IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 23, 2019, accepted September 5, 2019, date of publication September 11, 2019,
date of current version September 24, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2940386

WiGrus: A Wifi-Based Gesture Recognition System Using Software-Defined Radio

TAO ZHANG, TINGYU SONG, DAOLIN CHEN, TIAN ZHANG, AND JIE ZHUANG, (Member, IEEE)
School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
Corresponding author: Jie Zhuang (jz@uestc.edu.cn)

This work was supported in part by the Sichuan Science and Technology Program under Grant 2017JY0223, and in part by the National Natural Science Foundation of China (NSFC) under Grant 61571090.

ABSTRACT With the proliferation of WiFi devices and infrastructures, the ubiquitous WiFi signals are used to transmit user data. Besides it is also capable of sensing and identifying human gestures. In this paper, we propose a WiFi-based gesture recognition system, namely WiGrus, which solves the problems of user privacy and energy consumption compared with the approaches using wearable sensors and depth cameras. WiGrus leverages the fine-grained Channel State Information (CSI) extracted from WiFi signals to recognize a set of hand gestures. First of all, we utilize timestamps attached to the extracted CSI values to split continuously received WiFi packets into gesture instances. Second, a Principal Component Analysis (PCA)-based method and the first order difference are employed to reduce the noise and mitigate multipath effects caused by the environment changes. Then, massive features are extracted from the processed CSI values to present the intrinsic characteristics of each gesture. Finally, a 2-stage-RF algorithm is proposed to classify the gestures. Our experiments are implemented with a wireless router and a Software Defined Radio (SDR) device, more specifically Universal Software Radio Peripheral (USRP), which are used as WiFi signal transmitter and receiver respectively. The experimental results demonstrate that WiGrus can achieve an average accuracy of 96% in Line-of-sight (LOS) scenario and 92% in Non-Line-of-Sight (NLOS) scenario in the office environment and is robust to the environment changes.

INDEX TERMS Channel state information (CSI), gesture recognition, WiFi, random forest (RF), timestamp, software defined radio (SDR), universal software radio peripheral (USRP).

I. INTRODUCTION

Human motion detection utilizes specific devices and approaches to extract the characteristics of a person's movement states. In previous works, many researchers focus on sensors to sense and detect human motions. Those methods require people to be equipped with dedicated sensors, such as motion sensors, accelerometer sensors, and gyroscopes, to collect movement information [1]–[4]. However, equipping with these sensors is inconvenient as it requires the cooperation of users, and is constrained by energy consumption since these sensors are usually wireless and energy-limited. Another prevalent method utilizes depth cameras [5]–[8]. Some commercial products, like Leap Motion [5] and Kinect [6], can detect human motions with exceedingly high accuracy. However, this camera-based method only works well in the line-of-sight (LOS) scenario. Moreover, it infringes the privacy of users, which is its main limitation.

With the ubiquity of WiFi devices and infrastructures, WiFi-based motion detection methods attract the interests of considerable researchers, e.g., [9]–[13], which solve the problems of privacy as well as requirements for specific environments or sensors. Furthermore, they can detect human motions passively, and are easily deployable. So far, many human motion detection systems are based on Received Signal Strength Indicator(RSSI) which can be easily obtained from WiFi wireless network adapters and smartphones. Many researchers have concentrated on studying RSSI, and relevant approaches [14]–[16] have been proposed

The associate editor coordinating the review of this manuscript and approving it for publication was Ding Xu.

to recognize coarse-grained motions (such as running, walking, and falling).

In recent years, fine-grained Channel State Information (CSI)-based schemes are gradually prevalent. Compared with RSSI, CSI contains detailed amplitude information as well as phase information. Hand gesture recognition, as one critical part of human motion detection, has proliferated in human-computer interfaces (HCI) fields for many years. Some researchers [11], [16] focus on the commercial off-the-shelf (COTS) WiFi devices to receive the WiFi signals, however these devices are not precise enough and even incur extra noise. Some researchers [17] utilize two Universal Software Radio Peripherals (USRP), which form a transceiver pair, to recognize hand gestures, whereas they do not adopt the WiFi protocol. Our gesture recognition system combines WiFi protocol and USRP which is different from previous works and the experimental results show that it has good performance. Combining WiFi with USRP has two advantages. Firstly, when we use USRP to receive the WiFi signals, USRP can correct frequency and phase offsets by means of the algorithm on the Orthogonal Frequency Division Multiplexing (OFDM) receiver [18]. So the CSI values we extracted are more refined than that extracted by the Intel 5300 wireless card, which accounts for the fact that both of the amplitude and phase information of CSI can be used, whereas 5300 NIC-based methods usually only use the amplitude information. Secondly, USRP can receive all signals of 52 subcarriers, whereas only 30 subcarriers can be detected by the Intel 5300 wireless card. Thus the CSI values obtained by our method are more complete and reliable.

In this paper, the main contribution of our work is that we build a gesture recognition system called WiGrus. Details are as follows:

- To the best of our knowledge, we are the first to combine USRP and WiFi for gesture recognition. At the same time, the phase and the amplitude information are creatively utilized for recognition.
- We present a method that directly extracts the CSI values located in the preamble of OFDM frames, based on a modified IEEE 802.11g OFDM receiver.
- We propose a classification algorithm called 2-stage-RF algorithm. Experimental results manifest that our algorithm is superior to other classification algorithms with an average accuracy of 96% in LOS, and 92% in NLOS scenarios.

II. RELATE WORK

On the whole, WiFi-based gesture recognition approaches are divided into two categories: specialized hardware-based and commercial hardware-based [11]. Theoretically, these methods implement gesture recognition by means of capturing the changes caused by body movements in the wireless channel metrics.

A. SPECIALIZED HARDWARE-BASED GESTURE RECOGNITION

The fine-grained radio signals can be collected by specialized hardware devices, such as USRP. WiSee [17] employs the USRP to capture the Doppler shifts of the received WiFi signals, and then uses them to identify 9 kinds of human motion gestures. It works well in 3 different scenarios: LOS, NLOS, and through the wall. AllSee [19] requires a specified analog circuit to extract the amplitudes of the received signals, and then finds out the characteristics of the signals to match the corresponding gestures. However, these methods require complex analog circuits and facilities, and are not integrated with the ubiquitous WiFi Infrastructures, which makes it difficult to be widely used in home and office environments.

B. COMMERCIAL HARDWARE-BASED GESTURE RECOGNITION

Many researchers employ the commercial hardwares for gesture recognition. CARM [13] employs a laptop with 5300 NIC to receive the WiFi signals. Then it extracts the CSI values to build a CSI-speed model and a CSI-activity model which are used to recognize 3 basic human activities. WiGest [16] uses the RSSI of the received WiFi signals to recognize a set of gestures, however, it requires 3 WiFi access points (APs) to achieve excellent performance. References [10], [11], [20]–[22] all utilize a laptop with NIC to obtain CSI values from the received WiFi signals to implement gesture recognition or localization. The accuracy of all the above researches is unsatisfied for the fact that only the amplitudes of CSI values are used in these studies and phases are not taken into account, which are indeed sensitive to environment changes and hard to deal with.

