Deep Spatial–Temporal Model Based Cross-Scene Action Recognition Using Commodity WiFi

Biyun Sheng, Fu Xiao, Letian Sha, and Lijuan Sun

***Abstract*—With the popularization of Internet-of-Things (IoT) systems, passive action recognition on channel state information (CSI) has attracted much attention. Most conventional work under the machine-learning framework utilizes handcrafted features (e.g., statistic features) that are unable to sufficiently describe the sequence data and heavily rely on designers' experiences. Therefore, how to automatically learn abundant spatial–temporal information from CSI data is a topic worthy of study. In this article, we propose a deep learning framework that integrates spatial features learned from the convolutional neural network (CNN) into the temporal model multilayer bidirectional long short-term memory (Bi-LSTM). Specifically, CSI streams are segmented into a series of patches, from which spatial features are extracted by our designed CNN structure. Considering long-term dependencies between adjacent sequences, the fully connected layer of CNN for each patch is taken as the Bi-LSTM sequential input to further capture temporal features. Our model is appealing in that it can simultaneously learn temporal dynamics and convolutional perceptual representations. To the best of our knowledge, this is the first work to explore deep spatial–temporal features for CSI-based action recognition. Furthermore, in order to solve the problem that the trained model fully fails with environmental changes, we use the off-the-shelf model as the pretrained model and fine-tune it in the new scenario. The transfer method is able to realize cross-scene action recognition with low computational consumption and satisfactory accuracy. We carry out experiments on indoor data and the experimental results validate the effectiveness of our algorithm.**

***Index Terms*—Action recognition, bidirectional long short-term memory (Bi-LSTM), convolutional neural network (CNN), transfer learning.**

I. Introduction

HUMAN action recognition has recently gained much attention because of its wide applications in healthcare, smart homes, and public security [1]. Traditional recognition methods resorting to cameras [2], radars [3], or sensors [4] have achieved great success. However, there exist great limitations, such as illumination effectiveness and privacy invasion, limited resolution, and inconvenience, respectively, for camera-based, radar-based, and sensor-based recognition [1], [5]. To tackle the aforementioned limitations, ubiquitous signals between WiFi transmitters and receivers which can be influenced by human activities have recently been focused on and WiFi-based action recognition in indoor environments has become a hot research topic.

Prior works on WiFi-based recognition mostly use received signal strength (RSS) which coarsely describes WiFi channels and cannot exploit subcarriers in an orthogonal frequency-division multiplexing (OFDM) system for richer multipath information [6], [10]. Then, a fine-grained description of wireless signals named channel state information (CSI) is presented and widely applied for its robustness and adequacy. Raw CSI measures with high noise ratio cannot directly and effectively to represent action types. The majority of works extract handcrafted features from denoised CSI and combine classifiers to recognize actions under the machine-learning framework [6]–[8].

Despite the success of manual features, there still exist some weaknesses. First, manually designed features are dependent on prior knowledge and incapable of adequately mining spatial–temporal information in CSI streams. Second, separate stages for feature extraction and classifier learning may reduce the recognition results. Therefore, it is worth exploring the problem how to nonmanually obtain spatial–temporal features and jointly optimize feature learning as well as the classification process. In recent years, the deep learning model long short-term memory (LSTM) or variants bidirectional LSTM have been successfully applied for CSI-based action recognition [5], [9]. In these approaches, CSI amplitudes at each moment are fed into the LSTM model to capture temporal information of time series. However, the spatial information and local dependencies among CSI streams are ignored.

In this article, we attempt to design an end-to-end deep learning framework in which the convolutional neural network (CNN) features with spatial cues for CSI snippets are taken as the input of a two-layer bidirectional LSTM (Bi-LSTM) model. First, in order to capture the spatially local dependencies among CSI channels and adjacent sequences, we divide CSI streams of a sample into multiple segments, from which spatial features are extracted by a CNN model. Then, the original time sequences are converted into CNN feature arrays, each of which denotes spatial features for the corresponding CSI snippet. Furthermore, a Bi-LSTM model which includes

Manuscript received December 6, 2019; revised January 9, 2020; accepted February 5, 2020. Date of publication February 11, 2020; date of current version April 14, 2020. This work was supported in part by the Key Program of the National Natural Science Foundation of China under Grant 61932013, in part by the National Natural Science Foundation of China under Grant 61803212, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20180744, in part by the National Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 18KJB520034, and in part by the China Postdoctoral Science Foundation under Grant 2019M651920. *(Corresponding author: Fu Xiao.)*

The authors are with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China, and also with the Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210023, China (e-mail: biyunsheng@njupt.edu.cn; xiaof@njupt.edu.cn; ltsha@njupt.edu.cn; sunlj@njupt.edu.cn).

Digital Object Identifier 10.1109/JIOT.2020.2973272

forward and backward layers is utilized to process the CNN features extracted beforehand. Finally, the recognition results are directly gained by the classification layer of Bi-LSTM.

In real application scenes, the activity recognition model trained in a specific environment cannot work well if being applied to predict action types in another environment. Retraining models from scratch for each scenario will lead to highly expensive training cost. Inspired by the pretraining method in computer vision which fine-tunes the model trained on Imagenet for a new task [14], [15], we propose a similar transfer approach instead of retraining from the beginning. In contrast to retaining all layers from randomly initialized network parameters, fine-tuning certain layers with others fixed can not only reduce the computation consumption but also improve the recognition results.

In summary, the main contributions of this article are listed as follows.

1) We propose a novel deep learning framework that automatically learns features without experience and acquires recognition results without training additional classifiers.
2) We design a deep CNN structure that can capture the local dependencies and spatial characteristics. Besides, the scale invariance of CNN is beneficial for reducing the intraclass errors.
3) For the first time, the spatial and temporal information are simultaneously mined under the deep learning structure for CSI-based action recognition. The learned more discriminative features are beneficial for improving recognition accuracy.
4) A transfer learning method that fine-tunes the pretrained model of the original environment in a new environment is presented to reduce the computation burden in the training process.
5) Experiments are carried out on data sets collected in the indoor scenario to demonstrate the superiority of our proposed method.

The remainder of this article is organized as follows. Section II reviews recently related works on WiFi action recognition. Section III illustrates the background knowledge about CSI and Bi-LSTM. Section IV introduces our proposed deep learning framework. Section V presents the details of our transfer learning method. Then, in Section VI, we describe the experimental setup and the data collection process. A series of experiments and result analysis are also conducted. Finally, we conclude this article in Section VII.

