CARING: Towards Collaborative and Cross-domain Wi-Fi Sensing

A case study for human activity recognition

Xinyi Li, Fengyi Song, Mina Luo, Kang Li, Liqiong Chang, Xiaojiang Chen*, Zheng Wang

Abstract—The quality of a learning-based Wi-Fi sensing system is bounded by the quantity and quality of training data. However, obtaining sufficient and high-quality data across different domains is difficult due to extensive user involvement. We present CARING, a federated-learning-based framework to support collaborative and cross-domain Wi-Fi sensing. A key challenge of CARING is to allow the effective exchange and learning of knowledge across local models that are derived from heterogeneous data sources with uneven data distributions. We overcome this challenge by first extracting the activity-related representation to train local models. The shared global model aggregates received local model parameters and sends them back to individual devices for fine-tuning locally in the deployed environment. By leveraging the crowdsourced knowledge, CARING allows local models to quickly adapt to domain changes using just a few samples seen at test time. We demonstrate the benefit of CARING by applying it to activity recognition across three public datasets collected from 5 environments, 7 deployments, 31 users, and 29 activities. Experimental results show that CARING is highly effective and robust, improving the alternative approach for using single-sourced training data by up to 47%, giving an accuracy of over 80% (up to 100%) for various cross-domain scenarios.

Index Terms—Wi-Fi Sensing, Cross-domain, Collaborative sensing, Federated learning, Human activity and gesture recognition

1 INTRODUCTION

HUMAN activity and gesture recognition build upon Wi-Fi-based sensing techniques promise many exciting application scenarios, from fitness tracking [1] and health monitoring [2] to safety authentication [3]. However, Wi-Fi signals usually carry adverse domain[1] information is unrelated to human activities and gestures. As a result, the classifiers trained with primitive signals in one domain usually undergo drastically drop in accuracy with another domain. This drawback leads to poor user experience, limiting the scale at which Wi-Fi sensing can be operated.

Efforts have been made to address changing domain factors of Wi-Fi sensing – a problem known as *cross-domain* sensing [4], [5], [6]. Existing works in this direction include generating virtual samples to train a "one-fit-for-all" model for different domains [7]. Such data augmentation approaches, while important, are unlikely to cover all possible domain changes using a single set of training data at design time. Other techniques try to convert the test samples into a domain-agnostic representation to minimize the environmental impact [8], [9]. However, it is very hard, if not impossible, to find a single transformation function that works across domains in large-scale deployment. In addition, all existing approaches have a fundamental drawback by requiring a labour-intensive and time-consuming process of collecting training measurements to characterize how wireless channel metrics - such as channel state information (CSI) or received signal strength indicator (RSSI) - are affected by different domains. While collecting such data from each occupant of a home may be feasible, asking each employee or visitor to provide training measurements in a large organization is impractical.

We present CARING[2], a novel cross-domain Wi-Fi sensing approach. CARING is a federated learning (FL) framework [10], designed to aggregate the information collected from distributed users to reduce the labor and time cost of collecting training measurements for learning a high-quality sensing model, while achieving good performance in cross-domain recognition. CARING also enables a service provider to tailor the decision models of a Wi-Fi sensing system to each domain during deployment. One example is personalising and customising a local model for different households and Wi-Fi device deployments. By doing so, CARING thus improves the performance and reliability of a deployed model. Furthermore, since all raw signal data remain on the user devices, CARING respects the user privacy constraint when Wi-Fi sensing is used for sensitive tasks like human activity monitoring in healthcare settings [11].

Fig. 1 gives a high-level workflow of CARING. Here, end-users first download a *global model* from a service provider to their devices such as a smartphone, wireless router or dedicated IoT systems for sensing. CARING then improves

- *Xinyi Li, Fengyi Song, Mina Luo, Kang Li, Liqiong Chang, and Xiaojiang Chen are with Northwest University, China.*
  E-mail: xjchen@nwu.edu.cn
  **Xiaojiang Chen is the corresponding author.*
- *Xinyi Li, Fengyi Song, Mina Luo, and Liqiong Chang are with Northwest University IoT Research Center, China.*
- *Xiaojiang Chen is with International Joint Research Centre for the Battery-Free IoT, China.*
- *Zheng Wang is with the University of Leeds, UK.*
  E-mail: z.wang5@leeds.ac.uk

1. In this work, a domain is a deployment setup including factors like users, deployment environment (e.g., a particular room or office), and device setup. A new domain emerges when one or multiple domain factors change.

2. CARING =**C**ollaborative **a**nd C**r**oss-domain W**i**-Fi Sens**ing**.

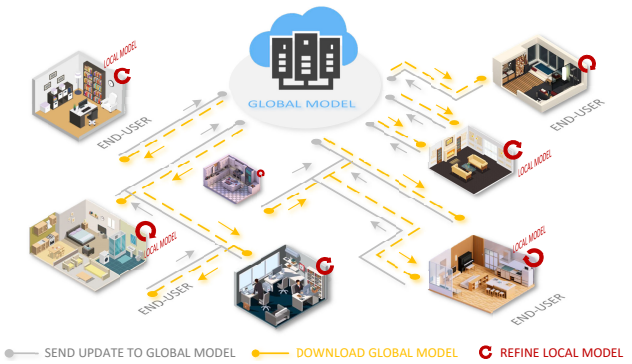


Fig. 1: The workflow of CARING. CARING enables a service provider to use collaborative learning to improve Wi-Fi sensing performance without compromising user privacy.

the *local model* by performing on-device learning from a handful of data samples collected from the target domain in the deployment site. It summarizes the changes made to the local model as a small set of focused weight updates. It then only sends the local model's update to the service provider, which is used to update the global model with data collected from other user models. The updated global model can be later sent back to the user device to refine the local model further, allowing the local model to quickly adapt to domain changes using crowdsourced knowledge. By doing so, CARING allows the service provider to leverage knowledge learned from large-scale, diverse, and real-life deployments to improve the generalization ability of the global model and boost the learning process of a local model. FL is demonstrated to be effective in improving the generalization ability of a decision model while providing personalized services in a number of industry-scale applications like speech recognition and online search [12]. This computing paradigm is increasingly accepted by users as the end-users benefit from improved service. However, few works so far have attempted to develop an FL-based Wi-Fi sensing framework.

To unlock the potential of FL for Wi-Fi sensing requires finding ways to aggregate and utilize the heterogeneous and uneven data distributions collected from different sources to train a global model [13]. Furthermore, when collecting data from different end-users and deployment sites, the labels of the data samples are likely to be imbalanced. This is because end-users may only perform a subset of all activities, e.g., a user may only have {push, pull} while another may only perform {throw, circle}. In the context of WiFi sensing, these challenges can manifest when different activity samples (e.g., gestures) are obtained from different settings (e.g., offices and meeting rooms) and different users. Although the global model can access many model updates learned from different deployments and domains, care must be taken to avoid model overfitting.

CARING overcomes the aforementioned challenges through carefully designed signal processing methods and learning strategies. We start by employing a simple yet effective method to cancel the domain noise to extract activity-related signal representation. We use this method to minimize the impact of heterogeneous data resulting from different domains. Using the activity-related but domain-independent features also permits learning a local model that is resilient to domain changes. Consequently, this maximizes the chance for the global model to reuse model updates obtained from different domains. To overcome the label imbalance issue, we learn a personalized local model for each domain (e.g. a specific device set up in a room). This strategy also boosts the learning process of local model fine-tuning – as we will show later, we need a small number of samples to retarget a global model to a new domain.

Our model for activity recognition is a stacked neural network consisting of convolutional neural networks (CNN) and recurrent neural networks (RNN). This architecture is shown to be effective in recent work on human activity recognition [14]. We use the same architecture for the local and global models to facilitate weight sharing and model fine-tuning. Our model takes as input the preprocessed signal measurements. It then extracts and integrates activity-related representation from the frequency and time domain to perform activity recognition. To avoid overfitting when aggregating local model updates, we introduce a weight adaption mechanism to train the global model. Our scheme is designed to guide the global model to focus on model updates obtained from new domains that are more likely to improve the generalization ability of the resulting model.