Compared with the existing approaches, WiGrus can extract more fine-grained CSI values from the received WiFi signals, with the USRP working as the 802.11g OFDM receiver [18]. Both amplitude and phase are used to extract gesture features, which markedly promote the precision and robustness of our recognition system.

III. PRIMER

A. CHANNEL STATE INFORMATION

According to the IEEE 802.11g protocol [23], CSI can be obtained in each used OFDM subcarrier (52 in total, including 48 data subcarriers and 4 pilot subcarriers). We denote $X(f,t)$ and $Y(f,t)$ as the transmitted and received signal respectively in the frequency domain on a certain subcarrier, where f denotes the carrier frequency and t denotes the time. Then we have:

$$Y(f,t) = X(f,t) \times H(f,t) \tag{1}$$

where $H(f,t)$ is the channel frequency response (CFR) of wireless channel.

CSI is actually the sample of CFR on each subcarrier. It can express the variations of the WiFi channel. Taking multipath propagation into consideration, CFR is formulated as:

$$H(f,t) = e^{-j2\pi\Delta ft} \sum_{i=1}^{n} a_i(f,t) e^{-j2\pi f \tau_i(t)} \tag{2}$$

where n denotes the number of propagation paths, $e^{-j2\pi\Delta ft}$ denotes the phase shift caused by frequency offset Δf since

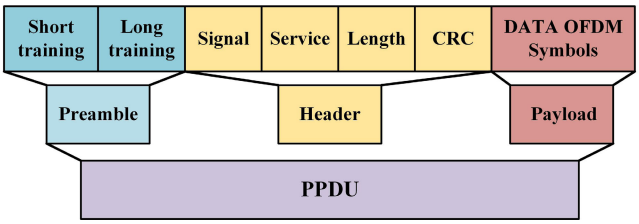


FIGURE 1. 802.11g OFDM frame structure.

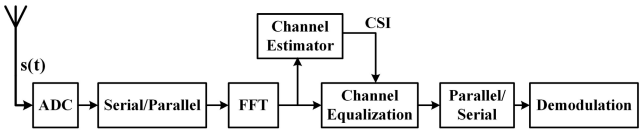


FIGURE 2. Signal processing procedure of the OFDM receiver.

the local oscillators of sender and receiver might work on slightly different frequencies, $a_i(f,t)$ and $\tau_i(t)$ are the complex attenuation factor and propagation delay in the i-th path respectively.

B. OFDM RECEIVER

The original 802.11g OFDM receiver [18] is used to decode the actual data (payload) from the received OFDM frames which traverse across the radio network. As Fig. 1 shows, the OFDM frame has three distinct regions: Preamble, Header, and Payload. The Preamble is used for synchronization and channel equalization, and it includes 10 short training symbols and 2 long training symbols. We make some modifications to extract the CSI values from the long training symbols at the beginning of the received OFDM frames. A minor modification on the OFDM receiver source code allows us to extract more fine-grained CSI data. This is another innovation in this paper. To the best of our knowledge, no one has adopted such a method for gesture recognition yet. Fig. 2 depicts the brief architecture of our OFDM receiver.

In our OFDM system, before decoding the received signal, we extract CSI directly after the channel estimator. More importantly, our OFDM receiver can correct the frequency and phase offsets incurred by the desynchrony of local oscillators between the sender and receiver. Due to space limitation, we refer the interested reader to [18] for details.

C. MULTIPATH MITIGATION

According to previous research [11], the propagation paths of WiFi signals can be split into two categories: non-user reflected paths and user reflected paths. The former, including LOS paths and paths reflected from static objects, is resilient to human movements, and only relates to the static environment. Let $H_s(f)$ represent the aggregated CFRs of all non-user reflected paths. The paths reflected from people can be further divided into two detailed subcategories: the first is that the signals initially reflect from the human body and then return directly to the receiver, the second is that the signals experience further reflections after reflecting from the user before arriving at the receiver in the environment. The amplitudes of the second subcategory in the received signals are relatively lower than that of the first subcategory, since the signals have experienced severe fading after multiple reflections. Therefore, only the first component in the second category is considered as the valid paths reflected from the human body. We denote the aggregated CFRs of the user reflected paths as $H_d(f,t)$. It is formulated as:

$$H_d(f,t) = \sum_{k \in D} a_k(f,t) e^{-j2\pi f \tau_k(t)} \tag{3}$$

where D is the set of the user directly reflected paths. Therefore, Eq. (2) can be rewritten as:

$$H(f,t) = e^{-j2\pi \Delta f t}(H_s(f) + H_d(f,t)). \tag{4}$$

As mentioned in III-B, the phase and frequency offsets of CSI values have been corrected. In other words, the item of $e^{-j2\pi \Delta f t}$ is removed during the signal reception. Thus, the actual received signal is formulated as:

$$H(f,t) = H_s(f) + H_d(f,t). \tag{5}$$

We differentiate the above equation with respect to t. Since the item of $H_s(f)$ is not related to time t, it will be eliminated after the differentiation. Then we derive

$$dH(f,t) = dH_d(f,t). \tag{6}$$

In practice, the first order difference is implemented on the CSI values of the denoised subcarrier to obtain $dH(f,t)$. After that, we can effectively mitigate the multipath effects, and promote the robustness of our system to the environment changes.

IV. THE WIGRUS SYSTEM

WiGrus is a wireless system that exploits WiFi signals to recognize human hand gestures based on the SDR platform. Fig. 3 presents the gesture sketches. All these gestures are the most frequently used in HCI and relevant researches. Fig. 4 shows the flowchart of WiGrus. It starts by using a laptop connected with the USRP to collect CSI from the received WiFi signals in our experimental environment. Then, our system extracts the gesture features from the CSI data after the signal preprocessing as well as noise reduction and multipath mitigation. After processing, each gesture has its unique features. At last, we use the 2-stage-RF algorithm to classify these gestures.

A. CSI COLLECTION

The first step of gesture recognition is collecting CSI data. Based on section III-A and section III-B, CSI is obtained by sampling on CFR which is parsed from the preamble located at the beginning of the WiFi OFDM frame [23]. According to Eq. (1), the CFR value is calculated by

$$H(f,t) = \frac{Y(f,t)}{X(f,t)} \tag{7}$$

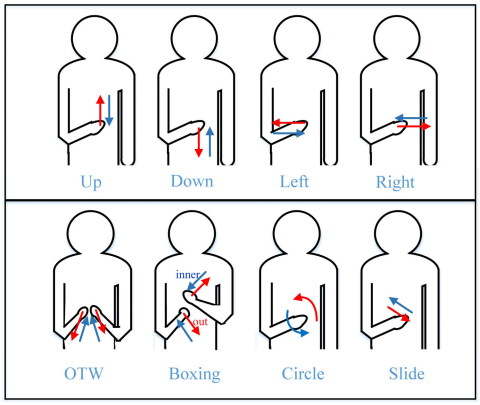


FIGURE 3. Sketches of eight hand gestures. OTW is the acronym of "Open The Window".