II. Related Works

The WiFi-based recognition can be roughly divided into two categories, namely, RSSI based and CSI based. Sigg *et al.* [10] extracted statistical features, such as maximum, average, and variance from RSSI to recognize gestures. Abdelnasser *et al.* [6] proposed an RSSI-based gesture recognition system that analyzes different RSSI signal states, velocity, and amplitude to realize the recognition task. Sigg *et al.* [11] combined RSSI features and classifiers to recognize human actions. However, RSSI is a coarse description of signal channels without considering the number of antennas and subcarriers. Besides, unstable and noisy RSSI measures limit the performance of RSSI-based recognition.

In contrast to RSSI, CSI with multiple subcarriers between each pair of send–receive antennas is more informative and stable. Wang *et al.* [1] proposed a CSI-based human activity recognition and monitoring system, which consists of the CSI-speed model and the CSI-activity model. With the two models, the correlation between CSI and action type is established. Zeng *et al.* [12] presented a framework that can monitor the CSI variations to identify a person. They constructed CSI features of the time domain and frequency domain which are then compared to signatures for identity prediction. Li *et al.* [13] operated discrete wavelet transform (DWT) to compress the CSI waveform length by extracting the approximate sequence. Then, the dynamic time warping (DTW) algorithm is leveraged to compute the distance between two time series and identify the sample similarity. Wang *et al.* [8] changed CSI measurements into spectrograms from which features are extracted to characterize the walking pattern. Zhang *et al.* [18] tried multiple combinations of manual features and classifiers and designed an expert system to select a most suitable combination. Although manual features have been successfully applied in CSI-based recognition, it is difficult to design general features suitable for various action types.

With the development of computational capacity and data scale, deep learning has achieved great success for video-based action recognition [14], [15]. Due to its superior performance, researchers tend to solve the CSI-based action recognition problems by learning deep models. In [5], raw CSI amplitude data are fed into LSTM and the performance exceeds the traditional machine-learning framework with handcrafted features. Jiang *et al.* [17] presented a deep-learning activity recognition framework that can remove the environment-subject specific information and obtain environment-subject independent features. Zhang *et al.* [18] used deep supervised autoencoder to model the environment and then online get possible locations of the object. In general, research on CSI-based recognition by deep learning is still preliminary with huge room for improvement. Existing deep learning models cannot explore the spatial and temporal characteristics at the same time. The effective information loss may decrease the discriminative power of features and the final recognition rate.

Ideally, we hope to train a general model suitable for data collected from all environments. However, the trained model is usually environment-specific and completely loses efficacy when the environment changes. Jiang *et al.* [17] proposed a novel adversarial training network model to learn discriminative features with powerful classification capacity and weak domain discriminant ability. Zhang *et al.* [18] established a model which transfers source-domain features into those in the target domain. With transferred features, the learned classifier can successfully finish the recognition task in the target environment. Despite the cross-domain generalization ability, these recognition models rely on comprehensive transfer models or abundant expert knowledge. In the computer vision field, a deep convnet model is trained on a large-scale data set

Imagenet for image classification and models of new tasks, such as action recognition and object detection are further fine-tuned [14], [15] based on the pretrained model. Due to its simplicity and superiority, it is worthy of attempting to transferring the model trained in the source environment to the target scene in CSI activity recognition.

III. Preliminaries

In this section, we briefly introduce CSI and Bi-LSTM as background.

A. Channel State Information

Recently, WiFi devices widely use the OFDM multi-carrier modulation technology based on which the whole band is divided into multiple orthogonal subcarriers [20]. WiFi devices support multiple-input–multiple-output (MIMO), namely, multiple transmitting (TX) antennas and receiving (RX) antennas. The wireless signals received by each transmit–receive (TX–RX) pair can be denoted as

$$Y = H \times X + N \tag{1}$$

where Y is the received signal, X is the transmitted signal, and N is the noise signal. H represents the channel state matrix which can be expressed as follows:

$$H = \left[H(f_1), H(f_2), \ldots, H\left(f_{N_C}\right)\right] \tag{2}$$

where N_C is the number of OFDM subcarriers. In this article, $N_C = 30$ because we use Intel 5300 device which provides 30 subcarriers for each pair of TX–RX antenna. In MIMO scenario, the dimension of channel state matrix is extended to $N_T \times N_R \times N_C$, where N_T and N_R, respectively, denote the transmitting antennas and receiving antennas.

The CSI value of the kth subcarrier $H(f_k)$ can be expressed by

$$H(f_k) = |H(f_k)|e^{j\sin\theta} \tag{3}$$

where $|H(f_k)|$ and θ, respectively, represent the amplitude and phase. Due to carrier frequency offset (CFO) and sampling frequency offset (SFO) errors, phase information are rarely used for action recognition in previous works [5]; therefore, we use CSI amplitudes for further analysis in this article.

B. Bi-LSTM

LSTM with the capacity of sequential modeling has been successfully applied for CSI-based activity recognition [5]. It avoids the gradient vanishing and exploding problem by incorporating memory units that can preserve useful information with long-term dependencies [22].

As shown in Fig. 1, the LSTM structure includes an input gate i_t, a forget gate f_t, an output gate o_t, an input modulation gate g_t, a memory cell c_t, and a hidden unit h_t. The LSTM parameters at timestep t can be updated as follows:

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ f_t &= \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \end{aligned}$$

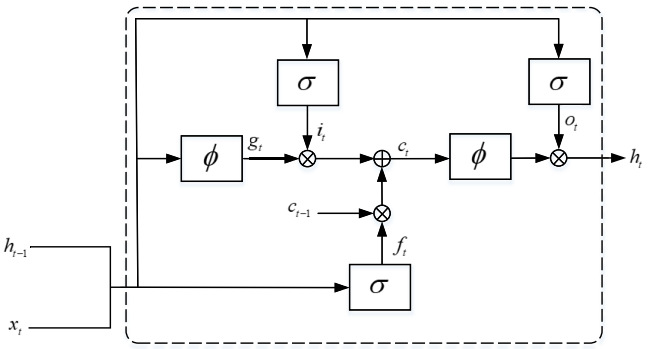


Fig. 1. LSTM diagram.

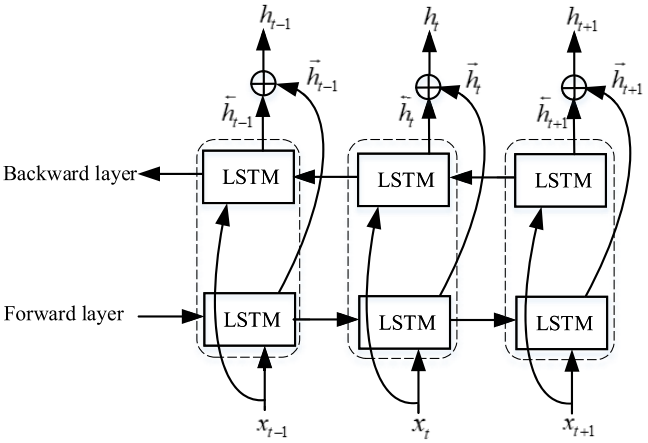


Fig. 2. Bi-LSTM diagram.

$$\begin{aligned} g_t &= \phi\left(W_{xg}x_t + W_{hg}h_{t-1} + b_g\right) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= \phi(c_t) \odot o_t \end{aligned} \tag{4}$$

where the function, σ and ϕ, respectively, denote sigmoid and hyperbolic tangent activation functions.