We implemented a prototype of CARING and evaluated it on three datasets for activity and gesture recognition deployments, 31 users and 29 activities. Our evaluation represents the largest cross-domain evaluation seen in the literature [4], [15], [16]. The datasets contain CSI data collected from 5 environments, 7 devices. Experimental results show that CARING improves the alternative of using single-source training data by up to 47%, with an accuracy of over 80% (up to 100%) for various cross-domain scenarios. We show that CARING is robust, delivering consistently good performance in challenging situations where the global model has to be trained with imbalanced data samples.

This paper makes the following contributions:

- It introduces an FL framework for collaborative and cross-domain Wi-Fi sensing for human activity recognition.
- It presents a set of new signal processing methods and learning strategies to support collaborative machine learning for Wi-Fi-based sensing tasks.
- It demonstrates how FL can be employed to support cross-domain Wi-Fi sensing with a handful of training samples.

2 RELATED WORK

Wi-Fi-based sensing system. Recent years have witnessed many activity and gesture recognition systems based on Wi-Fi devices [17], [18], [19], [20], [21], [22]. However, the change of domain factors (e.g., users, environments, and device deployments) can lead to skewed signal characteristics which can significantly decrease the recognition system's performance. To make Wi-Fi-based sensing systems practical in the real world, many innovations have been developed to adapt the recognition system in various domains. WiAG [7] can generate virtual samples of any activity in any position and orientation. CrossSense [9] uses a machine-learning model to generate training samples for cross-site sensing. CrossGR [23] uses data augmentation to reduce the cost of collected training data and uses the

deep neural network to extract the user-agonistic activity features from the Wi-Fi channel information. Fido [5] uses Variational Autoencoders (VAEs) to generate synthetic fingerprints and designs a joint classification-reconstruction structure to predict the new users' location. However, such data augmentation approaches are unlikely to cover all possible domain changes using a single set of training data at design time. Other works attempt to find domain-independent features to characterize signals. For example, Widar3.0 [4] designs an environment-independent feature that can generalize the ability of the system. EI [8] leverages the adversarial network to remove environmental factors. However, the complexity of wireless signals in large-scale deployments makes it difficult to find universal features that are friendly to all domains. In addition, the above-mention systems require centralized obtaining sufficient and high-quality sensing data across different domains. Meta-learning-based systems [24], [25], [26] aim to improve the robustness of human activity recognition systems. They all train a model on a large number of raw signal measurements. They require a labor-intensive and time-consuming process of collecting training data. CARING is fundamentally different from these prior meta-learning approaches. It is designed to aggregate the model information (rather than the raw data) to reduce the cost of data collection. Our novelty lies in a crowdsourcing framework for aggregating model information to learn and improve machine-learning models for Wi-Fi-based activity recognition. We propose a set of techniques to address the data imbalance and noise canceling. As far as we know, CARING achieves the largest cross-domain scale with high precision to date. For example, cross datasets means that users, environments, devices, and deployment domains change at the same time. Finally, we can achieve aggregate between different label data to establish a generalized global model, which advances the application of Wi-Fi-based sensing systems in reality.

Human activity recognition with FL. Recently, FL has been applied to human activity recognition [27], [28], [29]. Sozinov et al. [30] applied FedAvg to solve the problem of human activity recognition. However, it does not solve the new challenges brought by human activity recognition so that it achieves slightly worse accuracy compared to the traditional centralized models. ClusterFL [13] is a novel FL system enabling collaborative learning among similar nodes for human activity recognition. Hermers [31] is an FL framework that achieves personalization under data heterogeneity. They solve the heterogeneous data issue, high communication costs and computation efficiency for mobile devices. Compared with these systems, CARING aims to utilize the FL to make the distributed end-users can collaborate to reduce the labor-intensive and time-consuming process of collecting training measurements for learning centralized models. In a potentially concurrent work, WiFederated [32] employs an FL framework for scalable deployment of multi-location CSI-based WiFi sensing systems. CARING advances WiFederated by addressing the label imbalance issue for learning on crowdsourced WiFi model parameters. CARING also leverages effective methods to cancel the domain information from the original Wi-Fi signals.

The model aggregation in FL. Yang et al. [33] aims to reduce communication costs and improve the learning performance of FL by suggesting a synchronous model update strategy and a temporally weighted aggregation method. FedAMP [34] introduces a novel attentive message-passing mechanism to significantly facilitate the collaboration effectiveness between clients without infringing their data privacy. FedProx [35] is designed to tackle heterogeneity in federated networks. These prior studies focus on improving the communication efficiency of FL. They do not address the problem of Wi-Fi-based human activity recognition through crowdsourcing. Unlike the prior work, CARING addresses the cross-domain and label imbalance issues by applying FL to Wi-Fi-based human activity recognition.

3 BACKGROUND

3.1 Problem Definition

In this work, the term *domain* refers to a specific development setup that is independent of the activity (e.g., activities/ gestures). Here, a domain can include factors like users, environments (e.g., the layout of a room) and wireless device setup (like the position, location and distance of two Wi-Fi routers). A new domain can emerge if one or multiple domain factors change. For example, rearranging the wireless device setup, adding new users, and changing the room's layout can result in a new domain. We consider a task to be *cross-domain sensing* when the test domain differs from the domain where the model training data is obtained. Our work considers two types of cross-domain sensing scenarios, *cross single-domain-factor* and *cross multi-domain-factors*. The former refers to the evaluation setup where only one domain factor changes between training and testing. The latter refers to the scenario where multiple domain factors change between training and testing.

In addition, CARING uses crowdsourced knowledge from different data sources to improve the generalization ability of the global model and boost the learning process of a local model. However, the label imbalance issue usually exists in the crowdsourcing scenario where different organizations may share different label information. In detail, each organization may only identify a subset of the activities associated with the global recognition model. As a result, the label distribution in training datasets from different institutions is unbalanced.

3.2 Motivation

3.2.1 Cross-domain issue

In this section, we take one example of *cross single-domain* (i.e., cross-user domain) and one example of *cross multi-domain* (i.e., cross-device and device-location domains) as examples to explore the major limitations of Wi-Fi-based sensing systems. We use the public Widar3.0 dataset to quantify the differences between data samples from different domains by using dynamic time warping (DTW) distances. For cross-user domains, the data is collected by the different users. Only the user factor (i.e., the physical somatotypes of participators) changes. We use data from two users to calculate the cumulative distribution function (CDF) diagram in the same user domain and cross-user domain. For cross-device and device-location domains, the

TABLE 1: Statistics of participants of WiAR dataset [15]. G, H and W stand for the gender, height (cm) and weight (kg).

G	H	W	G	H	W	G	H	W
Male	173	85	Male	180	75	Male	165	65
Female	160	60	Female	170	60	Female	155	65
Male	180	85	Male	175	70			

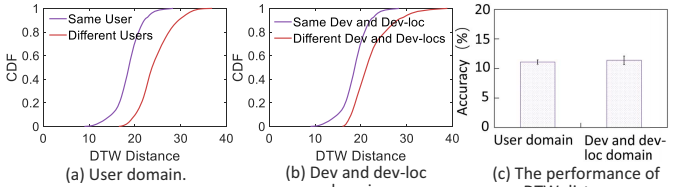


Fig. 2: DTW distance of CSI samples from the same or different domains. Dev and Loc stand for devices and location.

data is collected by the different devices and device locations at the same time. We use the data of two devices (i.e., Rx1 and Rx2 in the Fig. 17) to calculate the CDF diagram in the same or cross-device and device-location domain. For this diagram (Fig. 2), "Same User" or "Same device and device-location" means the data comes from the measurement of a single user or a single device respectively. "Different User" or "Different device and device-location" means the data comes from the dataset of different users or different devices, respectively. We use the CDF of single/two users to show that CSI measurements have a small DTW distance when they are collected from the same domain. By comparison, those taken from different domains will have a larger DTW distance.