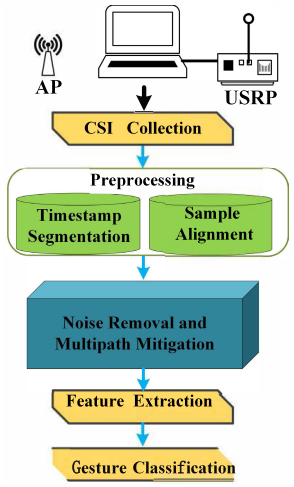


FIGURE 4. The flowchart of WiGrus.

since $H(f,t)$ is a complex number, it can be mathematically defined as:

$$H(f,t) = Ae^{j\varphi} \tag{8}$$

where A and φ are the amplitude and phase of CFR respectively.

The CSI measurement is represented by

$$H = \begin{bmatrix} H_{1,1} & H_{1,2} & \dots & H_{1,52} \\ H_{2,1} & H_{2,2} & \dots & H_{2,52} \\ \vdots & \vdots & \ddots & \dots \\ H_{n,1} & H_{n,2} & \dots & H_{n,52} \end{bmatrix} \tag{9}$$

where n denotes the number of sampling points. Each column denotes a CSI stream. In our OFDM system, the amount of subcarriers is 52. So we collect 52 CSI streams for each gesture measurement in total.

B. DATA PREPROCESSING

Data preprocessing mainly contains two steps: Timestamp segmentation and sample alignment. Due to the discontinuous transmissions of wireless signals, the number of sampling points in each CSI measurement is unequal, which is not conducive to subsequent processes. To address this problem,

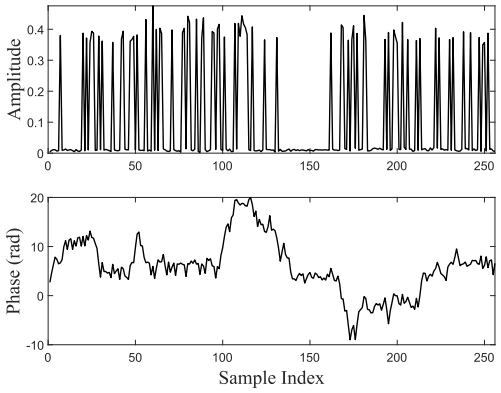


FIGURE 5. Raw CSI amplitude and phase (unwrapped) of circle gesture on the 10-th subcarrier.

each gesture is constrained to be finished within 2 seconds and timestamps are attached to each group of the collected CSI measurements. Thanks to these timestamps, the CSI data can be segmented into pieces by every 2 seconds so that the raw CSI data group can be split into multiple gesture instances. In our experiments, when we use a device (e.g., laptop or smartphone), which connected to the wireless router that has 100M network bandwidth, to download files with maximum speed, more than 1300 CSI sampling points are collected per second. We randomly sample 1024 points per second from these received data to guarantee each CSI measurement has equal sampling points. Then through applying equidistant down-sampling, we obtain the sampling rates of 512, 256, 128, and so on, which are used for comparing the impacts of different sampling rates. Taking the computation and accuracy into account, we set the default sampling rate for subsequent experiments to 128.

C. NOISE REDUCTION AND MULTIPATH MITIGATION

The received CSI data are extremely noisy. Fig. 5 shows the raw amplitude and phase of CSI data. To deal with the noise and multipath effects stemmed from the static environment, some techniques are employed.

One major source of noise comes from the internal state transitions in the sender and receiver, e.g., transmission power change and transmission rate adaption [13]. This noise results in high amplitude impulse and burst noise, which brings a challenge for traditional filters such as low-pass or median filters. So a PCA-based denoising method is adopted, which is proposed in CARM [13]. In WiGrus, we apply the PCA-based denoising method to all CSI streams, and then use a low-pass filter to filter out the rest noise. Subsequently, as section III-C mentioned, the first order difference is carried out on the filtered signal to remove the multipath effects stemmed from the static environment. Finally, we get 52 reconstructed CSI streams, which are called principal components (PCs). Only the second PC among the new PC streams is reserved, because the 2nd has the minimum noise and retains enough information to identify the gestures according to our observation. Fig. 6 shows the first three reconstructed PCs. In the end, we extract the phase and amplitude from the selected PC.

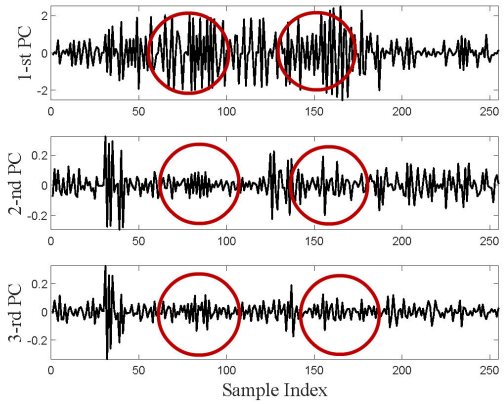


FIGURE 6. The first three PCs of the new reconstructed 52 CSI streams. According to the left circles, the first PC has too much noise; according to the right circles, the third PC loses valid information; so the second PC is the most suitable.

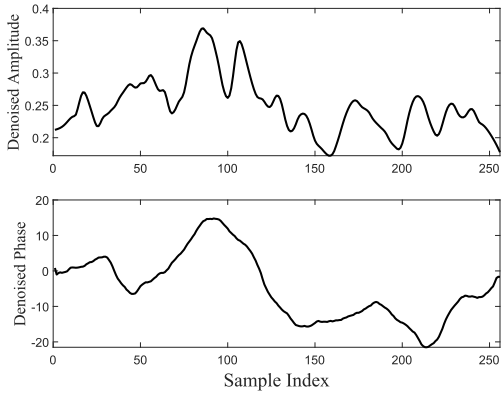


FIGURE 7. The CSI amplitude and phase of selected PC.

As Fig. 7 presents, the processed amplitude and phase have similar variation tendency.

D. FEATURE EXTRACTION

Two kinds of feature extraction mechanisms are proposed. One is extracting the mean values of CSI data after background removal. The other is Discrete Wavelet Transform (DWT)-based statistic feature extraction.

Background CSI data is collected before performing gestures, and then we subtract the background data from the collected gesture data. Before implementing the noise reduction and multipath mitigation, we extract the mean values of amplitude and phase on each subcarrier respectively in advance. Here we obtain 104 mean values of the amplitude and phase from 52 subcarriers in total.

To extract the intrinsic features of the gesture, two aspects should be taken into consideration simultaneously: duration and frequency. Duration represents the persistent period of a gesture, while frequency represents the speed of the user performs a gesture. DWT [24], [25] provides a time-frequency representation of a signal, which exactly satisfies our requirements. In the decomposition stage of DWT, a signal is recursively split into a detail coefficient vector and an approximation coefficient vector for specified levels [25].

In our work, we use Daubechies (db4) wavelet with six levels to decompose our selected CSI stream after the noise reduction and multipath mitigation. Then we use the corresponding six detail coefficient vectors and one approximation coefficient vector to extract features from 7 statistical magnitudes which include the mean, maximum, minimum, maximum energy, minimum energy, variance, and mean of energy. In the end, we obtain 98 DWT-based features. With the previous 104 features, we have 202 features in total.