The input gate i_t and forget gate f_t selectively forget previous cell or process the current input. Similarly, the output gate o_t determines the memory cell transferred to hidden gates. The LSTM network deals with the sequential data in one direction, only considering the past information. However, the Bi-LSTM extends the traditional LSTM into two directions by exploring information from both the previous and the future. The Bi-LSTM structure is illustrated in Fig. 2.

In Fig. 2, the Bi-LSTM model includes the forward layer and the backward layer which, respectively, transmits information along temporal order and reverse order. The hidden unit of Bi-LSTM at the tth moment can be expressed as follows:

$$h_t = \overrightarrow{h}_t \oplus \overleftarrow{h}_t \tag{5}$$

where $\oplus$ is the concatenation operation and $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ are hidden units in the forward layer and the backward layer.

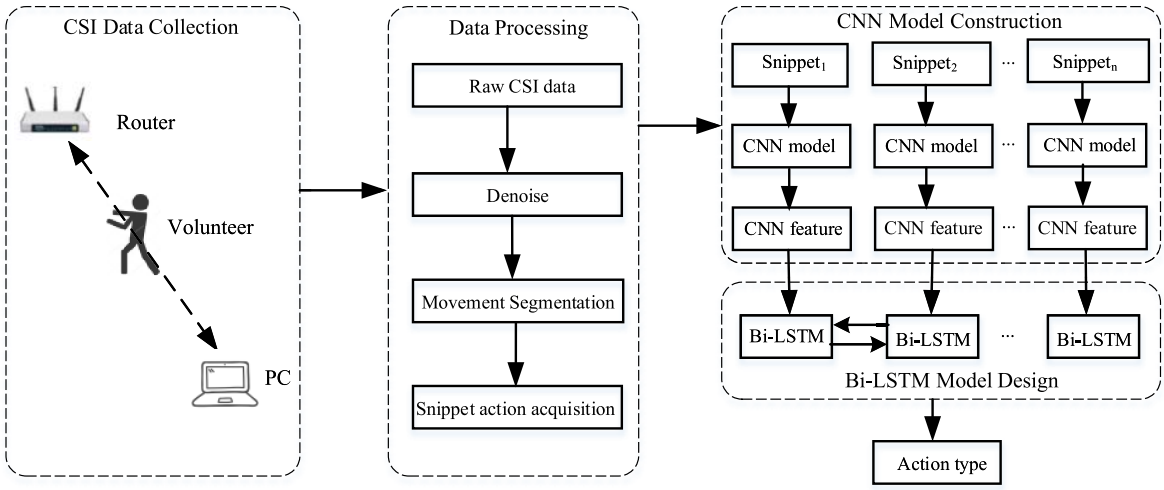


Fig. 3. Framework of our approach.

IV. System Design

In order to simultaneously mine spatial and temporal information and improve the discriminative power of CSI-based action features, we develop a deep learning system that integrates the CNN features into the multilayer Bi-LSTM model. As illustrated in Fig. 3, the framework of our proposed method consists of data processing, CNN model construction, and multilayer Bi-LSTM network design. In this section, we will discuss the three parts in detail.

A. Data Processing

1) Denoise: The WiFi signals collected by our equipment contain many interferences which are mainly caused by high-frequency noises. The raw CSI data cannot be directly used for the subsequent processing and we use Butterworth low-pass filter to obtain the denoised data. Taking a sample, for example, the CSI amplitude waveforms of the first subcarrier before and after denoising are shown in Fig. 4. It can be easily observed that the denoised signals are smoother with low noise level.

2) Movement Segmentation: For CSI data collection of a time series, volunteers are asked to operate a certain action ten times with time intervals. Therefore, it is necessary to detect the start and end points of activity from the whole time series. In Fig. 4, variations of the nonaction part are relatively small while human activity results in a typical increasing and decreasing trend in CSI waveforms.

In this article, we present a segmentation algorithm based on differences of time series. Specifically, we use a series of overlapped sliding windows and calculate the mean absolute differences for all CSI streams in each window

$$\bar{X} = \frac{\sum_{i=1}^{d} \sum_{j=s}^{s+w-1} \left| X_{i,j+1} - X_{i,j} \right|}{d * w} \tag{6}$$

where d, s, and w, respectively, denote the CSI dimension, the start point, and the size of sliding window.

Then, we compare the calculated $\bar{X}$ with the preset start threshold T_1 and end threshold T_2 ($T_1 > T_2$) to detect the action part

$$\begin{cases} P_s = j, & \bar{X} \geq T_1 \\ P_e = j, & \bar{X} \leq T_2 \text{ and } P_s \neq \emptyset \end{cases} \tag{7}$$

where P_s and P_e are, respectively, the start point and end point of a certain activity.

Finally, we remove false-positive points from the obtained point sets by postprocessing. For example, the timespan for some action should be within a range. We list subcarrier 1 of some segmented action samples in Fig. 5. Generally, the same actions appear similar while different actions show differences in the waveform.

3) Snippet Action Acquisition: Considering the fact that CSI amplitudes in a short time have a local dependency and spatial correlation, we attempt to segment the action samples into multiple short-time fragments. Then, it is convenient for us

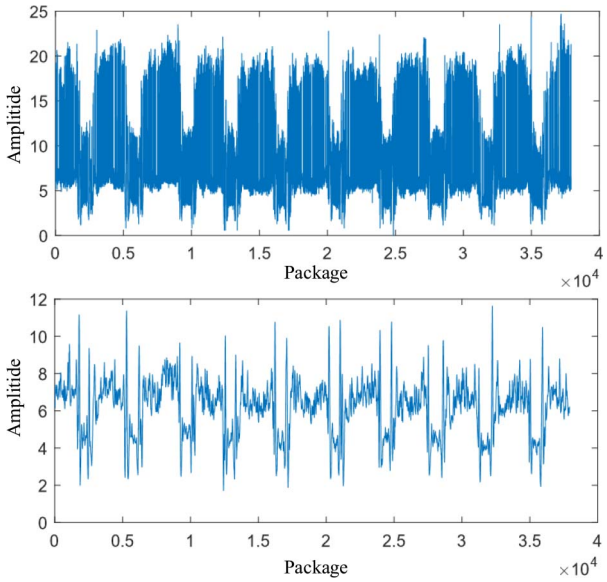


Fig. 4. Illustration of raw CSI (the first row) and denoised CSI amplitude (the second row).