To further evaluate the prediction accuracy when using DTW distance as a feature, we use the data of user 1 as the baseline, and use the data of user 2 to user 9 to calculate the DTW distance and test prediction accuracy. Similarly, for cross-device and device-location domains, we use the data of user 1 collected by Rx1 as the baseline data, and the data of Rx2 to Rx6 to calculate the DTW distance and test prediction accuracy. From Fig. 2(c), we can see that the results are disappointing. Because the DTW distance aims to assess how close the data distribution is, it cannot separate the domain information from the CSI data.

For quantified evaluation, we use the CNN-based neural network depicted in Fig. 3 to train a classifier. We consider two scenarios. The first is to train and test the system on data collected from the same domains. The second is to train the system on data collected from some domains but test the system on data from other previously unseen domains. For each case, we apply the grid-search method [36] to find the optimal parameters. We use the WiAR dataset and Widar3.0 dataset to evaluate the cross-user issue and the cross-device and device-location issue, respectively. For the cross user issue, we can obtain the statistics of participants from WiAR [15] as shown in TABLE 1. We use the data of the first 6 users as the training set. The data of the last 2 users are used as testing data. For the cross-device and device-location issue, as shown in Fig. 17, Wi-Fi signals collected in Rx1 and Rx2 are used as the training and testing data. As shown in Fig. 4, the bars are results from different evaluation setups; hence, the numbers are different. Specifically, the left purple bar is the accuracy when the testing data and training data are collected from the same user domain. For example,

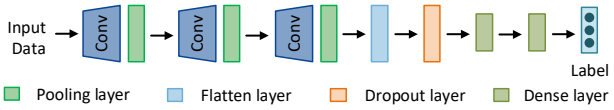


Fig. 3: Architecture of CNN-based classifier.

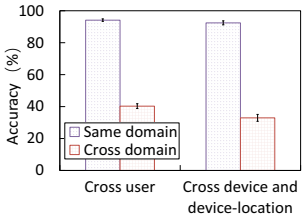


Fig. 4: Cross-domain and non-cross-domain results.

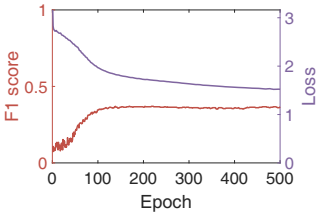


Fig. 5: The performance of label imbalance.

we use the data of user 1 to train and test the model, and we keep other domains (i.e., environment, device, and device location, etc.) unchanged. By contrast, the right purple bar is the accuracy when the testing data and training data are collected from the same device and device-location domain. For example, we use the data of Rx1 (Fig. 17) to train and test the model, and we keep other domains (i.e., environment, users etc.) unchanged. When the classifier is trained and tested on the same domain, it achieves over 90% accuracy for both two cases. However, the accuracy drops by at least 55% in different domains since the learned models are sensitive to data distribution changes, suggesting the model gives poor performance when it is tested on data from a different domain. In addition, we can infer that there is a large accuracy drop in performance even if the differences between domains are small (i.e., Rx1-Rx2 distance is only 0.9 m). The same conclusion is also proved by the RISE [37].

3.2.2 *Label imbalanced issue*

In this evaluation, we consider two public datasets collected from different domains. The first (Widar 3.0) dataset includes 9 activities from 9 users in the classroom, and the second dataset [16] includes 7 activities collected from 8 volunteers in the office. As the two datasets are gathered by independent researchers from different environments, they mimic a typical crowdsourcing setup. We use the combination of these two datasets to evaluate the recognition performance under the label imbalance issue. Specifically, we train a CNN model using the data of 7 users from Widar3.0 dataset and 6 users from the [16] dataset. Each dataset has 2 users uninvolved in training as the testing users to test the system. The CNN model has the same structure and training method presented in Section 3.2.1. The softmax function output is the V-dimensional prediction distribution. We set $V = 16$ in this experiment. The results are shown in Fig. 5. The F1 Score is less than 40%. It indicates that the imbalance label is an issue when training the model using data aggregated from multiple organizations.

4 System Overview

Fig. 6 gives an overview of CARING.

Noise Dispelling Scheme (NDS). We propose a simple but effective method to dispel the domain information from the original Wi-Fi signals before putting them into the

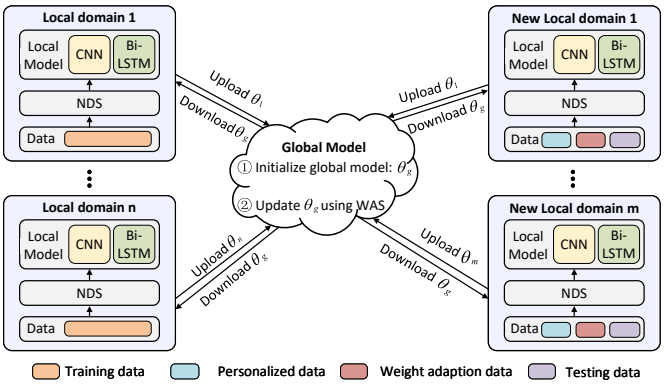


Fig. 6: An overview of CARING.

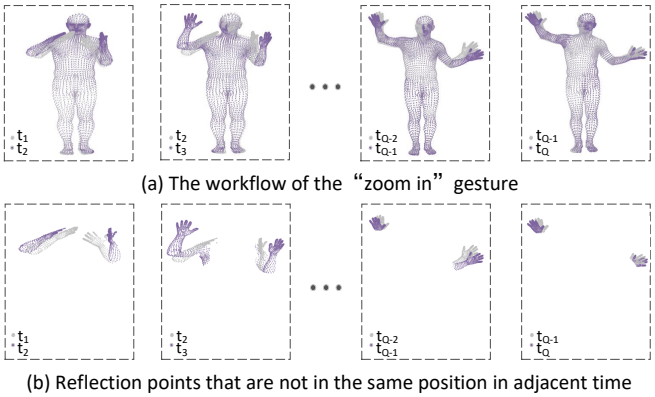


Fig. 7: A toy example of intuition underlying the noise dispelling scheme. The activity from the previous moment is represented by the grey 3D grid, while the activity from the following moment is represented by the purple 3D grid.

deep learning framework to extract activity-related features. More details are in Section 5.2.

Feature Extraction Scheme (FES). We present a hybrid deep learning framework that automatically extracts effective features from processed data by integrating CNN and RNN. The framework is designed to study the frequency domain inter-internal relationships between sub-carriers as well as the time domain intra-internal features of a sub-carrier. It works over the global model and local models as depicted in Section 5.3.

Personalized Local Model. We consider each domain as a subtask and train a unique local model for itself. To achieve personalized predictions in new domains, we first download and reuse the global model's parameters, then collect one or two samples to refine and retune the personalized local model's parameters (Section 5.4.1).

Weight Adaption Scheme (WAS). We apply a weight adaption scheme to make the global model more generalized to new domains. This approach focuses greater emphasis on domains whose parameters contribute more to the new domain. This will be discussed in detail in Section 5.4.2.

5 System Design

5.1 Data Formulation

We provide a general description of the data definition. For simplicity in description, we define an overall dataset D_o with $n+m$ domains, which is the union of the training domain data D_t and the new domain data D_l, i.e., $D_o = D_t \sqcup D_l$. Specifically, the training data has n domains as $D_t = \{D_t^1, D_t^2, ..., D_t^n\}$, and the new data has m domains $D_l = \{D_l^1, D_l^2, ..., D_l^m\}$. Each domain has a different amount of data and correspondingly different labels. The new domain data D_l is divided into three categories, i.e., the personalized data D_{lp}, the weight adaption data D_{lw}, and the testing data D_{lt}. Both D_{lp} and D_{lw} contain a small amount of labeled data (less than 2 samples) for local model personalizing and global model generalization. D_{lt} contains a large amount of unlabeled data to test the performance of CARING.