E. GESTURE CLASSIFICATION

So far, we have obtained a mass of features, then the last step is building our classification model. According to previous researches [9], [11], [12], there are mainly three machine learning algorithms that have been used to classify gestures, including K-Nearest Neighbors (KNN) [26], Support Vector Machine (SVM) [27], and Decision Tree (DT) [28]. In this paper, we propose a 2-stage-RF algorithm based on Random Forest(RF) [29]. In order to verify the performance of our algorithm, we compare it with the most popular classification algorithm, convolutional neural network (CNN).

1) 2-STAGE-RF

Random Forest is an extended variant of Bagging [30] ensemble learning, and all base learners of Random Forest are constructed with classification decision trees. In order to achieve a great integration, the individual learner should have a high accuracy and the difference between the learners should be as large as possible. Thus, bagging and random feature selection are introduced in Random Forest to promote the randomness in the process of constructing decision trees. Bagging increases the diversity of base learners by sampling with replacement on the raw dataset to grow new base learners, while random feature selection exploits different feature subspaces drawn at random from the whole feature space to split the node in the process of tree construction. We implement the combination by averaging the prediction probability of all trees, and the class probability of a single tree is the fraction of samples of the same class in a leaf node.

In our work, more than 200 features are extracted, yet some features may be useless. Excessive features even bring the costs of training speed and model size. Therefore, feature selection is required. In Random Forest, out-of-bag (OOB) estimate is used to evaluate the feature importance which is non-negative and sum-one. We use the feature importance to remove the useless features, then retrain our model with the remaining features. Details of 2-stage-RF are described in Algorithm 1.

Our proposed algorithm mainly includes two stages: in the first stage, training a simple Random Forest model with T_1 trees, and the feature importance of this model is used to select the meaningful features. In the second stage, we retrain the forest model of T_2 trees with the selected features, to generate the final Random Forest model. T_1 does not need to be a large number. In our work T_1 is set to 3 which means the cost of training such a forest is almost negligible, and it works

Algorithm 1 The 2-stage-RF Algorithm for Gesture Classification.

Input:

Gesture dataset D;
Tree number of the first stage: T_1;
Tree number of the second stage: T_2;
Feature subspace size: f;
Feature selection threshold: ε.

Output:

Classification model $\phi_{T_2}(D)$.

1: for $t = 1, 2, \ldots, T_1$.
2: Generate a new dataset D_t from D by sampling with replacement, and the instances that are not sampled constitute D_t^{oob}.
3: Train a decision tree with D_t, denoted as $\phi(D_t)$. For each node in the tree, drawn at random f features from all features, and choose the feature with the least Gini index value to split the node.
4: end for.
5: Combine all trees with the average probability of the same class instances in a leaf node, and obtain model $\phi_{T_1}(D)$.
6: Use all OOB collections $D_t^{oob}(1 \leq t \leq T_1)$ to evaluate the importance of each feature.
7: Update dataset D from the raw dataset, and only reserve the features whose feature importance is larger than ε.
8: Replace T_1 with T_2 and repeat the steps 1-5, producing the model $\phi_{T_2}(D)$.
9: **return** $\phi_{T_2}(D)$.

well for selecting the meaningful features and promoting the overall training speed as well as the recognition accuracy. f is set to $\sqrt{m}$, where m is the number of all features, and ε is set to 0.001. The experimental results will be presented in the next section.

2) CONVOLUTIONAL NEURAL NETWORK

Manual extraction of features often requires prior knowledge and elaborate design. CNN-based feature extraction methods have become highly popular in recent years. We investigate the use of CNN for gesture feature extraction and classification. Since the CSI data have 52 subcarrier signals and each signal has both amplitude and phase information, we now do not use the PCA-based noise reduction methods. Only the background elimination, multipath mitigation, and low-pass filtering are employed to retain information for all subcarriers. Thus, the input size of each gesture instance can be expressed as $(N\_SAMPLE, N\_SUBCARRIER, N\_CHANNEL)$, where N_SAMPLE represents the number of sampling points per gesture instance, $N\_SUBCARRIER$ represents the number of subcarriers, and $N\_CHANNEL$ is the number of channels. For this input size, we can easily use CNN to extract features and classify it, just same as what we do in image processing.

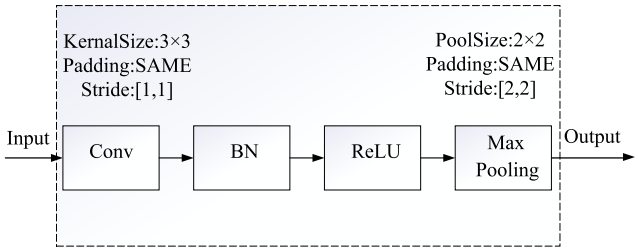


FIGURE 8. The structure of convolutional module. Except for the number of input and output channels of the convolutional layer, the other parameters of each module are exactly the same.

Considering that gesture instances are segmented by timestamp, this implementation is simple but not precise. Thus, we choose to use Random Forest as the final output layer, which was proposed in Deep Neural Decision Forests [31]. Compared with using SVM or logistic regression as the output layer, using Random Forest as the final output layer can effectively deal with large disturbances of input data (because the gesture segmentation according to timestamp cannot accurately guarantee that the current time period corresponds to the beginning and ending of a gesture). This problem can be effectively alleviated by averaging the outputs of multiple trees in the forest.

Our network consists of convolutional modules and probability decision trees, which are connected by fully connected layers. A convolutional module consists of a convolutional layer, a Batch Normalization (BN) [32] layer, a ReLU layer and a max-pooling layer, as Fig. 8 shows. The probabilistic decision tree consists of split nodes and leaf nodes. The inputs of the split nodes are learned by the convolutional modules, and the leaf nodes are the prediction probability of the category to which the gesture instance belongs. The predicted output of the tree is calculated by summing the predicted output of each leaf node.

The network architecture is illustrated in Fig. 9. The details of the network input and output are given in Table 1. In the table, N_BATCH is the batch size that will be used to train the network, and the number "2" in the raw input size indicates the amplitude and phase. The number of leaf nodes N_LEAF is determined by N_DEPTH which means the depth of the tree ($N\_LEAF = 2^{N\_DEPTH}$), and N_TREE represents the number of the probability decision trees. The details about this network design are not the focus of this article. Limited by space, interested readers please refer to [31] for details. The hyper-parameters of the network are determined by a 5-folds cross-validation. In addition to the common parameters of traditional CNN, N_DEPTH and N_TREE are included. The final prediction of the model is the average of the output of all trees. In this paper, we utilize RMSProp [33] as optimizer, and the learning rate is set to 0.001. In the process of gradient descent, RMSProp can prevent severe oscillation and accelerate convergence. The selection of learning rate is based on the cross-validation. With a small learning rate, the convergence process is more stable. Using RMSProp and fixed learning rate is only a rough and simple choice.

