance is generally affected by the complexity of the indoor environment. To evaluate the robustness of the proposed methods, we choose a room with many furniture and obstacles in order to increase the complexity of the environment. The room is a closed space with a length of 13.82 meters and a width of 8.58 meters, as shown in Fig.6. A rectangular area between two desks of the room is uniformly divided into 2×8 blocks. The center of each block is the reference point (or training/testing point), as denoted by the blue circle in Fig. 6. The distance between two horizontal adjacent points is 1.2 m, and the distance between two vertical adjacent points is 1.4 m. The router (transmitter) is located on the front desktop, and the receiver is located on the rear desktop. Both are one meter above the ground.

A personal computer (PC) equipped with an Intel 5300 network interface card is used as a receiver, and a TP-Link WA901-D wireless network router is adopted as a transmitter. The PC for training the DNN is equipped with i7-3770 CPU, a memory of 20GB and a GTX1060 display card.

In order to increase the robustness of the system, three persons with different body shapes are chosen for evaluation. Each person stands at the training point for one minute. The network receiver captures packets, and the developed packet data-parsing tool stated previously extracts RSS and CSI data for training and testing. The total number of valid samples of

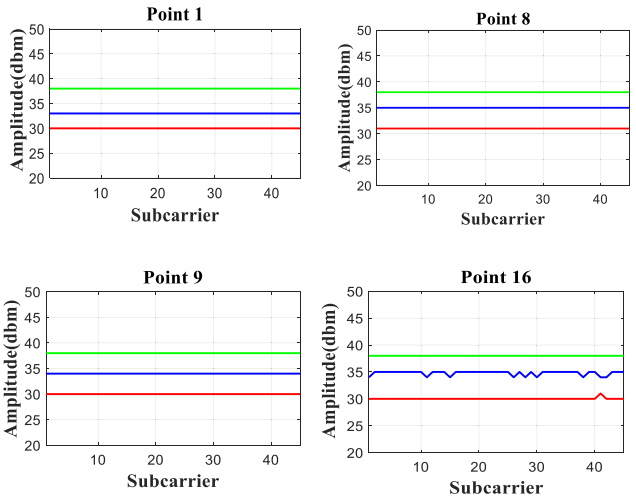

FIGURE 7. RSS waveforms measured by the same subject at different reference points.

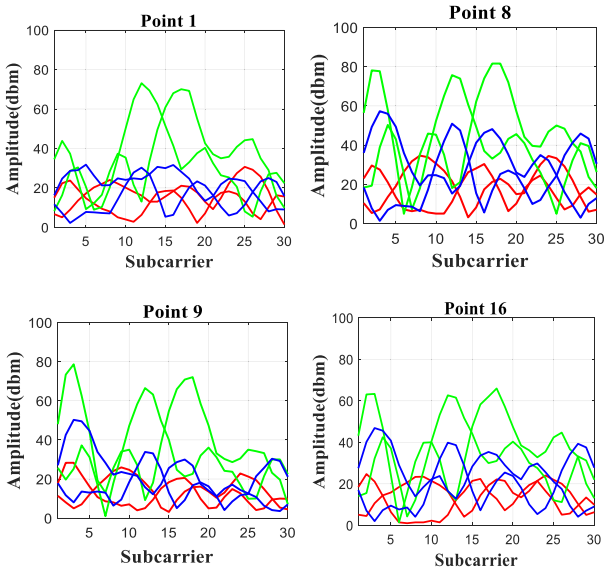

FIGURE 8. CSI waveforms measured by the same subject at different reference points.

RSS and CSI are 16,753 and 251,388, respectively. For both RSS and CSI, 90% of the total samples are randomly chosen for training, and 10% for testing.

B. MEASUREMENT OF RSS AND CSI SIGNALS

Fig.7 and Fig. 8 show the signal power waveforms of the RSS sample and the CSI sample, respectively; different colors in the figures indicate signals measured by different receiving antennas. It is difficult to observe RSS since each packet contains only three records of data. Therefore, we concatenate 15 packets into a waveform for easier observation. The CSI and RSS waveforms change when the person stands at different points, as shown in the figures. However, the changes generated by RSS are relatively small, which means that RSS is less sensitive to environmental changes than CSI.