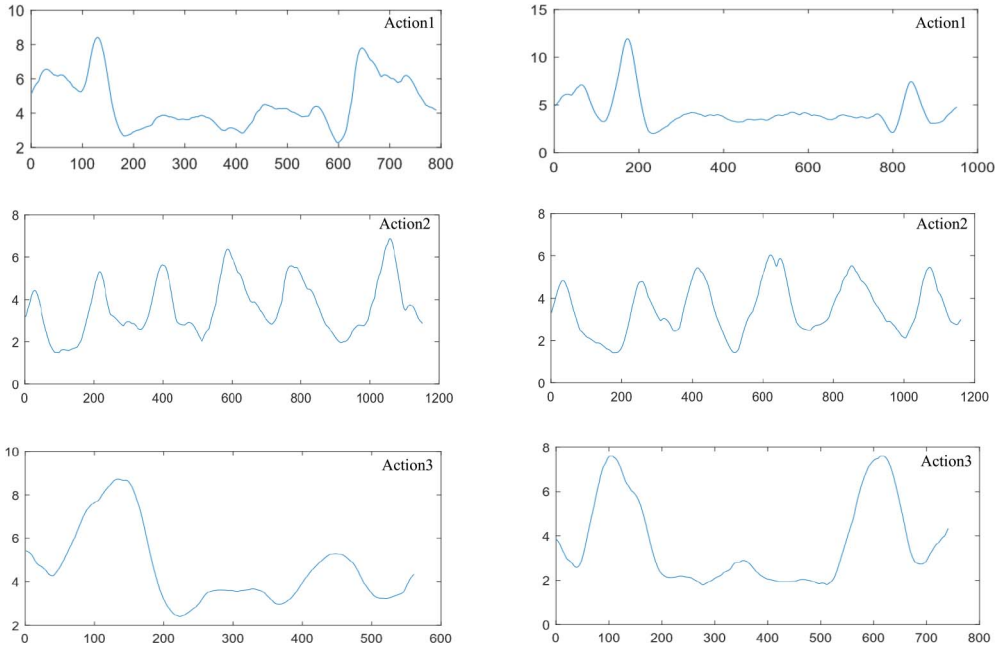


Fig. 5. Subcarrier 1 of segmented action examples.

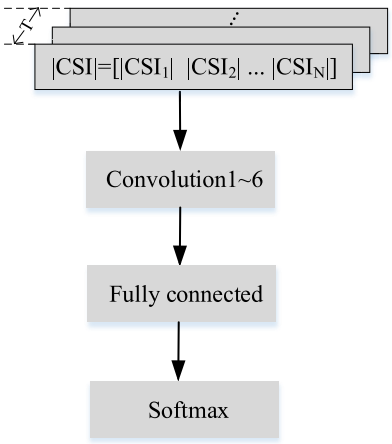


Fig. 6. Overall architecture of our CNN.

TABLE I
PARAMETERS IN OUR CNN STRUCTURE

CNN Units		
Convolutional Layers	Activation Function	Pooling Layers
Conv1 (16 1×3)	ReLULayer	Max pooling (1×2)
Conv2 (24 1×3)	ReLULayer	Max pooling (1×2)
Conv3 (32 1×3)	ReLULayer	Max pooling (1×2)
Conv4 (48 1×3)	ReLULayer	Max pooling (1×2)
Conv5 (64 1×3)	ReLULayer	Max pooling (1×2)
Conv6 (96 1×3)	ReLULayer	Max pooling (1×2)

to further explore the spatial information within each snippet and the temporal characteristic between adjacent snippets. Compared with those taking the whole sequence as LSTM input [5], our proposed approach can not only mine spatial features by CNN but also reduce the length of time series in Bi-LSTM.

B. CNN Model Construction

CNN which has advantages of modeling spatial context information in the 2-D space has achieved great success in the field of image classification [15]. However, there are limited works concentrating on how to construct a CNN model suitable for wireless sensing problems. In this article, we design a CNN structure for CSI-based action recognition in Fig. 6.

As shown in Fig. 6, the CNN structure consists of six convolutional layers, one fully connected layer, and a softmax layer. Convolutional layer filters are connected to local regions in the forward propagation process and suitable to explore local correlation in input space. Then, the top fully connected layer combines together various spatial information in lower layers.

Specifically, the convolutional layers in our CNN architecture are described in Table I. The first convolutional layer densely filters the input with 16 kernels of size 1×3 followed by the activation, normalization, and pooling operations. Similarly, the second convolutional layer filters the output of the first convolutional layer with 24 kernels of size 1×3. The remaining convolutional layers are sequentially connected with 32 kernels of size 1×3, 48 kernels of size 1×3, 64 kernels of size 1×3, and 96 kernels of size 1×3, respectively. The output size of neurons in the fully connected layer is equivalent to the action type number. Local area perceptions from multiple convolutional layers are finally combined by the high-level fully connected layer to obtain global information and spatial context are captured with the deep architecture of convolutional design.

Mathematically, suppose inputs for CNN structure are denoted as $S = \{S_1, S_2, \ldots, S_i, \ldots, S_m\}$, where m is the number of snippets, $S_i \in \mathbb{R}^{N \times T}$ is a 2-D matrix in which N is the subcarrier number between transmitter and receiver pairs and T is the time duration of a snippet. The inputs S are

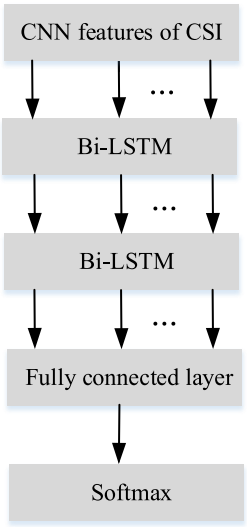


Fig. 7. Overall architecture of our Bi-LSTM.

processed by multiple convolutional layers and a fully connected layer to generate fixed-size feature vectors $\Phi(S) = \{\Phi(S_1), \Phi(S_2), \ldots, \Phi(S_i), \ldots, \Phi(S_m)\}$, in which $\Phi(S_i) \in \mathbb{R}^k$ and k is the action type number.