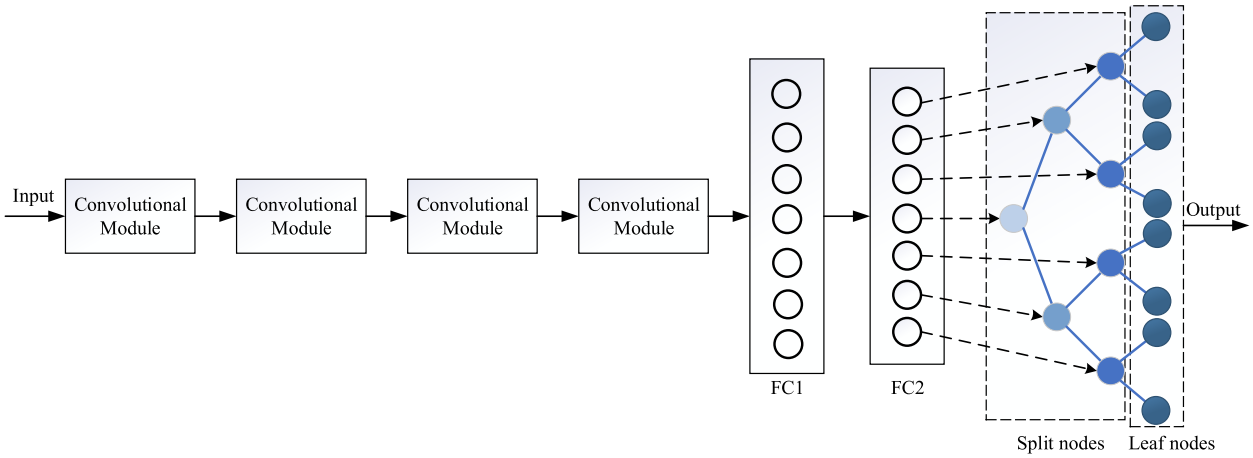


FIGURE 9. The architecture of our CNN model. The predicted output of the tree is obtained by summing the predicted output of each leaf node, and the final output of the model is the average of the outputs of all trees.

TABLE 1. The details of our CNN model. CM means convolutional module, while FC means fully connected layer.

type	channels	weight/keep_prob	input_size	output_size
CM(1)	16	-	(N_BATCH,256,52,2)	(N_BATCH,128,26,16)
CM(2)	32	-	(N_BATCH,128,26,16)	(N_BATCH,64,14,32)
CM(3)	64	-	(N_BATCH,64,14,32)	(N_BATCH,32,8,64)
CM(4)	128	-	(N_BATCH,32,8,64)	(N_BATCH,16,4,128)
FC(1)	-	(8192,1024)/0.5	(N_BATCH,8192)	(N_BATCH,1024)
FC(2)	-	(1024,N_LEAF)/0.5	(N_BATCH,1024)	(N_BATCH,N_LEAF)

Interested readers may consider using learning rate scheduling and Adam optimizer, which may have better performance. The dropout [34] rate of fully connected layer is set to 0.5, and *N_DEPTH* is 5, meaning that *N_LEAF* is 32. In addition, *N_TREE* is 10, and 50 epochs are trained. We extract 20% from the whole dataset as validation set for early stopping.

V. EVALUATION

A. EXPERIMENT METHODOLOGY

Our experimental device consists of two major components. A wireless router (TP-Link TL-WR886N) with 3 antennas is fixed as the transmitter, and the USRP N210 with one SBX daughterboard as well as one antenna (VERT2450) is fixed as the receiver which is connected to a laptop installed with GNU Radio [35]. All the experiments are conducted in the 2.4GHz frequency band with 20MHz bandwidth channels.

We conduct experiments in three different environments. The first one is an office room whose size is $7.35 \times 7.6\ m^2$, and there are multiple tables and chairs in the room. Fig. 10 is the plan of the office. The second is a bedroom whose size is $4.5 \times 5.2\ m^2$, and the room contains a bed, a table, a sofa, and a closet. The third is a corridor without any obstacles and with the size $1.8 \times 11.4\ m^2$. Most of the following experiments are done in the first environment, unless otherwise specified.

We invite 5 volunteers, including four males and one female whose age ranges from 21 to 26. Data collection is done in the LOS scenario of the office, as Fig. 11a shows. The AP is placed on the table at a height of about 1.2 m, and the USRP is on another table with a distance of 1.5 m. The volunteers sit in the middle of the room and perform the gestures, and each volunteer collects 400 gesture instances (eight kinds of gestures, 50 for each gesture). Each volunteer sits in the same position at each collection, and all gestures are collected at different times within 3 weeks, and the positions of the tables and chairs in the room have been changed. In the end, we collect a total of 2,000 gesture instances. The data of all volunteers are mixed and processed simultaneously.

B. OVERALL PERFORMANCE

We compare the performance of various machine learning algorithms on our datasets, including SVM, KNN, Decision Tree, XGBoost [36], 2-stage-RF, and CNN. We randomly extract 20% of the entire dataset as a test set, and then use a 5-folds cross-validation on the remaining dataset for model parameter selection and network structure design. Table 2 shows the recognition accuracy of all algorithms. Obviously, our 2-stage-RF outperforms all other algorithms, reaching a recognition rate of 96.4% (which is equal to the average value of the confusion matrix as Fig. 12 presents). Although the CNN has similar recognition accuracy, the training cost of CNN is quite expensive. Even if we use a NVIDIA-1060 graphics card for training, it takes about four minutes to

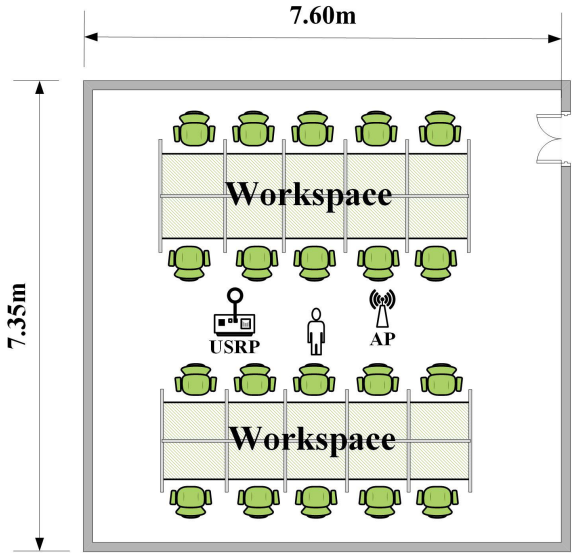


FIGURE 10. The plan of office environment.

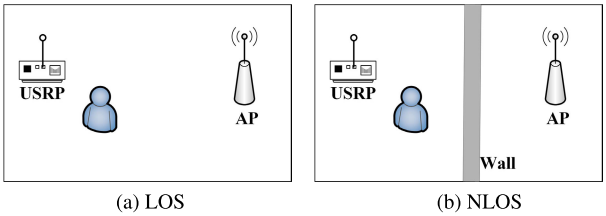


FIGURE 11. Scenario layouts of LOS and NLOS.