C. SELECTION OF DNN ARCHITECTURES

In this work we use two different signals (RSS and CSI) and two deep neural networks (MLP and 1-D-CNN) to establish four different indoor localization methods: MLP-RSS, MLP-CSI, CNN-RSS, and CNN-CSI. For comparing the networks, two metrics, including prediction accuracy and total number of parameters, are adopted. The second metric reveals the complexity of a network. The higher the complexity, the larger the computational load is required. We apply the popular deep learning platform *Keras* to evaluate the performance of all deep learning architectures discussed in this paper. The total number of parameters required in an architecture can be easily calculated with *Keras*.

The design of architectures of deep neural networks includes settings of a large variety of hyper-parameters, such as the number of hidden layers, nodes (neurons) of every layer, number of channels, learning rate, etc. Utilizing the grid search [34], [43] and applying our experience, we aim to obtain better architectures through the process of trial and error.

Eight MLP architectures are designed in our experiment. The first architecture utilizes four hidden layers with different number of neurons in each layer. The next five architectures have five hidden layers with different numbers of neurons in each layer. The last two architectures have 6, and 7 layers. We also design eight 1-D CNN architectures. The first architecture uses hidden convolution blocks with kernel size of 26. The next four architectures use three hidden convolution blocks with different kernel sizes and different number of filters. The last three architectures employ four, five and six hidden convolution blocks. For comparing those architectures, two metrics, including prediction accuracy and total number of parameters, are adopted. The second metric reveals the complexity of a network. The higher the complexity, the larger the computational load is required. We aim to select the best architecture from the eight MLPs and eight CNNs stated above using the two metrics.

Table 1 lists the performance comparison of the eight MLP architectures using RSS signal. For an MLP, if the numbers of nodes of all layers are the same, we denote the network architecture as $m \times k$, which represents that there are k layers and every layer has m neurons in the network. For example, 360×5 denotes a network with 5 layers, and each layer has 360 neurons. On the contrary, if the numbers of nodes of all layers are not exactly the same, the particular number of nodes of a layer is specified. For instance, 960-860-560-460-360 represents a network in which the first layer has 960 nodes, the second layer 860 nodes, and so on.

The *6*th architecture in this table, 960-860-560-460-360, achieves similar prediction accuracy to the third architecture, but it has much less network complexity. Thus, we choose the *6*th architecture. Table 2 shows the comparison of the eight MLP architectures using CSI signal. Again, the *6*th architecture is chosen since it performs the best with lower network complexity. Note that when the MLP exceeds 5 layers, the prediction accuracy drops sharply, which may be caused by overfitting.

TABLE 1. Comparison of MLP architectures using RSS signal.

MLP Architectures	Total Number of Parameters	Prediction Accuracy(%)
360 x 4	412577	67.94
360 x 5	542176	78.26
720 x 5	2121136	81.25
960 x 5	3749776	80.53
720,620,520,420,320	1161736	77.37
960,860,560,460,360	1782576	81.07
960,860,560,460,360,360	1912897	6.92
960,860,560,460,360,360,360	2042857	9.07

TABLE 2. Comparison of MLP architectures using CSI signal.

MLP Architectures	Total Number of Parameters	Prediction Accuracy(%)
360 X 4	460816	99.43
360 X 5	590776	99.54
720 X 5	2218336	99.80
960 X 5	3879376	99.61
720,620,520,420,320	1258936	99.83
960,860,560,460,360	1912176	99.93
960,860,560,460,360,360	2042136	99.68
960,860,560,460,360,360,360	2172096	99.61

Table 3 and Table 4 list the performance comparison of 1-D CNN architectures using RSS and CSI, respectively. As shown in Table 3, the first architecture uses two hidden convolution blocks (16-32) with kernel size of 26. The next four architectures utilize three hidden convolution blocks (16-32-32) with different kernel sizes and different number of filters. The last three architectures employ four, five and six hidden convolution blocks. The architecture 16-32-32 with kernel size of 16 performs the best for RSS signal. As shown in Table 4, all architectures obtain approximately the same prediction accuracy, which is excellent. The architecture 16-32-32 with kernel size of 26 is selected since it has the best performance with relatively lower network complexity.

Taking both the prediction accuracy and network complexity into account, the best architectures selected are as follows:

MLP network: the architecture 960-860-560-460-360 is selected for both RSS and CSI.