Then, we apply the softmax function as the network classifier to represent the probability distribution over k different possible categories

$$y_{\text{pre}} = \text{Softmax}\left(W^T \Phi(S) + b\right) \tag{8}$$

where y_{pre}, W, and b are separately denoted as the predicted action labels, the weights coefficients, and the bias.

Finally, the classification layer is used to compute the loss between the predictions y_{pre} and the actual values y_{act}. In this article, we adopt cross-entropy loss L for multiclass activity classification problems

$$L = -\sum_{i=1}^{M} y_{\text{act}}^i \log\left(y_{\text{pre}}^i\right) \tag{9}$$

where M is the snippet number of training samples.

C. Bi-LSTM Model Design

With the trained CNN model, we extract the top fully connected layer features to represent successive snippets and attempt to model the time sequence structure. LSTM is a powerful tool for encoding the temporal dependencies [5] and conventional LSTM only considers the influences of past information on the current hidden nodes. However, future information is also important for synthesizing temporal dynamics. For example, "boxing" and "clap" both starting with "lifting the wrist" represent two different actions. Bi-LSTM with forward and backward layers is capable of simultaneously modeling the current nodes with past and future information. Therefore, we use Bi-LSTM to incorporate CSI bidirectional information and learn discriminative context features. The proposed Bi-LSTM schema is illustrated in Fig. 7.

Our Bi-LSTM model works by passing the fixed-length vector representation $\Phi(S_i)$ obtained from the CNN feature extractor Φ. With features of input sequence computed in spatial space, the sequence model is then constructed for $\{\Phi(S_1), \Phi(S_2), \ldots, \Phi(S_m)\}$. Our sequential structure is composed of two-layer Bi-LSTMs: 1) a fully connected layer and 2) a softmax layer. The Bi-LSTM layers from the top–down are separately defined with the "sequence" and "last" output mode. Namely, the first Bi-LSTM outputs at each time are taken as the input of the next Bi-LSTM layer which transmits features at the last time into the fully connected layer. With two-layer Bi-LSTMs designed, more fine-grained temporal dynamics are established. The final softmax and classification layers are similar to that in the CNN structure. By building the sequential model for CNN features, the temporal information is further explored.

D. Network Parameters Training

With randomly initialized parameters, action labels are produced by propagating the input with multilayer networks in the forward propagation process. Then, errors evaluated by the loss function between actual labels and predicted labels are propagated backward through the network to compute the gradient in the backward propagation process. Network parameters are updated during each iteration until convergence. Referring to other deep CNN works [23], [24], we use the stochastic gradient descent (SGD) [25] and adaptive moment estimation (ADAM) [26] optimization algorithm, respectively, for the CNN and Bi-LSTM training processes.

V. Transfer Learning

The obtained CSI signals usually carry much information that is specific to the collection environment and the human subjects performing actions. Therefore, the trained model on specific subjects in a specific environment cannot be directly applied to recognize other subject actions in different environments. Retraining a new model for a new scenario requires a good deal of computational resources.

The problem can be solved if less new data and computation are required when the scene changes. To address the challenge and improve the robustness to dynamics, we consider a transfer learning algorithm which fine-tunes the new model based on the parameters learned on another similar task instead of training from scratch. In detail, we train our proposed model on the activity recognition task T_1

$$\theta(T_1) = \arg\min_{\theta \in \Theta} \text{Loss}(\text{Input}(T_1), \theta) \tag{10}$$

where Θ, $\text{Input}(T_1)$, and Loss, respectively, denote the parameter domain, input data of T_1, and the model loss function.

Then, we try to finish another similar task T_2 in which the data collection scene and subjects are different from that in T_1. In this article, we adopt the transfer learning approach that fine-tunes the parameters $\theta(T_1)$ trained on T_1 for the T_2 task

$$\theta(T_2) = \arg\min_{\theta \in \Theta} \text{Loss}(\text{Input}(T_2), \theta(T_1) + \theta) \tag{11}$$

where $\text{Input}(T_2)$ is the subset of the data collected in T_2. In our approach, the parameters of convolutional layers in CNN and Bi-LSTM layers are fixed with only the fully connected layer

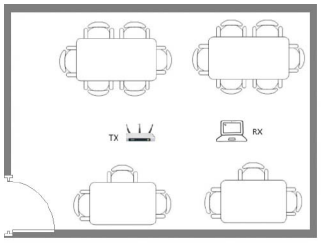


Fig. 8. Site layout Room1.

Fig. 9. Schematic of our defined action types, which, respectively, presents bend, box, wave, and clap from left to right.

parameters fine-tuned by a part of data in T_2. Consequently, the data demand and computation consumption can be reduced when the scene is transferred.

VI. Experiments

In order to validate the performance of our proposed approach, we conduct a series of experiments. In this section, we first introduce the experimental setup and data description. Then, we illustrate the performances on our collected data sets and explore the parameters influence. Furthermore, we discuss the transfer learning results when the environment changes. Finally, we show the experimental results on a larger data set with more action types.

A. Experimental Setup and Data Description

Our proposed action recognition system is based on commercial off-the-shelf (COTS) wireless router continuously sending WiFi signals and laptop computer equipped with Intel 5300 network card. The router with three antennas acts as a transmitter, the transmission rate of which is set as 100 packets/s in our experiment. The receiver contains three antennas responsible to receive WiFi signals. Each TX-RX antenna pair has 30 subcarriers. Therefore, we can collect 270 CSI subcarriers based on Linux 802.11n CSI tool [19] at each time and select 90 CSI signals from three TX-RX pairs for the following analysis.

The data are collected in a typical indoor scene (a meeting room named Room1) with the 9 × 6 m size, the layout of which is shown in Fig. 8. In the experiment, we employ five volunteers to perform four different actions, including "bend," "box," "wave," and "clap" shown in Fig. 9. We let the volunteers repeat these four activities in for ten rounds without any restrictions. In total, we collect 200 sets of activity data.

In order to validate the capacity of transfer learning, we further collect action data in another two site layouts, namely, a meeting room (Room2) and lab (Room3) shown in Fig. 10

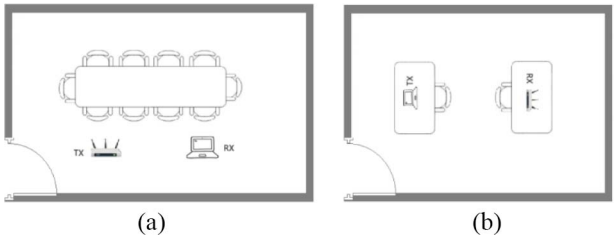


Fig. 10. Another two site layouts. (a) Room2. (b) Room3.