5.2 Noise Dispelling Scheme

The received Wi-Fi signals in reality are a mixture of multiple signals that travel along different paths and are reflected by a certain number of objects. Specifically, the signal propagation paths reflected by users with different biometric characteristics in different environments are different, thus affecting the basic signal properties such as amplitude and phase. In addition, the fabrication diversity of transceiver devices also induces signal bias. Consequently, certain noise or errors are introduced to the received signals, yielding the characteristic inconsistency in CSI patterns of even the same activity. Traditional methods extract hand-crafted features from the raw measurements to train the classifier. However, these hand-crafted - such as statistical features (e.g., histograms of signal amplitudes [17]) or physical features (e.g., power curves of Doppler frequency shifts [38]) - typically carry domain information that is irrelevant to the activity. This leads to significant degradation of system performance in new domains.

To address this challenge posed by the domain changes, we propose a noise-dispelling scheme (NDS) to remove the domain information included in the original signals. The key observation is that the impacts experienced by two neighboring time series sample points are nearly identical from a domain perspective because of oversampling (i.e., the sample rate is larger than 1,000 packets per second in the activity and gesture recognition [9], [39]), with the primary variation being the dynamic travel path of the signal owing to the motion vector. This setup leads to the adjacent samples having similar characteristics after oversampling, which can be used to dispel the domain information. Fig. 7 displays a toy example to illustrate the observation from the user domain. We construct a 3D mesh of the user performing the "zoom in" activity, which contains a significant number of 3D points that describe the human body's outer surface. The activity from the previous moment is represented by the grey 3D grid, while the activity from the following moment is represented by the purple 3D grid. Wi-Fi signals bounce off these points and reaches the receiver through multiple paths. Fig. 7(a) depicts the workflow of the activity. We can simply extract the motion reflection points (in Fig. 7(b)) by subtracting the 3D mesh points at two consecutive moments since most of the user's body components stay constant.

Suppose the received signal is $S = [s_1, s_2, s_3, ..., s_Q]$, where Q is the number of samples. In theory, s at each moment is comprised of four components, *i.e.*, $s = \{E, U, G, D\}$, where E is the environment noise, U is the noise caused by the static body part of the user, G is the

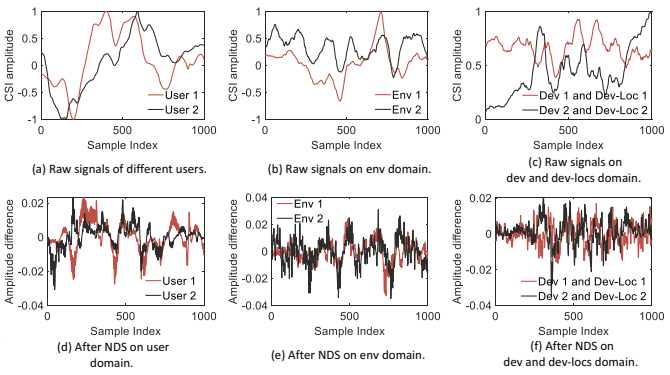


Fig. 8: The performance of noise dispelling scheme. Env, Dev and Loc stand for environment, devices and location.

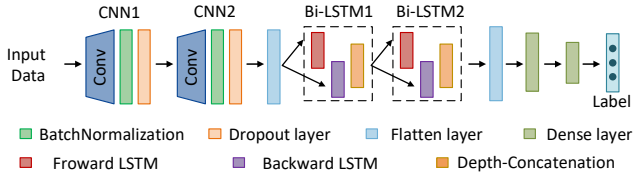


Fig. 9: The structure of proposed deep learning framework.

signal induced by the dynamic body part and corresponds to the motion vector, and D is the total noise introduced by all associated devices, including the transmitter and receiver. Then the x-th and y-th samples are as follow,

$$\begin{aligned} s_x &= D_x + E_x + U_x + G_x \\ s_y &= D_y + E_y + U_y + G_y, \end{aligned} \tag{1}$$

where $x \in [2, Q]$ and $y \in [1, Q-1]$. By calculating the difference between s_x and s_y to remove the common domain factors in Eq. 1, we have

$$s_x - s_y = (D_x - D_y) + (E_x - E_y) + (U_x - U_y) + (G_x - G_y)\,. \tag{2}$$

From the two sample points' perspective, at any adjacent time, the device for collecting Wi-Fi signals is identical. That is, $(D_x - D_y) \cong 0$. Furthermore, the environment noise $(E_x - E_y) \cong 0$ and the noise of user static body shape $(U_x - U_y) \cong 0$. The reason is that they are relatively static between adjacent sample points because of oversampling. Combining all the analysis above, we have:

$$diff\,(s_x - s_y) \approx (G_x - G_y)\,. \tag{3}$$

We conduct three sets of experiments to verify the effectiveness of NDS. The first two deal with the user's body noise and environmental noise respectively. The third one focuses on noise removal in multiple domains (i.e., devices and device locations). In detail, we compare CSI measurements collected from two users who perform the same activity of "zoom in" in the first experiment. And in the second experiment, we collect CSI measurements in two different environments (classroom and hall) when the same user performs the "push & pull" activity with the same device. In the third experiment, we utilize two off-the-shelf mini-desktops equipped with an Intel 5300 wireless NIC to collect measurements from different locations simultaneously. For each above-mentioned experiment, we first interpolated or downsampled the CSI data to 1000 samples to ensure that the testing and training data in each experiment have the same dimension. The raw measurements of three experiments are shown in Fig. 8(a)-(c), respectively. After that, we apply the NDS to each experiment and exhibit the corresponding results in Fig. 8(d)-(f). The raw measurements for each experiment are inconsistent across domains, however, the measurements after the dispelling procedure are highly consistent with only minor deviations. The results demonstrate that our method successfully eliminates the domain's effects and enables domain-independent sensing.

5.3 Feature Extraction Scheme

After the NDS, we were inspired by DeepSense's framework [14] to explore removing the domain-related features and refining activity-related features via integrating CNN and RNN (Fig. 9). The reason is that it is shown to be effective in modeling two types of data: information between various mobile sensors and information between different time segments of one sensor. For our problem, we also need to model two types of information The first is the frequency domain information (i.e., the relationship between various subcarriers in a time segment). The second is the time domain information (i.e., the relationship between different time segments for one subcarrier). Note that all labeled and unlabelled data will pass through these layers to generate their feature vectors.

CNN layer. The COTS Intel 5300 Wi-Fi card with a publicly available driver reports 30 OFDM sub-carriers for each transmission [23], thus a large number of activity patterns are usually generated between neighboring frequencies. To efficiently extract subcarrier correlation in the frequency domain, CARING employs two CNNs, where each CNN includes a convolutional layer, a batch normalization layer, and a dropout layer. The 2D filters in the convolutional layers are used to learn the interaction between subcarriers. In the first and second convolutional layers, the number of filters is set to 128 and 64, respectively. The size of the filters for each convolutional layer is 7×7. All the activation functions are the ReLU. We apply batch normalization [40] after the convolutional layer to decrease internal covariate shift. Then we employ dropout to reduce overfitting and make the network generalize better in practice. The dropout rate is set to 0.3. The CNN can be denoted as F_c:

$$F_c = CNN\,(D_o; \theta_c)\,, \tag{4}$$

where θ_c is the set of all parameters.

RNN layer. In addition to extracting inter-subcarrier characteristics within the frequency domain from the CSI measurements, we also can generate dynamic features of the activity from subcarriers in the time domain. This is because the trajectory of the activity is continuous in the space-time domain. RNN [41] is a powerful architecture that can learn meaningful features from complex dynamics temporal sequences. We choose long short-term memory (LSTM) [42], a special type of RNN that is able to learn problems with long-distance time dependency. Traditional LSTM can extract forward features in time series. The activity information, on the other hand, is reliant not only on the future (forward) but also on the past (backward) moment. Therefore, only using LSTM will inevitably lose some key features and degrade the system's performance.