CNN network: the architecture 16-32-32 with kernel size of 16 is selected for RSS, and the same architecture with kernel size 26 is selected for CSI.

We use the best architectures above for performance analysis and comparison. The prediction accuracy of each reference point is shown in Table 5. Both MLP and CNN using CSI feature achieve prediction accuracy over 99% at all

TABLE 3. Comparison of CNN architectures using RSS signal.

Kernel sizes	Number of filters	Total Number of Parameters	Prediction Accuracy(%)
26	16-32	38465	70.68
10	16-32-32	38976	80.65
16	16-32-32	48288	82.92
26	16-32-32	63808	82.32
36	16-32-32	79328	79.94
26	16-32-32-32	92033	63.46
26	16-32-32-32-32	118817	67.10
26	16-32-32-32-32-32	145601	66.38

TABLE 4. Comparison of CNN architectures using CSI signal.

Kernel sizes	Number of filters	Total Number of Parameters	Prediction Accuracy(%)
26	16-32	106144	99.04
10	16-32-32	108096	99.79
16	16-32-32	117408	99.93
26	16-32-32	132928	99.97
36	16-32-32	148448	99.97
26	16-32-32-32	159712	99.96
26	16-32-32-32-32	196496	99.90
26	16-32-32-32-32-32	213280	99.82

TABLE 5. Comparison of prediction accuracy.

Prediction Accuracy (%)		Methods				
		MLP-RSS	MLP-CSI	CNN-RSS	CNN-CSI	2DCNN-CSI
Reference points	1	85.39	100.00	90.57	100	100
	2	96.58	100.00	93.41	100	90
	3	99.12	99.94	96.77	100	100
	4	94.06	100.00	95.45	99.93	100
	5	87.31	100.00	94.34	99.88	100
	6	86.24	99.89	91.43	100	100
	7	65.79	99.69	51.69	99.84	100
	8	61.00	99.76	59.63	99.81	100
	9	81.60	100.00	80.67	100	90
	10	84.00	100.00	90.29	100	100
	11	75.79	99.93	82.00	100	100
	12	91.15	100.00	85.37	100	100
	13	78.10	100.00	87.18	99.88	100
	14	67.71	99.79	67.86	99.93	100
	15	62.14	99.81	64.00	99.62	100
	16	68.69	100.00	86.52	100	100

reference points. However, the accuracy can be as low as 51.69 % when RSS feature is used. Table 6 shows that the average prediction accuracy using CSI is significantly higher than that using RSS, basing on either MLP or CNN network. Table 6 also shows the variation of accuracy over different reference points using RSS is significant with standard deviation more than 10%. On the contrary, the standard deviation is only 0.11% for MLP and 0.05% for CNN using CSI. This indicates that the prediction stability using CSI signal is much better than using RSS signal. This is because CSI provides fine-grained physical layer (PHY) information [44] and therefore achieves better performance, as kexpected.

As to MLP-CSI (or CNN-CSI), the accuracy is higher when the reference point is close to the wireless network receiver. When the reference point moves further from the receiver, the accuracy reduced, as expected. However, to MLP-RSS (or CNN-RSS), the relationship between accuracy and distance seems to be random. This implies that CSI signal is less affected by multipath. Therefore, it shows better stability than RSS.

As mentioned above, we use the mean and standard deviation of the prediction accuracy to evaluate the localization accuracy and localization stability. Moreover, we utilize the total number of parameters of the deep neural networks to evaluate the system complexity. Table 6 indicates that based on CSI signal, either MLP or CNN can achieve excellent prediction accuracy with very small deviation. However, the total number of parameters required in CNN is less than 1/14 of the total number of parameters required in MLP. Therefore, we conclude that CNN-CSI performs the best in terms of localization accuracy, localization stability and system complexity.

To prove the effectiveness of the 1-D CNN, we implement ConFi network [27], which is based on 2-D CNN network using CSI signal and is state-of-the-art of the CNN-based approaches. ConFi is a five-layer network that mainly includes three 2D convolution layers, and two fully connected (dense) layers. The details can be found in [27]. We evaluate our proposed networks and ConFi, and the results are shown in Tables 5, 6 and 8. It is seen from Table 6 that CNN-CSI achieves slightly better prediction accuracy (both average and deviation) than ConFi. However, the total number of network parameters of ConFi is approximately 690 times that of CNN-CSI. Obviously, the proposed 1-D CNN-CSI reduces the network complexity significantly without sacrificing prediction accuracy. In addition, our 1-D CNN reduces the overhead of 2-D CNN that requires converting time-series data into 2D images.