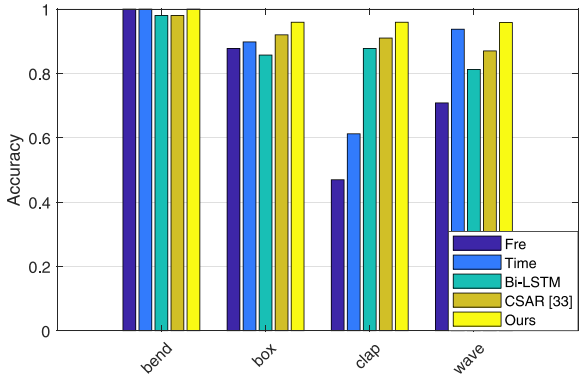


Fig. 11. Comparison results on baselines in which the time-domain and frequency-domain features-based framework are, respectively, named "Time" and "Fre."

where volunteers are also required to do actions, including bend, box, wave, and clap.

Furthermore, considering evaluations of our approach on more different action types, we use the data set collected in an indoor office area where the actions are defined as "Lay down, Fall, Walk, Run, Sit down, Stand up" [5]. Six volunteers are asked to do these activities 20 rounds in line-of-sight (LOS) condition and videos are recorded to label data at the same time. We segment the action time series according to the given video labels as [5].

In the following experiments, the data sets are divided into ten splits randomly. We separately select one split as the testing set with others as the training set. Then, the final accuracy is evaluated by averaging the ten results. The following listed experimental results all adopt the tenfold cross-validation evaluation method if not specified.

B. Experimental Results on Data Collected From Room1

Comparisons With Baselines: We first compare the performances with those by some baselines, such as traditional handcrafted features or basic LSTM-based machine-learning algorithm. In detail, we try to extract time-domain features (e.g., average, minimum, maximum, etc.) and frequency-domain features (e.g., energy, domain-frequency ratio, and FFT peaks) [6], [12], [18] followed by a support vector machine (SVM) classifier [28]. Besides, we construct a simple one-layer Bi-LSTM model with 100 hidden units. Fig. 11 shows the comparison results.

We can observe that our presented framework achieves better performance in contrast to all the baselines. Traditional hand-crafted features fail to sufficiently capture spatial or

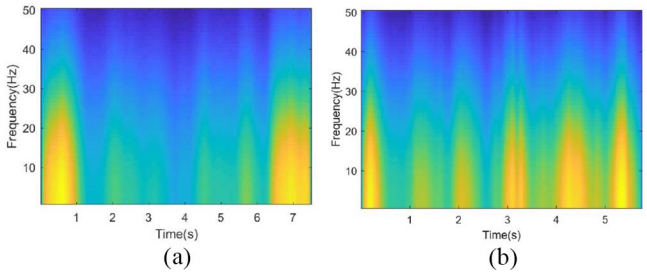


Fig. 12. CSI spectrograms of two typical activities. (a) CSI spectrogram of bend activity. (b) CSI spectrogram of box activity.

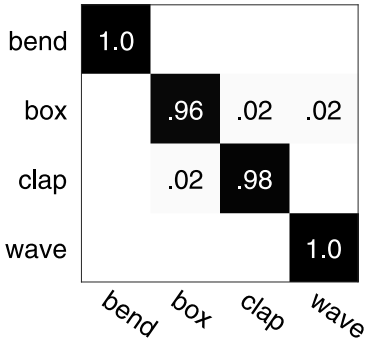


Fig. 13. Confusion matrix of the proposed method on our collected data from Room1.

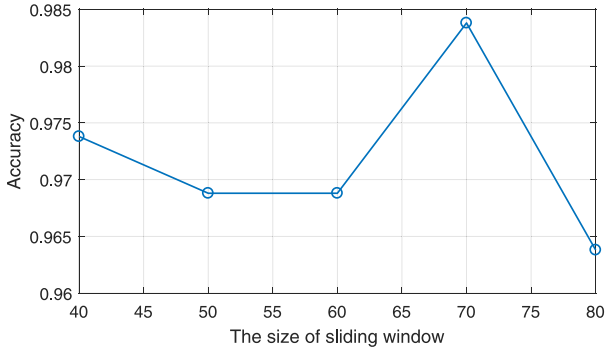


Fig. 14. Impact of the sliding window size on performance.

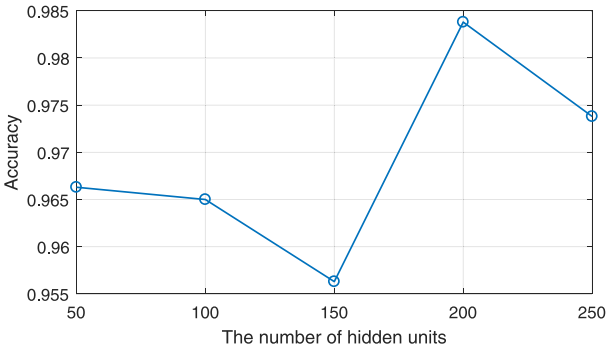


Fig. 15. Impact of the hidden units number on performance.

temporal information which leads to inferior performances. Actually, there exist action feature information in denoised CSI data. Taking the bend and box activities, for example, we list related CSI spectrograms in Fig. 12. High energy appears at the start and end stages in Fig. 12(a) due to "bend forward–pause–get up" processes involved in the bend action. In contrast, the CSI spectrogram in Fig. 12(b) changes periodically within a frequency range which matches with cyclic arm movements in the box action. Therefore, the deep learning Bi-LSTM model which has the superior capacity of automatically learning discriminative features from CSI data can beat the listed hand-crafted features in accuracy. Based on Bi-LSTM, the CSAR framework [30] which selects good quality channels can improve the recognition accuracy. Furthermore, our proposed method which tries to mine deep spatial–temporal features performs best.

Specifically, we illustrate the related confusion matrix by the proposed approach in Fig. 13. All action performances exceed 95% and the overall recognition accuracy can achieve a satisfactory result 98.38%. Activities box and clap are confused sometimes because they are all concerned about arm movements with several similarities.

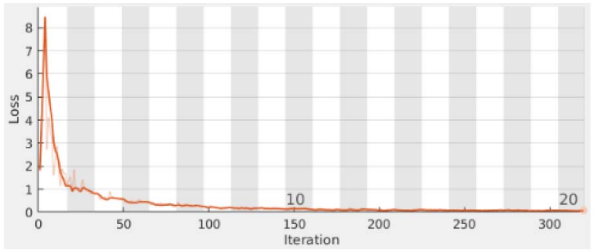
Impact of the Sliding Window Size: In this article, we set l and $l/2$, respectively, as the window size and step size to sample CNN inputs. In this part, we investigate the impact of the window size on the final accuracy in Fig. 14. The overall accuracy fluctuates with the sliding window size within 2%. The window size for the CNN structure determines the trade-offs between space and time resolution of our method. With a larger window size, our framework has higher spatial resolution but lower time resolutions. Therefore, a reasonable size is more beneficial to mine spatial–temporal information and further improve the accuracy. Considering the action duration and performance, we set $l = 70$ for our collected data.