To address this issue, we use the bidirectional LSTM (Bi-LSTM), which contains two time flows from start to

end and from end to start. We extract the future features from subcarriers via forwarding states and past features via backward states. Specifically, CARING adopts two Bi-LSTMs to model the intra-subcarrier relationship. The first Bi-LSTM has 32 units and the second has 16 units. After CNNs, the inter-subcarriers representations F_c are fed into the Bi-LSTMs F_r,

$$F_r = BiLSTM\left(F_c; \theta_r\right), \tag{5}$$

where θ_r is the set of all parameters.

The output layer. After flattening, two fully connected layers are applied to extract features. To predict the activity label, we use the softmax function to non-linearly map F_r to the V-dimensional prediction distribution, which corresponds to the V labels in our system:

$$y_c = softmax(W_c F_r + b_c), \tag{6}$$

where W_c and b_c are parameters. V various for different datasets. For the labeled data, such as D_t, D_{lp}, and D_{lw}, the predicted distribution y_c^l is compared to the ground truth y via the cross-entropy loss as follow:

$$L_c = \sum\nolimits_{\mathrm{i}=1}^{V} -y^i log y_c^{l_i}. \tag{7}$$

5.4 Federated Representation Learning

5.4.1 Personalized Local Model

When the training data follows independent and identical distributions, federated learning can approach centralized learning successfully and achieve good performance. Due to the heterogeneity of labels and data distribution, federated learning does not perform well for Wi-Fi-based sensing. The diversity of users and devices, as well as variations in furniture locations and device deployments, are likely to add noise to the signal, leading the test and training data distributions to diverge. Although the above approach effectively separates domain-related information from activity-related information, the label imbalance issue persists because different institutions may upload different activity labels to the global model, preventing the building of a general sensing model through crowdsourcing.

To overcome the above challenge, each user is considered as a subtask and a unique local model is trained, thereby they can recognize different label data, respectively. Then, the global model aggregate all the label information to establish a universal model for all end-users. For n local models, in each iteration, they download the parameters θ_g from the global model and update θ_g to θ_i using its local data during the training process which can help the local model to achieve personalized recognition, then upload the new parameters θ_i to the global model, where $i \in [1, n]$. In the same way, m new domains download θ_g from the global model to initialize their local models. But the parameters can not be applied directly to the new domain due to the uneven distribution of data among local domains. However, although each domain has different labels and data distribution, the local activity recognition tasks from different domains are conceptually similar. This is highly similar to the problem of optimization-based meta-learning [43]. So, we can personalize the local model by using a small number of labeled data D_{lp} (less than two samples) from the new domain, and then upload these parameters to the global model to further fine-tune the global model's parameters.

5.4.2 Weight Adaption Scheme

Although the global model can access a large number of model updates learned from different domains, making it more generalizable to new domains, it needs to carefully use the available data to avoid overfitting. This is because local models are obtained from various data sources with heterogeneous and uneven data distributions, which collectively can cause skewed training data distributions for the global model. In fact, not all local models contribute equally to the global model. Some local models have parameters that contribute positively to new domains, and some negatively.

To cope with this problem, we propose a weight adaption scheme to automatically assign weights for each local model. We use D_{lw} (less than two labeled data) from the new domain to test the $m + n$ local models. The prediction accuracy of D_{lw} via cross-entropy loss is:

$$L_{lw_c} = \sum\nolimits_{\mathrm{i}=1}^{V} -y_{lw}^i log y_{lw_c}^i, \tag{8}$$

where $y_{lw_c}^i$ is the predicted distribution of D_{lw}. We set the weight values to 1 for m new domains to make the updating tendency of θ_g more generalized for the new domain. The initialized weight values are 1 for n local training domains. If the prediction accuracy is improved compared to the previous iteration, we keep the weight as 1. Otherwise, we adjust the weight to 0.3. Note that 1 and 0.3 are the best weight values respectively for the positive and negative local model parameters evaluated in our experiment. By doing so, the detrimental impact on the global model's generalization ability is significantly decreased. Then, the weight adaptation parameters are uploaded to the global model as:

$$\theta_g \leftarrow \theta_g + \frac{\sum_{k=1}^{m+n} \lambda_k \left(\theta_i - \theta_g\right)}{\sum_{k=1}^{m+n} \lambda_k} \tag{9}$$

where λ_k is the weight. Note that the evaluation is conducted by using fine-tuned local models. The function of the global model is that aggregate the parameters of local models and use the WAS to automatically assign weights for each local model's parameters. The parameter θ_i of each local domain after fine-tuning can be different.

5.5 The Algorithm of CARING

Assume that the overall dataset involves $n + m$ domains, $i = 1, 2, ..., m + n$, as shown in Algorithm 1. We first put D_o into the NDS to eliminate domain-related information. Note that each domain has a local model. The global model is initialized with the dataset from a selection of training domains. Each of n training domains downloads the current parameter θ_g from the global model and updates the local model with its labeled training data. Each of m new domains makes use of θ_g to establish the local model and retune the parameters using a small amount of personalized data. To determine the combination of the weights, D_{lw} is applied to assess the current local models' performance before uploading the new parameters θ_i to the global model. Then the global model updates θ_g through the WAS. Finally,

Algorithm 1: The CARING Algorithm

Input: $n+m$ domains with overall dataset $D_o = \{D_o^i, i = 1, 2, ..., n+m\}$.
Output: a generalized global model and personalized local models for each domain.
Initialization: Put D_o into the NDS. Initialize the global model with a subdataset of D_t to obtain initial network parameters θ_g;
Training:
for *round=1 to MaxRound* **do**
  for *i=1 to n* **do**
    Download θ_g from the global model;
    Train j (≥ 1) epochs on the local model with D_t^i, get locally updated θ_i;
    Upload θ_i to the WAS;
  end
  for *i=1 to m* **do**
    Download θ_g from the global model;
    Fine-tune θ_g with D_{ip}^i, get locally updated θ_i;
    Upload θ_i to the WAS;
  end
  Update $\theta_g \leftarrow \theta_g + \frac{\sum_{k=1}^{m+n} \lambda_k(\theta_i - \theta_g)}{\sum_{k=1}^{m+n} \lambda_k}$, where λ_k is the weight;
  Return θ_g to the global model;
end

TABLE 2: Signal features used in classical CSI analysis.

Domain	Features
Time	min, max, min/max 10th/90th, variance, mean, skewness standard deviation, kurtosis, q-quantiles (q=0.25, 0.5, 0.75), inter-quartile range, etc. over a time window.
Frequence	domain-frequency ratio, energy, FFT Peaks, etc.

a personalization and generalization procedure is carried out to produce personalized local models and a generalized global model for all domains.

6 IMPLEMENTATION

6.1 Data Preparation

Widar3.0 dataset (Dataset 1) [44]. Zheng et al. collect thousands of CSI measurements for 22 frequent daily activities. There are 17 users, 3 environments, and 6 devices involved in these measurements. The 22 activities include two categories, the first category is common to hand activities, such as push&pull, sweep, clap, slide, draw-O, zigzag, and so on. and the second category is complex and semantic activities (i.e., drawing number 0∼9). The Wi-Fi packets are collected at a rate of 1,000 packets per second using off-the-shelf mini-desktops equipped with an Intel 5300 wireless NIC. The transmitter activates one antenna and works in the monitor mode, on channel 165 at 5.825 GHz. We use dataset of the first category to evaluate our system.

WiAR dataset (Dataset 2) [15]. Guo et al. collect CSI measurements of 16 activities from 10 volunteers in a 6 m $\times$ 10 m meeting room with a small number of office tables and chairs. The 16 activities include three categories: upper-body, lower-body, and whole-body activities. The upper body activities include horizontal arm waves, two hands waves, tossing paper, draw tick, phone calls, draw-X, hand claps, high arm waves, drinking water, and high throw. The lower body activities include forward kick and side kick. The whole-body activities include squatting, sitting down, bending and walking. They use 20 MHz bandwidth with 30 subcarriers in 5 GHz.

[16] dataset (Dataset 3). Yousefi et al. collect Wi-Fi data in an indoor office area. The dataset includes 8 persons and 7 activities (bed, fall, walk, run, sit down, stand up, and pick up). The receiver is equipped with a commercial Intel 5300 NIC with a sampling rate of 1 KHz.