TABLE 6. Performance summary of various methods.

	Mean of Prediction Accuracy (%)	Deviation of Prediction Accuracy (%)	Total No. of Parameters
MLP-RSS	80.29	12.34	1,782,576
MLP-CSI	99.93	0.11	1,912,176
CNN-RSS	82.32	13.94	48,288
CNN-CSI	99.98	0.05	132928
2DCNN-CSI	98.75	3.42	81,158,836

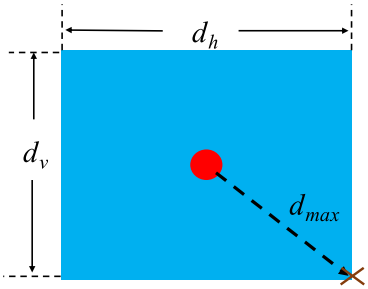

FIGURE 9. Localization error for correct prediction. •: reference point, x: estimate.

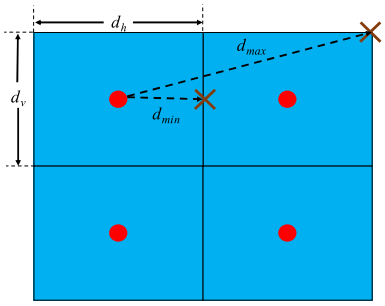

FIGURE 10. The minimal and maximal localization errors for horizontal error. •: reference point, x: estimate.

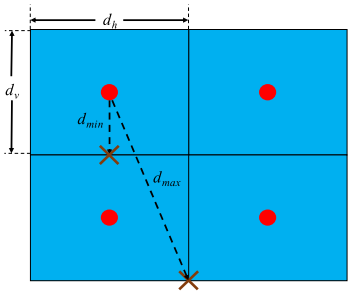

FIGURE 11. The minimal and maximal localization errors for vertical error case. •: reference point, x: estimate.

D. LOCALIZATION ERROR ANALYSIS

This work formulates the localization as a classification problem and utilizes the prediction accuracy or prediction error rate (1-prediction accuracy) to evaluate the localization performance. As stated previously, we partition the space into 2×8 blocks. If the target is at one block (class) and the predicted is at another block (class), the localization system will cause a misidentification, and the localization error is proportional to the distance between the target block and the predicted one. In this subsection, we aim to convert the prediction error rate into the localization error, which will achieve the requirements of the localization task.

The localization error is the distance between the position of the reference point and the estimated position. In our experiments, the total number of blocks is 16. For any target block, the possible predicted block is from 1 to 16. Therefore, by calculating the distance between the reference point in the target block i and any position at the predicted block j, $[d_{ij}]$, we can further obtain the distance matrix with 16×16 size. Refer to Fig. 3, the distance matrix can be calculated in the situations as follows.

1) LOCALIZATION ERROR FOR CORRECT PREDICTION

Correct prediction means that the predicted block (class) is the target block (class). In this case, the minimal localization error is 0, and the maximal localization error is $\sqrt{(0.5d_h)^2 + (0.5d_v)^2}$, as illustrated in Fig. 9.

2) LOCALIZATION ERROR FOR INCORRECT PREDICTION

Incorrect prediction means that the predicted block is not the target block. Three situations exist in this case.

a: HORIZONTAL ERROR

Assume the reference point is at the center of the block i, the estimated position will be at any position of the other block j in horizontal direction. Refer to Fig. 10, it is seen that the minimal localization error exists where the estimated location is in the middle of the left boundary of the block j. In addition, the maximal localization error exists where the estimated is at the top-right corner or bottom-right corner of the block j. We can calculate the minimal and maximal error as

$$d_{min} = d_h(|j - i| - 0.5), \tag{14}$$

$$d_{max} = \sqrt{((|j - i| + 0.5)d_h)^2 + (0.5d_v)^2}, \tag{15}$$

where $i, j = 1, 2, \ldots, 8$, d_h is the distance between two horizontal neighbors, and d_v is the distance between two vertical neighbors. In our implementation, $d_h = 1.2m$ and $d_v = 1.4m$.

b: VERTICAL ERROR

The reference point is assumed at the center of the block i, and the estimated location will be at any position of the other block j in vertical direction. Referring to Fig. 11, it is easy to calculate the minimal and maximal error as

$$d_{min} = d_h(|j - i| - 7.5), \tag{16}$$

$$d_{max} = \sqrt{((|j - i| - 7.5)d_h)^2 + ((|j - i| - 6.5)d_v)^2}, \tag{17}$$

where $i = 1, 2, \ldots, 8; j = i + 8$.