Impact of the Hidden Units Number: We try to explore the impact of the hidden units number on the recognition accuracy in Fig. 15. When the number of hidden units is set between 50 and 150, the recognition accuracy is around 96.5%. With more hidden nodes used, the recognition can be further improved to about 98%. Taking the computational consumption and the performance into consideration, we choose 200 as the number of hidden nodes.

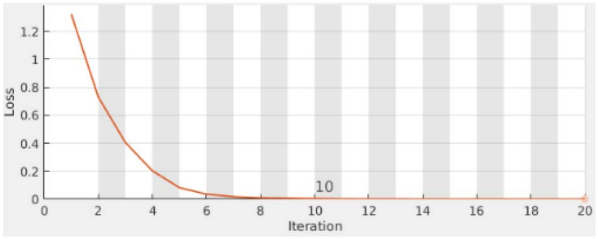
Impact of Sample Diversity and Scale: In order to test the performance of our algorithm on diverse and larger scale samples, we combine data from Room1, Room2, and Room3 together with tenfold cross-validation adopted. The achieved overall accuracy 96.96% illustrates that our model is robust to the sample variety and size.

Computational Cost: Experiments are conducted on the computer with a CPU 4110 2.1 GHz and a GPU of NVIDIA GeForce GTX1080Ti. We plot the training curves, including CNN and Bi-LSTM processes in Fig. 16. We can observe that the CNN training loss tends to convergence after iterating 150 times. Our Bi-LSTM training process converges faster with features learned from our deep CNN structure as input because CNN representations have discriminative power of classification.

Despite the relatively large time consumption during the training process, we only need to learn deep models offline and then identify the testing samples online. The computation time of our approach is about 0.003 s for testing a sample from our collected data. Therefore, the proposed model is efficient and suitable for real-time CSI-based action recognition.



(a)



(b)

Fig. 16. Convergence curves for deep models. (a) Loss for CNN training. (b) Loss for Bi-LSTM training.

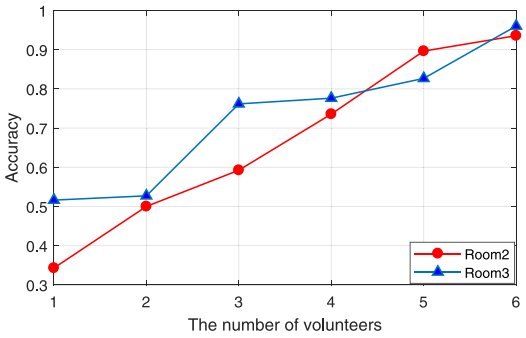


Fig. 17. Transfer learning performances.

C. Transfer Learning Results

In order to validate transfer learning results, the data collected from Room1 are taken as the training domain with Room2 and Room3, respectively, as the test scenarios. If the trained deep model on Room1 is directly utilized to recognize action samples in Room2 or Room3, the recognition rate, respectively, reaches 19.4% and 15.6% which are far from satisfaction. The drop results further verify the fact that the original model fails when the environment changes. In this article, we utilize the learned model on Room1 as the pretrained model and select parts of data collected in Room2 or Room3 to fine-tune the pretrained model. The recognition accuracy with the number of volunteers applied for fine-tuning is plotted in Fig. 17. We use 20 sets of activity data for each volunteer, and note that the performance can be improved to over 90% with six volunteer data involved fine-tuning which indicates the capacity of transfer learning.

D. Experimental Results on Data Set With More Action Types

In this section, we conduct experiments to validate the performance of our method on a larger data set with more activity types. In Fig. 18, the accuracies of most actions achieve over 90% except the "standup" action which is easy

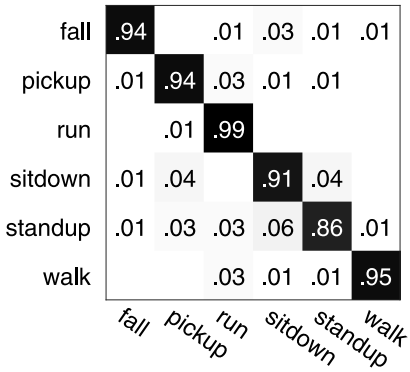


Fig. 18. Confusion matrix of the proposed method on a larger data set.

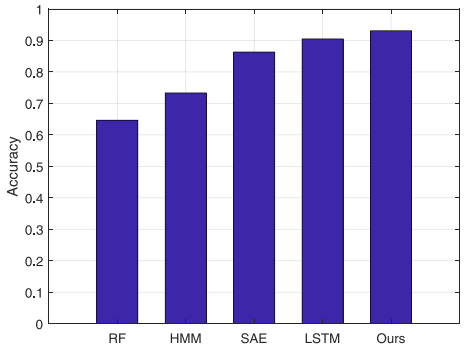


Fig. 19. Comparison results on baseline methods listed in [5].

to be misclassified into "sitdown." The main reason is that there exist high similarities in the movement processes of both activities. Besides, we can observe from Fig. 19 that the deep learning framework is obviously superior to traditional shallow algorithms, such as random forest (RF) and hidden Markov model (HMM) [5]. The sparse autoencoder (SAE) [29] deep framework performs better than hand-crafted features; however, it is still inferior to LSTM which considers the temporal dependencies in sequential data. Our presented framework with spatial–temporal information can further improve the overall performance. It can also be inferred from the result that our approach is suitable to be applied in various action scenarios.

VII. Conclusion

In this article, we proposed a framework that integrates the CNN features into the multilayer Bi-LSTM structure. This is the first work that attempts to mine deep spatial–temporal information for CSI-based action recognition. Completely different from previous hand-crafted features, a CNN architecture is established to automatically learn features with spatial cues from the CSI snippet streams, which are then fed into Bi-LSTM for exploring temporal context information. With time-scale and spatial-scale information simultaneously incorporated, the discriminative power of features and action recognition performance can be improved. In order to reduce the training burden and ensure recognition accuracy when the environment changes, we fine-tune the pretrained model

instead of learning from scratch. Extensive experiments on different data sets demonstrate the effectiveness of the proposed framework.

REFERENCES

[1] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of WiFi signal based human activity recognition," in *Proc. ACM Conf. Mobile Comput. Netw. (MobiCom)*, 2015, pp. 65–76.