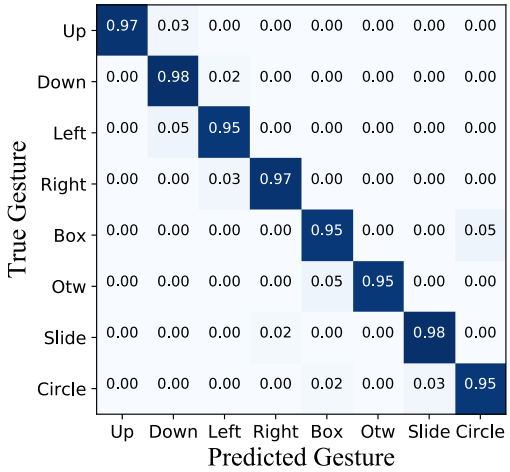


FIGURE 12. Confusion matrix of the proposed 2-stage-RF algorithm in LOS scenario.

train the model to converge and 1.34 seconds to complete the prediction (in which the signal processing time is 0.78s and 0.56s is used for CNN prediction). We tried to use a deeper network, but there was an obvious overfitting. Because of our limited training data, it was not suitable to use too deep or complex networks, such as the network with tens or even hundreds of layers used in computer vision. At the same time, the structure design and hyper-parameter selection of CNN are also very cumbersome, and it takes a lot of time to verify. The network with residual block or dense block may

TABLE 2. The performances of all algorithms.

Algorithm	Accuracy
SVM	0.926
KNN	0.896
Decision Tree	0.822
2-stage-RF	**0.964**
XGBoost	0.912
CNN	0.958

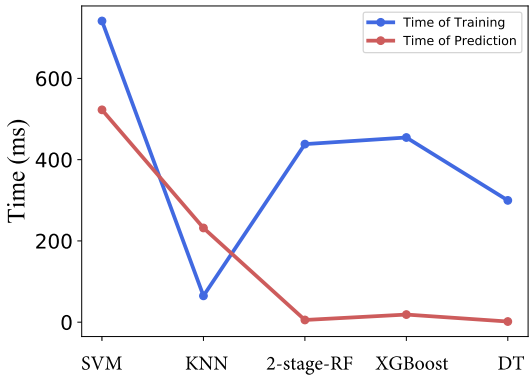


FIGURE 13. Running time comparison of the aforementioned algorithms. Since the training and prediction time of CNN is much higher than other machine learning algorithms, it is not marked in this figure.

solve the problem of overfitting, but it still needs massive data to train the model. Fig. 13 shows the running time of training and prediction of all algorithms. For the 2-stage-RF algorithm, although it needs to extract features manually, each extracted feature is definite and this algorithm can be flexibly applied to gesture recognition in various scenarios. However, when we use CNN, we cannot know exactly what features are extracted each time. CNN is a black-box model with poor interpretability. In summary, our 2-stage-RF is superior in prediction accuracy, time efficiency, and flexibility. The experimental result demonstrates that the WiGrus is robust to environment changes and works well on different individuals.

C. FEATURE SELECTION

In this paper, the number of features extracted manually is 202. Here we only discuss these algorithms other than CNN, because the features of CNN are automatically extracted by the convolutional modules. Fig. 14 plots the curves of recognition accuracy when increasing the number of features on the aforementioned algorithms. In order to compare intuitively, here Random Forest is trained only once with 100 trees, which is different from our proposed algorithm. According to our observation, when the tree number of this forest is 100, it achieves the prominent performance and almost no promotion as the number increases. Obviously, the curve of each algorithm is pretty zigzag, and some features even degrade the accuracy of prediction. Hence it cannot achieve the best performance when all features are used. In contrast, we train another Random Forest with 3 trees first, and then utilize the feature importance of this model to select the valid features. Subsequently, we retrain all models with the selected

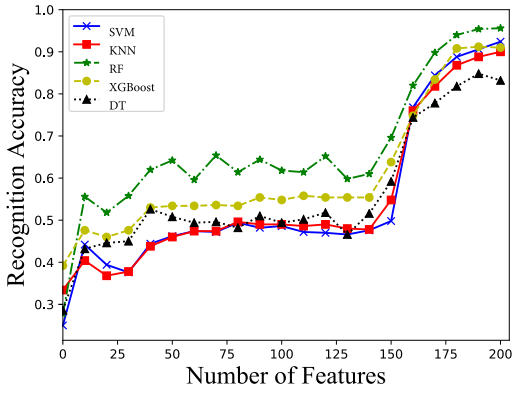


FIGURE 14. Recognition accuracy varies with the increase of feature number.

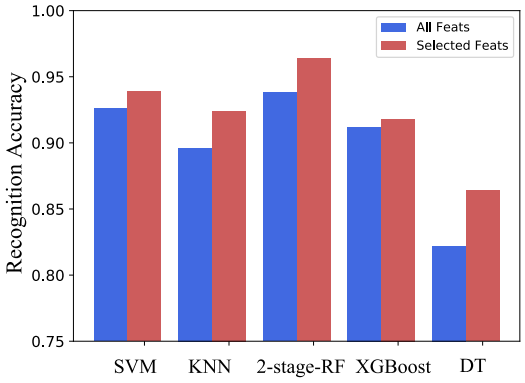


FIGURE 15. Recognition accuracy before and after feature selection.

features, except XGBoost which uses the feature importance of itself. Fig. 15 shows the variations of accuracy before and after feature selection. Obviously, the performances of all models have been improved, and the Random Forest-based model, exactly our proposed 2-stage-RF algorithm, maintains the most outstanding performance.

D. ROBUSTNESS ANALYSIS

We conducted several sets of comparative experiments to verify the robustness of WiGrus under different influential factors. The data of the first three sets of comparative experiments were collected in the LOS scenario of the office environment. The experimental data of NLOS were collected in the NLOS scenario of the office environment, and the data of the last experiment were collected in the bedroom and the corridor respectively. The subsequent experimental data collection was performed by the same volunteer, and we collected additional data for each set of comparative experiments. Subsequent experiments are conducted to verify the robustness of the 2-stage-RF algorithm. Moreover, it performs well under different circumstances.

1) IMPACT OF INTERFERENCE

We first study the impact of interference when there are different numbers of people in our office. We collect the CSI data while people are working on their seats, and the number of people is equivalent to the number of interferences (including

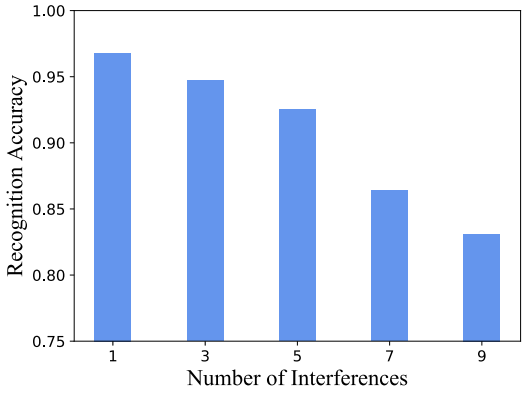


FIGURE 16. Performance comparison under different numbers of people working in the office.

the user who is performing gestures). Data collection was implemented under different interference numbers. In each case, 400 samples were collected (50 samples per gesture). Therefore, extra 2000 gesture samples were collected, and the data collection methods of other experiments were similar. Fig. 16 presents the result of our experiment, and it manifests that our system still has an accuracy of around 92% when there are 5 interferences. Even if the interference number is 9, WiGrus still holds a recognition accuracy of more than 80% which is much higher than that of a random guess. It demonstrates that WiGrus works well in a majority of home and office environments.