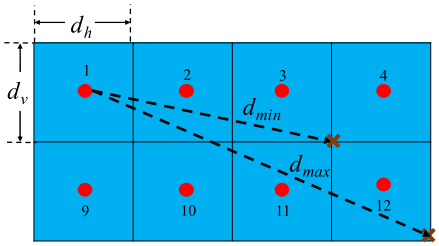

FIGURE 12. The minimal and maximal localization errors for oblique error case. •: reference point, x: estimate.

TABLE 7. 8 × 8 submatrix of the distance matrix.

	1	2	3	4	5	6	7	8
1	[0.00,0 .92)	[0.60,1 .92)	[1.80,3 .08)	[3.00.4 .26)	[4.20.5 .45)	[5.40,6 .64)	[6.60,7 .83)	[7.80,9 .03)
2	[0.60,1 .92)	[0.00,0 .92)	[0.60,1 .92)	[1.80,3 .08)	[3.00.4 .26)	[4.20.5 .45)	[5.40,6 .64)	[6.60,7 .83)
3	[1.80,3 .08)	[0.60,1 .92)	[0.00,0 .92)	[0.60,1 .92)	[1.80,3 .08)	[3.00.4 .26)	[4.20.5 .45)	[5.40,6 .64)
4	[3.00.4 .26)	[1.80,3 .08)	[0.60,1 .92)	[0.00,0 .92)	[0.60,1 .92)	[1.80,3 .08)	[3.00.4 .26)	[4.20.5 .45)
5	[4.20.5 .45)	[3.00.4 .26)	[1.80,3 .08)	[0.60,1 .92)	[0.00,0 .92)	[0.60,1 .92)	[1.80,3 .08)	[3.00.4 .26)
6	[5.40,6 .64)	[4.20.5 .45)	[3.00.4 .26)	[1.80,3 .08)	[0.60,1 .92)	[0.00,0 .92)	[0.60,1 .92)	[1.80,3 .08)
7	[6.60,7 .83)	[5.40,6 .64)	[4.20.5 .45)	[3.00.4 .26)	[1.80,3 .08)	[0.60,1 .92)	[0.00,0 .92)	[0.60,1 .92)
8	[7.80,9 .03)	[6.60,7 .83)	[5.40,6 .64)	[4.20.5 .45)	[3.00.4 .26)	[1.80,3 .08)	[0.60,1 .92)	[0.00,0 .92)

c: OBLIQUE ERROR

In this case, the estimated position will be at any position of the other block j in oblique direction of the reference point. Referring to an example in Fig. 12, we can calculate the minimal and maximal error as

$$d_{min} = \sqrt{((|j-i|-8-0.5)d_h)^2 + (0.5d_v)^2}, \quad (18)$$

$$d_{max} = \sqrt{((|j-i|-8+0.5)d_h)^2 + (0.5d_v)^2}, \quad (19)$$

where $i = 1, 2, \ldots, 8; j = i + 8$.

With the above procedure, we can obtain a 16×16 distance matrix $[d_{ij}]$ in which each entry has two values, minimal localization error d_{min} and maximal localization error d_{max}. Due to space limitation, we crop an 8×8 submatrix from an example of the distance matrix, as displayed in Table 7.

Since some elements in the distance matrix $[d_{ij}]$ are the same value, e.g., $d_{13} = d_{42}$, those elements should be combined into one distance. From the predicted results of the deep neural networks stated before, we can calculate the confusion matrix $[p_{ij}]$, which represents the error rate that the deep network misclassifies the class i into the class j. Combining the distance matrix with the confusion matrix, we can obtain localization error probability $p(d)$, where d is the localization error distance in meter. Table 8 shows the localization error probability for the best architectures selected from the four methods mentioned above and the existing 2DCNN [27], respectively.