[2] J. K. Aggarwal and S. R. Michael, "Human activity analysis: A review," *ACM Comput. Surveys*, vol. 43, no. 3, pp. 1–47, 2011.

[3] *Google Project Soli*, 2015. [Online]. Available: https://www.youtube.com/watch?v=0QNiZfSsPc0

[4] Y. Song, L. Liu, H. Ma, and A. V. Vasilakos, "A biology-based algorithm to minimal exposure problem of wireless sensor networks," *IEEE Trans. Netw. Service Manag.*, vol. 11, no. 3, pp. 417–430, Sep. 2014.

[5] S. Yousefi, H. Narui, S. Dayal, S. Ermon, and S. Valaee, "A survey on behavior recognition using WiFi channel state information," *IEEE Commun. Mag.*, vol. 55, no. 10, pp. 98–104, Oct. 2017.

[6] H. Abdelnasser, M. Youssef, and K. A. Harras, "WiGest: A ubiquitous WiFi-based gesture recognition system," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2015, pp. 1472–1480.

[7] L. Liu, Y. Song, H. Zhang, H. Ma, and A. V. Vasilakos, "A biology-inspired algorithm for the Steiner tree problem in networks," *IEEE Trans. Comput.*, vol. 64, no. 3, pp. 819–832, Mar. 2015.

[8] W. Wang, A. X. Liu, and M. Shahzad, "Gait recognition using WiFi signals," in *Proc. Int. Joint Conf. Pervasive Ubiquitous Comput. (UbiComp)*, 2016, pp. 363–373.

[9] Z. Chen *et al.*, "WiFi CSI based passive human activity recognition using attention based Bi-LSTM," *IEEE Trans. Mobile Comput.*, early access.

[10] S. Sigg, M. Scholz, S. Shi, Y. Ji, and M. Beigl, "RF-sensing of activities from non-cooperative subjects in device-free recognition systems using ambient and local signals," *IEEE Trans. Mobile Comput.*, vol. 13, no. 4, pp. 907–920, Apr. 2014.

[11] S. Sigg, S. Shi, F. Büsching, Y. Ji, and L. C. Wolf, "Leveraging RF-channel fluctuation for activity recognition: Active and passive systems, continuous and RSSI-based signal features," in *Proc. Int. Conf. Adv. Mobile Comput. Multimedia (MoMM)*, 2013, pp. 43–52.

[12] Y. Zeng, P. H. Pathak, and P. Mohapatra, "WiWho: WiFi-based person identification in smart spaces," in *Proc. IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, 2016, pp. 1–12.

[13] M. Li *et al.*, "When CSI meets public WiFi: Inferring your mobile phone password via WiFi signals," in *Proc. ACM Conf. Comput. Commun. Security (CCS)*, 2016, pp. 1068–1079.

[14] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 568–576.

[15] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F.-F. Li, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 1725–1732.

[16] 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.

[17] W. Jiang *et al.*, "Towards environment independent device free human activity recognition," in *Proc. ACM Conf. Mobile Comput. Netw. (Mobicom)*, 2018, pp. 289–304.

[18] J. Zhang, Z. Tang, M. Li, D. Fang, P. Nurmi, and Z. Wang, "CrossSense: Towards cross-site and large-scale WiFi sensing," in *Proc. ACM Conf. Mobile Comput. Netw. (Mobicom)*, 2018, pp. 305–320.

[19] P. Yazdanian and V. Pourahmadi, "DeepPos: Deep supervised autoencoder network for CSI based indoor localization," *CoRR*, vol. abs/1811.12182, pp. 1–10 Nov. 2018.

[20] C. Wang, S. Chen, Y. Yang, F. Hu, F. Liu, and J. Wu, "Literature review on wireless sensing Wi-Fi signal based recognition of human activities," *Tsinghua Sci. Technol.*, vol. 23, no. 2, pp. 203–222, Apr. 2018.

[21] S. Zhang *et al.*, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, 2016, pp. 207–212.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] J. Zang, L. Wang, Z.-Y. Liu, Q. Zhang, G. Hua, and N. Zheng, "Attention-based temporal weighted convolutional neural network for action recognition," in *Proc. Int. Conf. Artif. Intell. Appl. Innov.*, 2018, pp. 97–108.

[24] X. Yao, "Attention-based BiLSTM neural networks for sentiment classification of short texts," in *Proc. Inf. Sci. Cloud Comput. (ISCC)*, 2017, pp. 1–8.

[25] M. Zinkevich, M. Weimer, A. J. Smola, and L. Li, "Parallelized stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2010, pp. 2595–2603.

[26] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–13.

[27] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, p. 53, 2011.

[28] C. C. Chang and C. J. Lin, "LibSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.

[29] J. Wang, X. Zhang, Q. Gao, H. Yue, and H. Wang, "Device-free wireless localization and activity recognition: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6258–6267, Jul. 2017.

[30] F. Wang, W. Gong, J. Liu, and K. Wu, "Channel selective activity recognition with WiFi: A deep learning approach exploring wideband information," *IEEE Trans. Netw. Sci. Eng.*, early access, doi: 10.1109/TNSE.2018.2825144.

Biyun Sheng received the B.S. and M.S. degrees from the School of Electrical and Information Engineering, Jiangsu University, Zhenjiang, China, in 2010 and 2013, respectively, and the Ph.D. degree from the School of Automation, Southeast University, Nanjing, China, in 2017.

She is currently with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing. Her research interests include pattern recognition, computer vision, machine learning, and wireless sensing.

Fu Xiao received the Ph.D. degree in computer science and technology from Nanjing University of Science and Technology, Nanjing, China, in 2007.

He is currently a Professor and a Ph.D. Supervisor with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing. His research papers have been published in many prestigious conferences and journals, such as IEEE INFOCOM, IEEE ICC, IEEE IPCCC, the IEEE/ACM TRANSACTIONS ON NETWORKING, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE TRANSACTIONS ON MOBILE COMPUTING, the *ACM Transactions on Embedded Computing Systems*, and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. His research interests are mainly in the areas of Internet of Things and mobile computing.

Prof. Xiao is a member of the IEEE Computer Society and the Association for Computing Machinery.

Letian Sha received the B.S. and M.S. degrees from the School of Computer, Southwest Jiaotong University, Chengdu, China, in 2007 and 2010, respectively, and the Ph.D. degree from the School of Computer, Wuhan University, Wuhan, China, in 2014.

He is currently with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include network security WSN security, IoT security, and information security.

Lijuan Sun received the Ph.D. degree in information and communication from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2007.

She is currently a Professor and a Ph.D. Supervisor with the School of Computer Science, Nanjing University of Posts and Telecommunications. Her main research interests are wireless sensor networks and wireless mesh networks.