6.2 Implementation and Evaluation Platforms

We implement CARING using python 3.6.5. The model uses Tensorflow v.1.10 as the back end and is implemented using the deep learning library encoding of keras v.2.2.0 and keras-contrib v.2.0.8. The hardware platform we used was a cloud server equipped with 2.4 GHz Intel(R) Xeon(R) E5-2620 V3 CPU, 64GB RAM, and Titan XP GPU. Run Centos 7 operating system on this platform.

7 MICRO-BENCHMARK

To verify the effectiveness of the NDS, FSE, and WAS, we take cross-user recognition as an example to run the following three benchmark experiments.

Experiment Setup. For Dataset 1, we use the data (9 users $\times$ 9 activities $\times$ 5 trails) collected in the classroom. The local model is trained using the data of 7 users, and 3 of them are utilized to initialize the global model. For Dataset 2, we use the data of 16 activities (each with 10 trails) performed by 8 users. The local model is trained using the data of 6 users, and the global model is initialized using the data of 3 of them. Dataset 3 (8 users $\times$ 7 activities $\times$ 10 trails) has the same allocation as Dataset 2. For all datasets, we use the data of 2 users who were not engaged in the training process as new domains to test the system performance.

Verification of Noise Dispelling Scheme. We compared the performance between the classical CSI analysis and NDS. For classical CSI analysis, the input features from both the time and the frequency domains are given in TABLE 2. From Fig. 10, we can clearly see that the performance of classical CSI analysis is disappointing (i.e., less than 31.7%) because these features include domain factors. In addition, compared with w/o NDS, w NDS can significantly improve the system performance on three datasets, which are 22.2%, 44.7%, and 13.1%, respectively.

Verification of Feature Extraction Scheme. We compare our deep-learning-based FES to 1) CNN method and 2) the combination of CNN and forward LSTM (CNN+LSTM) approach. Fig. 11 shows the recognition accuracy. The CNN method has a disappointing result with less than 65% accuracy on three datasets. CARING performs best with over 80% (up to 98%) accuracy. The reason is that CARING can extract not only frequency-domain inter-subcarriers features but also time-domain intra-subcarrier features.

Verification of Weight Adaption Scheme. We assess the system performance in three scenarios: without the global model, a global model with and without applying WAS, i.e., w WAS and w/o WAS. In the absence of a global model, we use new domain data to test each trained local model, and then we get the average accuracy as the final result. Note that we still use fine-tuning samples to refine each local model's parameters when using them to test the new domain. Fig. 12 depicts the recognition accuracy. For Dataset 1, the WAS increases accuracy by 2.5%. It means that most training domains have a beneficial impact on the new domains, resulting in little improvement of WAS. For

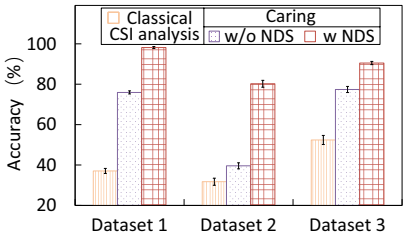


Fig. 10: Results of noise dispelling scheme.

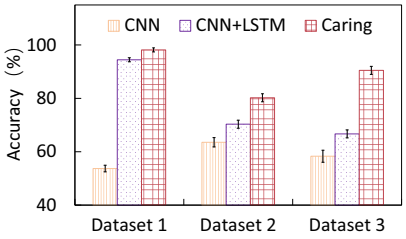


Fig. 11: Results of feature extraction methods.

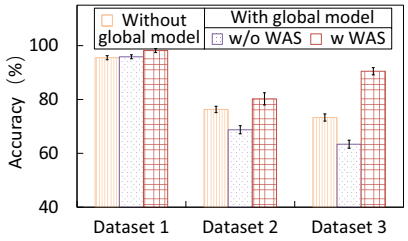


Fig. 12: Results of weight adaption scheme.

TABLE 3: The experimental settings in evaluation. GN, LN, and TN stand for the number of users used to initialize the global model, trained by the local model, tested by the testing domains, respectively. Env, Dev, Loc, and Ori stand for the number of environments, devices, torso locations, and face orientations, respectively.

Task		Dataset	GN: LN: TN (All)	Data (users×activities×trails)	Env	Dev	Loc	Ori
Cross Single Domain	User	Widar3.0	3 : 7 : 2 (9)	9×9×5	1	1	1	1
		WiAR	3 : 6 : 2 (8)	8×8×10	1	1	1	1
		[16]	3 : 6 : 2 (8)	8×7×10	1	1	1	1
	Loc	Widar3.0	3 : 9 : 9 (18)	18×9×5	1	1	2	1
	Ori	Widar3.0	3 : 9 : 9 (18)	18×9×5	1	1	1	2
Cross Multiple Domains	Dev and Dev-loc	Widar3.0	3 : 9 : 9 (18)	18×7×5	1	2	1	1
	User and Env	Widar3.0	3 : 9 : 1∼4/6 (10∼13/15)	(10∼13/15)×5×5	3	1	1	1
	Dataset	WiAR & [16]	3 : 8 : 1∼8 (9∼16)	(9∼16)×2×10	2	2	2	2
Label Imbalance Issue		Widar3.0 & WiAR	6 : 13 : 4 (17)	17×24×(5/10/15)	2	2	2	2
		Widar3.0 & [16]	6 : 13 : 4 (17)	17×16×(5/10)	2	2	2	2
		Widar3.0 & WiAR & [16]	9 : 19 : 6 (25)	25×29×(5/10/15/20)	3	3	3	3
Testing User Diversity		Widar3.0	3 : 5 : 1∼4 (6∼9)	(6∼9)×9×5	1	1	1	1
		WiAR	3 : 4 : 1∼4 (5∼8)	(5∼8)×16×10	1	1	1	1
		[16]	3 : 4 : 1∼4 (5∼8)	(5∼8)×7×10	1	1	1	1
Training User Diversity		Widar3.0	3 : 3∼6 : 2 (5∼8)	(5∼8)×9×5	1	1	1	1
		WiAR	3 : 3∼6 : 2 (5∼8)	(5∼8)×16×10	1	1	1	1
		[16]	3 : 3∼6 : 2 (5∼8)	(5∼8)×7×10	1	1	1	1
Dev and Dev-loc Diversity		Widar3.0	3 : 9 : 9/18/27/36 (18/27/36/45)	(18/27/36/45)×7×5	1	5	1	1
Activity Diversity		WiAR	3 : 6 : 2 (8)	8×(4/8/12/16)×10	1	1	1	1
Fine-tuning sample Diversity		[16]	3 : 6 : 2 (8)	8×7×10	1	1	1	1
Comparison to prior work		widar3.0	3 : 7 : 1 (8)	8×6×5	1	1	1	1
Different Weight Values		Widar3.0	3 : 7 : 2 (9)	9×9×5	1	1	1	1
		WiAR	3 : 6 : 2 (8)	8×16×10	1	1	1	1
		[16]	3 : 6 : 2 (8)	8×7×10	1	1	1	1

Datasets 2 and 3, performance of w/o WAS is lower than without the global model, this is because when we directly aggregate the parameters using the global model, some local models' parameters may generate negative impacts so that reducing the performance of the system. In addition, by comparing w WAS and w/o WAS, we can see that accuracy is improved by 11.5% and 27%, respectively. These results imply that the WAS can effectively counteract the negative impact of undesirable domains, allowing the system to be more generalized to the new domain.

8 Evaluation

We evaluate the performance of CARING with detailed settings shown in TABLE 3.

8.1 The Performance on Cross Single Domain

Cross Users. We now evaluate the performance of CARING in the cross-user scenario. We use the setting described in Section 7 for Dataset 1 and 3. For Dataset 2, we use the data of 8 activities performed by 6 users to train the system. We then test CARING using two "*unseen*" users in each dataset. Fig. 13 shows that CARING achieves high average accuracies of over 85.42% (up to 98.15%), suggesting that CARING works well for cross-user activity recognition.