TABLE 8. Comparison of localization error probability of various methods.

Localization Error Probability(%)		Methods				
		MLP-RSS	MLP-CSI	CNN-RSS	CNN-CSI	2DCNN-CSI
Error Range (m)	[0.00, 0.92)	81.07	99.93	82.92	99.97	98.75
	[0.60, 1.9 2)	3.82	0.05	2.93	0.028	0.625
	[0.70, 2.18)	0.48	0	0.77	0	0
	[0.92, 2.77)	1.13	0	1.37	0	0
	[1.80, 3.08)	2.63	0.012	2.39	0	0
	[1.93, 3.66)	0.78	0	1.97	0	0
	[3.00, 4.26)	0.9	0.004	0.9	0	0
	[3.08, 4.70)	1.85	0.004	0.84	0	0.625
	[4.20, 5.45)	2.93	0	2.32	0	0
	[4.26, 5.80)	1.25	0	1.01	0	0
	[5.40, 6.64)	0.41	0	0	0	0
	[5.45, 6.93)	0.95	0	1.73	0	0
	[6.60, 7.83)	0.36	0	0.12	0	0
	[6.64, 8.08)	0.6	0	0.54	0	0
	[7.80, 9.03)	0	0	0	0	0
	[7.83, 9.24)	0.12	0	0.12	0	0

It indicates that CNN-CSI performs the best with results in maximal localization error of 0.92 m with probability of 99.97%. It also shows that the localization error will be no larger than 1.92 m. The worst is MLP-RSS, which achieves maximal localization error of 0.92 m with probability of 81.07%; however, it possibly yields localization error up to 9 m. Besides, this table also demonstrates that the proposed 1-D CNN performs slightly better than the 2-DCNN method [27].

V. CONCLUSION

This paper has proposed deep learning-based methods for device-free indoor localization based on MLP and 1-D CNN. We used a wireless network router as the transmitter, and a network interface card as the receiver to collect packet data. We designed an efficient data packet-parsing tool to obtain

RSS and CSI signals. Based on the received two signals, four deep neural networks, MLP-RSS, MLP-CSI, CNN-RSS and CNN-CSI, were developed to estimate the location of a person in an indoor environment. It has been found that through experiments that the localization methods using CSI signals achieve much better localization accuracy than those using RSS signal. Moreover, the 1-D CNN-CSI network achieves excellent localization performance and much less network complexity compared with the other networks including the existing 2-D CNN-CSI.

However, the work involved only amplitude response of CSI signals. Since the phase response may provide useful information for indoor localization, the combination of amplitude and phase information will be investigated in the future study.

VI. ACKNOWLEDGMENT

The authors would like to express their sincere thanks to the anonymous reviewers for their invaluable comments, which truly helped towards an effective presentation of this paper.

REFERENCES

[1] F. Zafari, A. Gkelias, and K. Leung. (Mar. 2018). "A survey of indoor localization systems and technologies." [Online]. Available: https://arxiv.org/abs/1709.01015v2

[2] E. Elnahrawy, J. Austen-Francisco, and R. P. Ma, "Adding angle of arrival modality to basic RSS location management techniques," in *Proc. 2nd Int. Symp. Wireless Pervasive Comput.*, San Juan, Puerto Rico, Feb. 2007, pp. 464–496.

[3] E.-Q. Liliana, J. Correa, E. Quezada, P. A. Quezada-Sarmiento, and E. Zelaya-Policarpo, "System of location and control of time of arrival of University buses using smartphone," in *Proc. 12th Iberian Conf. Inf. Syst. Technol.*, Lisbon, Portugal, Jun. 2017, pp. 1–6.

[4] F. Wen and C. Liang, "Fine-grained indoor localization using single access point with multiple antennas," *IEEE Sensors J.*, vol. 15, no. 3, pp. 1538–1544, Mar. 2015.

[5] J. Xiao, K. Wu, Y. Yi, and L. M. Ni, "FIFS: Fine-grained indoor fingerprinting system," in *Proc. 21st Int. Conf. Comput. Commun. Netw. (ICCCN)*, Munich, Germany, Aug. 2012, pp. 1–7.