2) IMPACT OF SAMPLING RATE

WiGrus utilizes timestamps to split the consecutively received WiFi packets into gesture instances (each of which lasts 2 seconds). We require the volunteer to finish a gesture within 2 seconds. As section IV-A mentioned, we collect the raw CSI measurements with maximum downloading speed, which leads to a high sampling rate. Then through the down-sampling, we obtain the following sampling rates as Fig. 17 shows. The accuracy of WiGrus improves, on the whole, with the increases of the sampling rate. When the sampling rate is 128 (which means 256 sampling points for each gesture), our system has obtained a fairly high precision around 96%. When the sampling rate is 1024, 98% of accuracy is achieved. It obviously demonstrates our system works well even without high occupation on the bandwidth of our network.

3) IMPACT OF DISTANCE

We study the relation between WiGrus's recognition accuracy and the distance from the AP to USRP. Fig. 18 presents that the average accuracy decreases when increasing the distance from the WiFi signal source to the target receiver. The main reason is that one part of our feature extraction is based on the mean of signal amplitude and phase. With the change of distance, these features will be greatly affected, and degrade our recognition accuracy. WiGrus still has an accuracy around 90%, when the distance from the sender to receiver is 5.5 meters, which is sufficient for whole-home

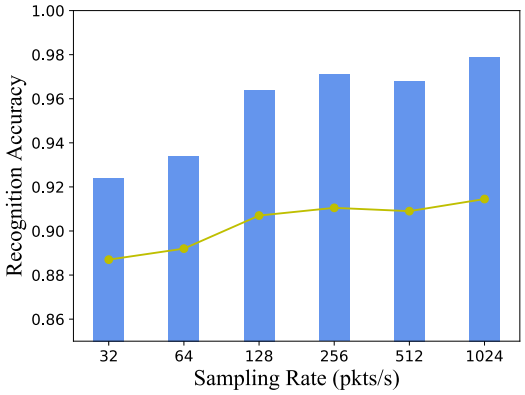


FIGURE 17. Impact of sampling rate on recognition accuracy. The line intuitively presents the variation tendency of recognition accuracy.

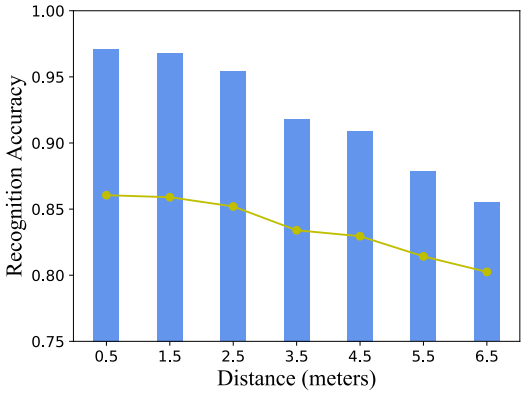


FIGURE 18. Impact of distance on recognition accuracy. The line intuitively presents the variation tendency of recognition accuracy.

gesture recognition in the majority of usage scenarios. Besides, only one AP is employed in our experiments.

4) IMPACT OF NLOS

As Fig.11b depicted, the laptop connected with the USRP, and one AP is placed in adjacent rooms separated by a wall, and the user is on the side of the USRP. Via this layout, we examine the availability of WiGrus in NLOS scenario. Fig. 19 shows the accuracy comparison under LOS and NLOS scenarios. The experimental result indicates that our system reaches an average accuracy of 96% in LOS and 92% in NLOS scenarios respectively. Fig. 20 shows the confusion matrix in NLOS scenario. This is because WiFi signal intensity decays dramatically when it passes through the wall. In the future, we may consider using higher frequency WiFi signals or more receiving antennas to solve this problem. This means WiGrus can be applied in a wide range of fields to recognize human gesture precisely.

5) PERFORMANCE ON DIFFERENT ENVIRONMENTS

We collect data in the bedroom and corridor environments to verify the performance of our system. In the bedroom environment, the USRP and AP are placed on the table and closet respectively, about 4 meters apart, and have about the same height. The volunteer sits on the bed to perform gestures, and the positions of the AP, volunteer, and receiver do not

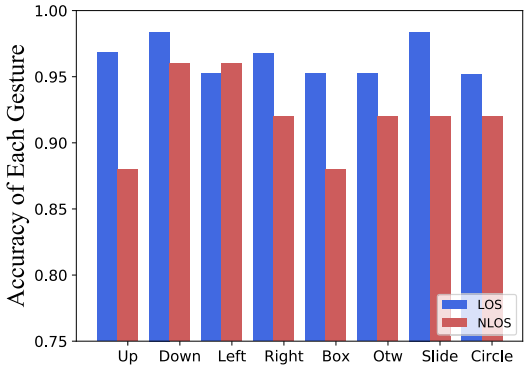


FIGURE 19. Performance comparison of each gesture recognition under LOS and NLOS scenarios.

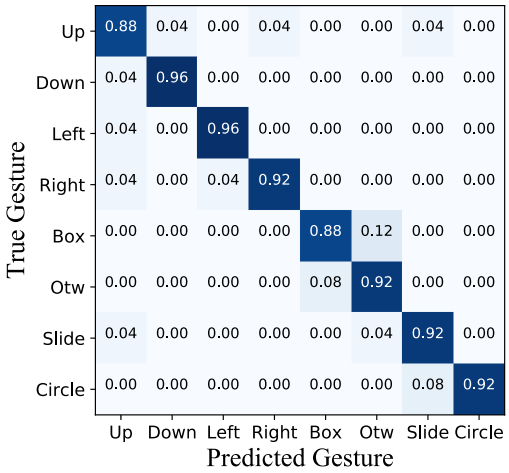


FIGURE 20. Confusion matrix in NLOS scenario.

constitute a straight line. In the corridor environment, the AP is placed on the ground, and the USRP is placed on a table with a height of 1.2 meters, and the distance between them is 7 meters. The volunteer sits between them and does not form a straight line. We use it to simulate this situation in which the signals are reflected multiple times from the wall before arriving at the receiver. Finally, we get a recognition rate of 95% in the bedroom environment, which is close to what we get in the office environment. However, in the corridor environment, the recognition rate is 91%. After the signals are reflected multiple times in the environment, the attenuation of the signals will have a serious impact on the identification. Nonetheless, the conclusions show that our system works well for a variety of environments with a good recognition accuracy.

VI. CONCLUSION

In this paper, we build a WiFi-based gesture recognition system, namely WiGrus. We creatively combine USRP and WiFi to obtain more fine-grained information from the WiFi signal including both amplitude and phase. Furthermore, we propose a 2-stage-RF algorithm for gesture classification. Then we conduct a series of experiments to verify the robustness of our recognition system. The experimental results demonstrate that our algorithm provides the best performance in

terms of both the recognition accuracy and time complexity, compared with other machine learning algorithms including SVM, KNN, Decision Tree, XGBoost and CNN. In addition, it also manifests that WiGrus can precisely identify the gestures under different scenarios and is robust to multi-person interference, the changes of distance and sampling rate. Our proposed WiGrus can recognize gestures with an accuracy of 96% in LOS and 92% in NLOS scenarios respectively in the office environment, and has an accuracy of 95% in the bedroom environment and 91% in the corridor environment respectively. Future works will focus on more advanced IEEE 802.11 protocols, unsupervised gesture classification, and multi-antenna techniques, and we will collect more data to train deeper and more complex neural networks, which may achieve better performances.