Cross User-location. We use Dataset 1 to evaluate the performance of CARING in cross user-location cases. The device deployment is shown in Fig. 17. We use the data in location 1 to train the system and test the system in location 2. Different locations have the same 9 activities with 5 trails done by 9 users. Fig. 14 shows that the average accuracy is over 96.3% for all activities. Therefore, CARING is robust to cross-location activity recognition.

Cross User-orientation. We use Dataset 1 to evaluate the performance of CARING in cross user-orientation cases. The data of nine users at orientation 1 and orientation 2 are used to train and test the model, respectively. The device deployment and activity information are the same as above. As shown in Fig. 15, the average accuracy for all activities are all above 81.48%. It demonstrates that CARING can effectively recognize the activity across different orientations.

8.2 The Performance on Cross Multiple Domains

Cross Device and Device-location. To evaluate the system performance in the cross-device and device-location scenarios, we use Dataset 1, which contains Wi-Fi signals received by two devices (i.e., Rx1 and Rx2). Note that the device and its location changed simultaneously. Fig. 17 shows the device setup, where all users are in location 1, facing orientation 1. As can be seen from Fig. 16, CARING gives good

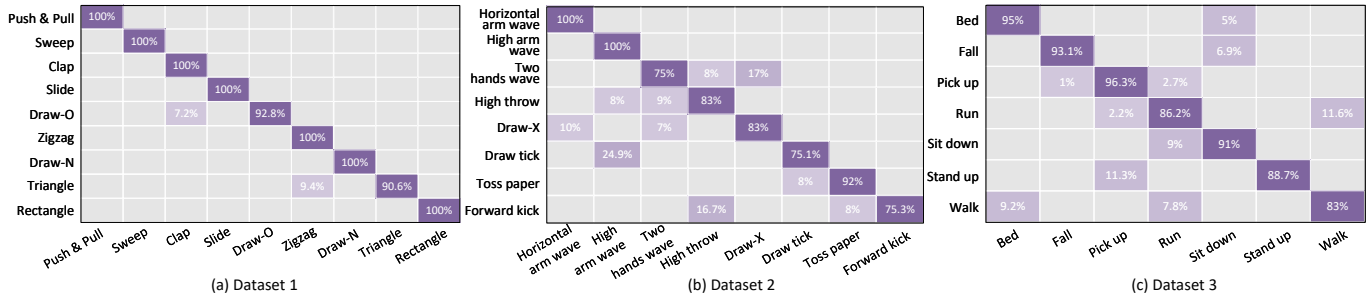


Fig. 13: The confusion matrix of three datasets in cross user case.

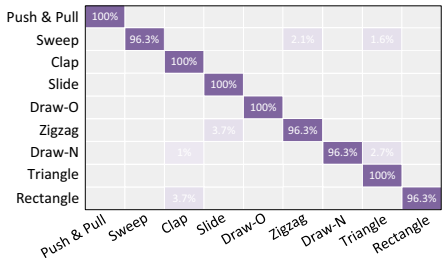


Fig. 14: The confusion matrix of cross user-location evaluation.

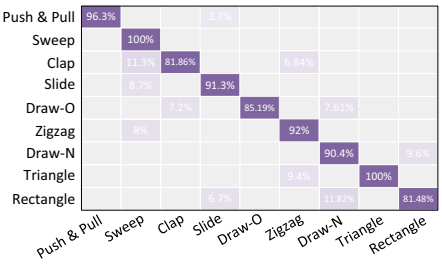


Fig. 15: The result of cross user-orientation.

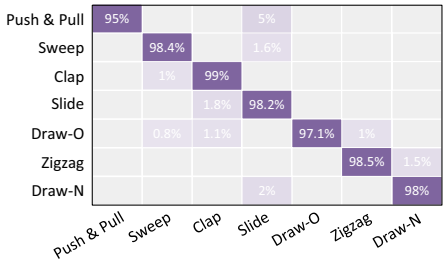


Fig. 16: The result of cross-device and device-location.

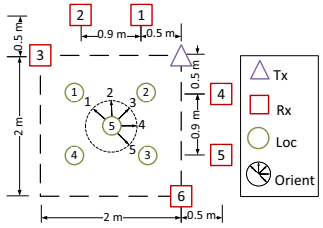


Fig. 17: The device deployment of Widar3.0 dataset.

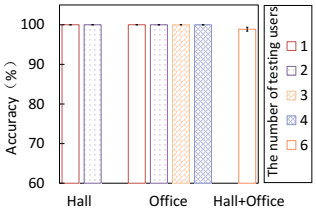


Fig. 18: The performance of cross user and environment.

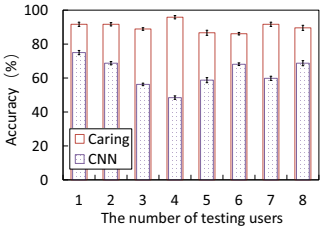


Fig. 19: The performance of cross dataset.

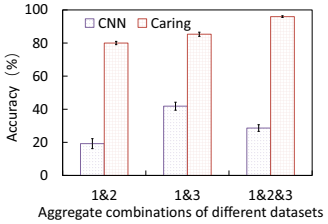


Fig. 20: The performance of label imbalance.

performance for cross-device and device-location recognition, with an accuracy of over 95% for all activities.

Cross User and Environment. We use Dataset 1 to investigate the efficacy of CARING in simultaneous cross-user and environment cases. The system is trained using data collected from 9 users in the classroom. And the data collected from 2 users in the hall and 4 users in the office is utilized to test the system's performance. Note that each of the 15 users has a distinct somatotype. We choose the same activities in 3 environments, such as push & pull, sweep, clap, draw-O, and zigzag, each with 5 trails. In particular, we increase the number of testing users in the hall from 1 to 2, and then from 1 to 4 in the office. In Fig. 18, even when the number of testing users rises from 1 to 4, the average accuracies remain 100%. Furthermore, we evaluate the system using the aggregated data from the hall and office (6 testing users), resulting in an average accuracy reduction of only 1.2%. Overall, CARING works efficiently cross multiple domains and as the number of testing users grows.

Cross Dataset. CARING supports collaboration to improve the quality of cross-domain sensing models. In other words, CARING can leverage the shared knowledge from different data sources to apply to many other new domains while achieving high recognition accuracy. To illustrate the advantage of CARING, we compare CARING against the traditional centralized learning method, namely CNN. The structure and training method of CNN are described in Section 3.2.1. We choose two identical activities from Dataset 2 and 3: walk and sit down. Except for the activity type, the two datasets collect CSI data under entirely different domain factors, such as the user, environment, device, and deployment. The training data consists of 8 users from Dataset 2, whereas the test data consists of 1-8 various numbers of users from Dataset 3. To make a fair comparison, we put the labeled data used to fine-tune the model parameters into the CNN for training. As shown in Fig. 19, among different numbers of test users, CARING consistently performs well with average prediction accuracy of more than 86.11% (up to 95.84%). When compared to CNN, the accuracy is improved by more than 16.7% (up to 47.4%).

8.3 Evaluation on Label Imbalance Issue

To verify that CARING can overcome the label imbalance issue, we aggregate three datasets to jointly train and test our system. As the three datasets are gathered by independent researchers from different environments, they mimic a typical crowdsourcing setup. We take the data of 7 users from Dataset 1, 6 users from Dataset 2, and 6 users from Dataset 3 as training data. The data of 2 users from each dataset are used as testing data. We set the prediction label $V = 29$ for classification because there are two identical activities in Dataset 2 and 3: walk and sit down, and one identical activity in Dataset 1 and 2: clap. In addition, we trained a centralized learning model CNN as a comparison experiment. The structure and training method of CNN is described in Section 3.2.1. For a fair comparison, the personalized data used in CARING are also fed into CNN for training. Fig. 20 shows that CNN has poor accuracy, but

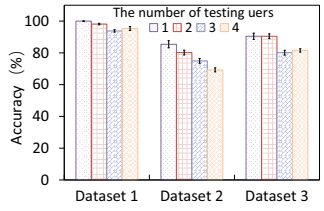


Fig. 21: The performance of testing user diversity.