[6] H. Santo, T. Maekawa, and Y. Matsushita, "Device-free and privacy preserving indoor positioning using infrared retro-reflection imaging," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, Kona, HI, USA, Mar. 2017, pp. 52–141.

[7] J.-Y. Kim, S.-H. Yang, Y.-H. Son, and S.-K. Han, "High-resolution indoor positioning using light emitting diode visible light and camera image sensor," *IET Optoelectron.*, vol. 10., no. 5, pp. 184–192, 2016.

[8] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "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.

[9] A. K. M. M. Hossain and W.-S. Soh, "A survey of calibration-free indoor positioning systems," *Comput. Commun.*, vol. 66, pp. 1–13, Jul. 2015.

[10] J. Xiao, Z. Zhou, Y. Yi, and L. M. Ni, "A survey on wireless indoor localization from the device perspective," *ACM Comput. Surv.*, vol. 49, no. 2, pp. 25_1–25_31, Oct. 2016.

[11] 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, 1st Quart., 2016.

[12] D. Lymberopoulos and J. Liu, "The microsoft indoor localization competition: Experiences and lessons learned," *IEEE Signal Process. Mag.*, vol. 34, no. 5, pp. 125–140, Sep. 2017.

[13] R. F. Brena, J. P. García-Vázquez, C. E. Galván-Tejada, D. Muñoz-Rodriguez, C. Vargas-Rosales, and J. Fangmeyer, "Evolution of indoor positioning technologies: A survey," *J. Sensors*, vol. 2017, Mar. 2017, Art. no. 2630413.

[14] M. A. Nuño-Maganda, H. Herrera-Rivas, C. Torres-Huitzil, H. M. Marín-Castro, and Y. Coronado-Pérez, "On-device learning of indoor location for WiFi fingerprint approach," *Sensors*, vol. 18, no. 7, p. 2202, Jul. 2018.

[15] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. Int. Conf. Comput. Commun.*, Tel Aviv, Israel, Mar. 2000, pp. 775–784.

[16] M. Shin and I. Joe, "An indoor localization system considering channel interference and the reliability of the RSSI measurement to enhance location accuracy," in *Proc. Int. Conf. Adv. Commun. Technol.*, Seoul, South Korea, Jul. 2015, pp. 582–583.

[17] C. Wu, Z. Yang, Z. Zhou, Y. Liu, and M. Liu, "Mitigating large errors in WiFi-based indoor localization for smartphones," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6246–6257, Jul. 2017.

[18] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity WiFi," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, Paris, France, Sep. 2015, pp. 53–64.

[19] Q. Song, S. Guo, X. Liu, and Y. Yang, "CSI amplitude fingerprinting-based NB-IoT indoor localization," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1494–1504, Jun. 2018.

[20] S. Shi, S. Sigg, L. Chen, and Y. Ji, "Accurate location tracking from CSI-based passive device-free probabilistic fingerprinting," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5217–5230, Jun. 2018.

[21] C. Xu, B. Firner, Y. Zhang, and R. E. Howard, "The case for efficient and robust RF-based device-free localization," *IEEE Trans. Mobile Comput.*, vol. 15, no. 9, pp. 2362–2375, Sep. 2016.

[22] P. Davidson and R. Piché, "A survey of selected indoor positioning methods for smartphones," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1347–1370, 2nd Quart., 2016.

[23] Z. Turgut, S. Üstebay, G. Z. G. Aydin, and A. Sertbas, "Deep learning in indoor localization using WiFi," in *Proc. Int. Telecommun. Conf.*, Istanbu, Turkey, Dec. 2018, pp. 101–110.

[24] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.

[25] X. Wang, L. Gao, and S. Mao, "CSI phase fingerprinting for indoor localization with a deep learning approach," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1113–1123, Dec. 2016.

[26] Z. E. Khatab, A. Hajihoseini, and S. A. Ghorashi, "A fingerprint method for indoor localization using autoencoder based deep extreme learning machine," *IEEE Sensors Lett.*, vol. 2, no. 1, Mar. 2018, Art. no. 6000204.

[27] H. Chen, Y. Zhang, W. Li, X. Tao, and P. Zhang, "ConFi: Convolutional neural networks based indoor Wi-Fi localization using channel state information," *IEEE Access*, vol. 5, pp. 18066–18074, 2017.