REFERENCES

[1] J. Gummeson, B. Priyantha, and J. Liu, "An energy harvesting wearable ring platform for gestureinput on surfaces," in *Proc. 12th Annu. Int. Conf. Mobile Syst., Appl., Services*, 2014, pp. 162–175.

[2] A. Parate, M.-C. Chiu, C. Chadowitz, D. Ganesan, and E. Kalogerakis, "RisQ: Recognizing smoking gestures with inertial sensors on a wristband," in *Proc. ACM 12th Annu. Int. Conf. Mobile Syst., Appl., Services*, 2014, pp. 149–161.

[3] H. Huang, X. Li, S. Liu, S. Hu, and Y. Sun, "TriboMotion: A self-powered triboelectric motion sensor in wearable Internet of Things for human activity recognition and energy harvesting," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4441–4453, Dec. 2018.

[4] J. Wu and R. Jafari, "Orientation independent activity/gesture recognition using wearable motion sensors," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1427–1437, Apr. 2019.

[5] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.

[6] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE Multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

[7] S. Oprisescu, C. Rasche, and B. Su, "Automatic static hand gesture recognition using ToF cameras," in *Proc. IEEE 20th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2012, pp. 2748–2751.

[8] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1297–1304.

[9] Z. Tian, J. Wang, X. Yang, and M. Zhou, "WiCatch: A Wi-Fi based hand gesture recognition system," *IEEE Access*, vol. 6, pp. 16911–16923, 2018.

[10] S. Tan and J. Yang, "WiFinger: Leveraging commodity WiFi for fine-grained finger gesture recognition," in *Proc. 17th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2016, pp. 201–210.

[11] A. Virmani and M. Shahzad, "Position and orientation agnostic gesture recognition using WiFi," in *Proc. ACM 15th Annu. Int. Conf. Mobile Syst., Appl., Services*, 2017, pp. 252–264.

[12] W. He, K. Wu, Y. Zou, and Z. Ming, "WiG: WiFi-based gesture recognition system," in *Proc. IEEE 24th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2015, pp. 1–7.

[13] 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 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 65–76.

[14] Y. Gu, F. Ren, and J. Li, "PAWS: Passive human activity recognition based on WiFi ambient signals," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 796–805, Oct. 2016.

[15] Y. Fu, P. Chen, S. Yang, and J. Tang, "An indoor localization algorithm based on continuous feature scaling and outlier deleting," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1108–1115, Apr. 2018.

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

[17] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proc. ACM 19th Annu. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 27–38.

[18] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, "An IEEE 802.11a/g/p OFDM receiver for GNU radio," in *Proc. ACM 2nd Workshop Softw. Radio Implement. Forum*, 2013, pp. 9–16.

[19] B. Kellogg, V. Talla, and S. Gollakota, "Bringing gesture recognition to all devices," in *Proc. NSDI*, vol. 14, 2014, pp. 303–316.

[20] M. Shahzad and S. Zhang, "Augmenting user identification with WiFi based gesture recognition," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 3, 2018, Art. no. 134.

[21] R. H. Venkatnarayan, G. Page, and M. Shahzad, "Multi-user gesture recognition using WiFi," in *Proc. ACM 16th Annu. Int. Conf. Mobile Syst., Appl., Services*, 2018, pp. 401–413.

[22] N. Yu, W. Wang, A. X. Liu, and L. Kong, "QGesture: Quantifying Gesture distance and direction with WiFi signals," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 1, p. 51, Mar. 2018.

[23] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band*, ANSI/IEEE Standard 802.11, IEEE 802.11 Working Group, 1999.

[24] N. Ricker, "Wavelet contraction, wavelet expansion, and the control of seismic resolution," *Geophysics*, vol. 18, no. 4, pp. 769–792, 1953.

[25] C. S. Burrus, R. A. Gopinath, H. Guo, J. E. Odegard, and I. W. Selesnick, *Introduction to Wavelets and Wavelet Transforms: A Primer*, vol. 1. Upper Saddle River, NJ, USA: Prentice-Hall, 1998.

[26] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN model-based approach in classification," in *Proc. OTM Confederated Int. Conf. 'Move Meaningful Internet Syst.'* Berlin, Germany: Springer, 2003, pp. 986–996.

[27] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, Jun. 1999.

[28] L. Breiman, *Classification and Regression Trees*. Abingdon, U.K.: Routledge, 2017.

[29] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[30] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[31] P. Kontschieder, M. Fiterau, A. Criminisi, and S. R. Bulo, "Deep neural decision forests," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1467–1475.

[32] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167

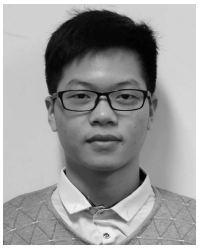
[33] T. Tieleman and G. Hinton, "Lecture 6.5-RMSPROP: Divide the gradient by a running average of its recent magnitude," *COURSERA, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[35] N. Manicka, *GNU Radio Testbed*. Newark, DE, USA: Univ. of Delaware, 2007.

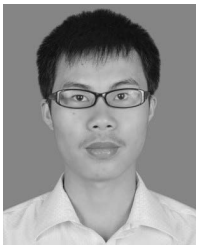
[36] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM 22nd SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.

TAO ZHANG received the B.S. degree in electronic information science and technology from Tianjin Polytechnic University, Tianjin, China, in 2017. He is currently pursuing the M.S. degree with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China, under the supervision of Dr. J. Zhuang. His research interests include machine learning and array signal processing.



TINGYU SONG received the B.S. degree in communication engineering from Nanchang University, Nanchang, China, in 2017. He is currently pursuing the M.S. degree with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China, under the supervision of Dr. J. Zhuang. His research interests include direction-of-arrival estimation, source localization, distributed beamforming, and machine learning.

DAOLIN CHEN received the B.S. degree in electronic science and technology from Southwest University, Chongqing, China, in 2018. He is currently pursuing the M.S. degree with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China, under the supervision of Dr. J. Zhuang. His research interests include direction-of-arrival estimation, source localization, distributed beamforming, and machine learning.



TIAN ZHANG received the B.S. degree in physics from Xiamen University, Xiamen, China, in 2015, and the M.S. degree from the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2019, under the supervision of Dr. J. Zhuang. He is currently with Zhongxing Telecommunication Equipment Corporation. His research interests include direction-of-arrival estimation, source localization, and machine learning.

JIE ZHUANG received the B.Eng. degree from the Chongqing University of Posts and Telecommunications, in 1998, and the Ph.D. degree in electrical and electronic engineering from Imperial College London, U.K., in 2011, where he was a recipient of the U.K./China Scholarship for Excellence Programme. He is currently an Associate Professor with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC). His research interests include array signal processing, MIMO wireless communications, and machine learning.