[28] X. Wang, X. Wang, and S. Mao, "CiFi: Deep convolutional neural networks for indoor localization with 5 GHz Wi-Fi," in *Proc. IEEE ICC*, Paris, France, May 2017, pp. 1–6.

[29] X. Wang, X. Wang, and S. Mao, "Deep convolutional neural networks for indoor localization with CSI images," *IEEE Trans. Netw. Sci. Eng.*, to be published.

[30] W. Shao, H. Luof, F. Zhao, C. Wang, A. Crivello, and M. Z. Tunio, "DePos: Accurate orientation-free indoor positioning with deep convolutional neural networks," in *Proc. Int. Conf. Exhib. Ubiquitous Positioning, Indoor Navigat. Location-Based Services*, Hubei, China, Mar. 2018, pp. 1–7.

[31] A. Mittal, S. Tiku, and S. Pasricha, "Adapting convolutional neural networks for indoor localization with smart mobile devices," in *Proc. GLSVLSI*, Chicago, IL, USA, May 2018, pp. 117–122.

[32] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann, "Weighted centroid localization in Zigbee-based sensor networks," in *Proc. IEEE Int. Symp. Intell. Signal Process.*, Alcala de Henares, Spain, Oct. 2007, pp. 1–6.

[33] D. Halperin. (2018). *Linux 802.11n CSI Tool*. https://dhalperi.github.io/linux-80211n-csitool/

[34] I. Goodfellow, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[35] S. Ruder. (2016). "An overview of gradient descent optimization algorithms." [Online]. Available: https://arxiv.org/abs/1609.04747

[36] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017.

[37] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[38] T. Williams and R. Li, "Advanced image classification using wavelets and convolutional neural networks," in *Proc. 15th Int. Conf. Mach. Learn. Appl.*, Anaheim, CA, USA, Dec. 2016, pp. 233–239.

[39] A. Rakhlin, A. Shvets, V. Iglovikov, and A. A. Kalinin. (2018). "Deep convolutional neural networks for breast cancer histology image analysis." [Online]. Available: https://arxiv.org/abs/1802.00752

[40] K. He, X. Zhang, S. Ren, and J. Sun. (2015). "Deep residual learning for image recognition." [Online]. Available: https://arxiv.org/abs/1512.03385
[41] C. Szegedy *et al.* (Sep. 2014). "Going deeper with convolutions." [Online]. Available: https://arxiv.org/abs/1409.4842v1
[42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn. (ICML)*, vol. 37, Jul. 2015, pp. 448–456.
[43] J. Brownlee. (Jan. 2019). *How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras*. [Online]. Available: https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/
[44] Y. Chapre, A. Ignjatovic, A. Seneviratne, and S. Jha, "CSI-MIMO: An efficient Wi-Fi fingerprinting using channel state information with MIMO," *Pervasive Mobile Comput.*, vol. 23, pp. 89–103, Oct. 2015.

CHAUR-HEH HSIEH received Ph.D. degree in electrical engineering from the Chung Cheng Institute of Technology, Taiwan, China, in 1990, where he joined the faculty of the Department of Electrical Engineering, in 1981, and became a Professor, in 1993. Since 1996, he has been with the Information Engineering Department, I-Shou University, as a Full Professor, where he was the Chairman, from 1999 to 2002. He was a Visiting Scholar with the Department of Electrical Engineering, University of Washington, in 2006. From 2007 to 2018, he was a Professor with Ming-Chuan University. He is currently a Professor with the Information Engineering College, Yango University, Fuzhou, China. His current research interests include signal and image processing, computer vision, and deep learning. He is an IET Fellow.

JEN-YANG CHEN was born in Hsinchu, Taiwan, China, in 1960. He received the M.S. degree in electrical engineering from Tatung University, Taipei, Taiwan, in 1992, and the Ph.D. degree in electrical engineering from Tamkang University, in 2000. He is currently a Full Professor with the Department of Electronic Engineering, Ming Chuan University. His research interests include deep learning, intelligent control, soft computing, and embedded micro-controller application and design.

BO-HONG NIEN was born in Taipei, Taiwan, China, in 1994. He received the Master's degree in electronic engineering from Ming Chuan University, in 2018. He is currently an Engineer of the System Service Technology Center, Industrial Technology Research Institute. His research interests include deep learning, embedded systems, computer science, and indoor positioning.