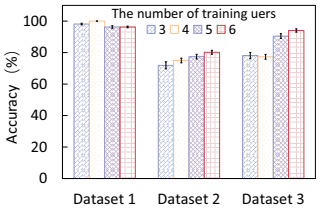


Fig. 22: The performance of training user diversity.

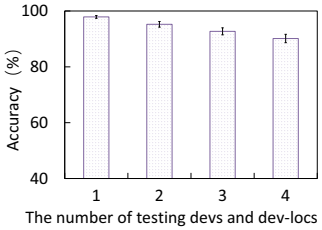


Fig. 23: The result of dev and dev-loc diversity.

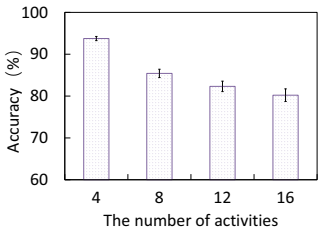


Fig. 24: The performance of activity diversity.

CARING has an average accuracy of over 80% (up to 96%) for all cases. These results suggest that CARING is capable of overcoming the label imbalance issue. Another interesting finding is that the accuracy of the aggregated dataset by three datasets is unexpectedly better than that of any two datasets. While data and label heterogeneity degrade the system's performance, additional training domains increase feature space coverage, resulting in a more generalized global model and more personalized local models.

8.4 The Performance on Diversity

Impact of Testing User Diversity. This experiment studies how the number of users in the testing dataset affects system performance. We run cross-user experiments on three datasets by varying the number of testing users from 1 to 4. The results are demonstrated in Fig. 21. On Dataset 1, the system maintains high accuracy of 95% when there are 4 testing users. On Dataset 3, when the number of testing users is raised to four, the performance drops by less than 10% compared to the best accuracy of over 90%. And on Dataset 2, each additional testing user results in a higher accuracy decline of 5%. This result is anticipated because Dataset 2 is the one that contains the highest number of activities of the three datasets. When more testing users need to be differentiated, the ambiguity of the retrieved features is highest in the three datasets, resulting in rather considerable variances in accuracy. Overall, these results demonstrate that CARING can achieve friendly results to the growth of testing users number.

Impact of Training User Diversity. We take cross user case as an example to evaluate the impact of the number of training data on the system performance. We increase the number of users in each dataset from 3 to 6 as the training data. Fig. 22 depicts that Dataset 1 consistently achieves good performance when the number of training users increases. This manifests that a small number of training domains in Dataset 1 can also obtain a high-precision prediction for new domains. The results of the other two datasets show that the accuracy improves as the number of training data grows. On Dataset 2 and 3, for example, when the number of users increases from 3 to 6, the accuracy rises by nearly 10% and 16%.

Impact of Device and Device-Location Diversity. We analyze the impact of device and device-location diversity on system performance with Dataset 1. To train the system, we take data from nine users received by device 1 and increase the number of testing devices from 1 to 4. Each device is deployed in a unique way. The device deployment is shown in Fig. 17. The results are shown in Fig. 23. When there is only one test deployment, the system obtains the highest accuracy of 98%. When the number of testing deployments is increased to 4, the accuracy drops by around 6%. Overall, CARING is robust to the deployment changes.

Impact of Activity Diversity. We adjust the number of activities in the WiAR dataset from 4 to 16 to see how it affects recognition accuracy. The results are shown in Fig. 24. We can observe that the recognition accuracy decreases as the number of differentiated activities grows. In particular, when just 4 activities are to be distinguished, CARING achieves a recognition accuracy of 93.75%. The accuracy drops to 80.2% when there are 16 activities. This result is reasonable because as more differentiated activities, the average difference between activity features decreases. In a word, these results show that CARING is resistant to the growth of activities number.

8.5 Impact of the number of fine-tuning samples

We evaluate the CARING's performance when the local model is fine-tuned using a different number of samples. We use Dataset 3 in this experiment. We use the data from 8 activities performed by 6 users to train the system. The data of 2 "unseen" users are used as testing data. We run cross-user experiments by varying the number of fine-tuned samples (i.e., 0, 2, 4, 6, 8). We note that the number of fine-tuning samples is the sum of samples in the personalized data and the weight adaption data. For example, a fine-tuning dataset of four samples would have two data points for the personalized and weight adaption datasets, respectively. As can be seen from Fig. 25, when using 0 fine-tuning samples, the accuracy of the model is less than 30%. This is because, without the help of fine-tuning samples, the model parameters of the training domain are difficult to migrate to the new testing domain. In addition, the model accuracy reaches a flat curve (above 96%) when using 4 fine-tuning samples. Using more samples beyond this threshold does not justify the cost of data collection. This experiment also confirms that CARING can use a small number of training samples for retargeting a global model to a new domain. How to achieve "zero-shot" fine-tuning samples is our future work.

8.6 Comparison with the state-of-the-art

We now compare CARING with Widar3.0 [4] and WiFederated [32], the state-of-the-art Wi-Fi-based cross-domain systems. In this experiment, we train each approach on Dataset 1 for every combination of the 7 users provided by the dataset. We then test with the data of the resting person. Fig. 26 shows that CARING outperforms Widar3.0 and WiFederated by improving the accuracy of Widar3.0 by

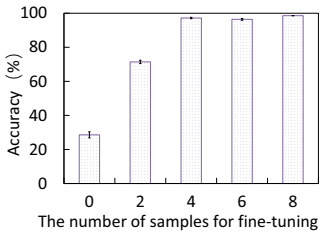


Fig. 25: Results of different fine-tuning sample numbers.

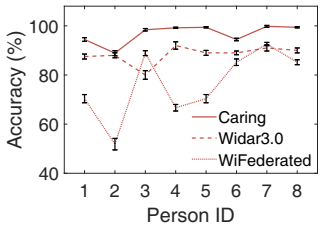
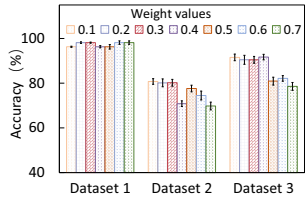
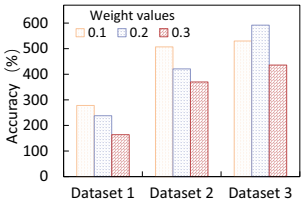


Fig. 26: Comparison to prior work.



(a) The accuracy of different weight values.



(b) Model training epoch on different datasets.

Fig. 27: The performance of different weight values.

20% and WiFederated by 37%, respectively. Furthermore, CARING gives a more robust performance whose accuracy is over 89% across 7 users.

8.7 Effect of Different Weight Values

We use the WAS to avoid overfitting and weaken training domains that do not contribute significantly to the new domain while enhancing those that do. To determine the optimal weight values, we run experiments on all three datasets with various weight values from 0.1 to 0.7 for negative domains. The weight value of the positive domain is set to 1. The experiment setup is the same as Section 7. Fig. 27(a) shows how the accuracy changes as the weight value changes. It is worth noting that the weight does not appear to have a major impact on the system performance in Dataset 1. As we previously analyzed, this is due to the training domain of the dataset being particularly favorable to the new domain, resulting in a little improvement in our scheme. In contrast, the other two datasets illustrate the efficacy of our approach. We set the weight value as 0.3 in our system. The reasons are as follows. On the one hand, when the weight is set to less than 0.3, our approach performs best on all datasets. Using a smaller weight value as the model parameter, on the other hand, requires a longer training epoch since the mode needs more iterations to converge (Fig. 27(b)). Hence, we choose 0.3 as the weight value since it provides a reasonable balance of model accuracy and training overhead.

9 Discussion

CARING is among the first attempts to tackle cross-domain and label imbalance issues via federated learning in Wi-Fi-based activity recognition. Naturally, there is space for improvement and further effort in the future. We go over a couple of topics here.

Environmental robustness. The proposed noise dispelling scheme is mainly to remove relatively static environment noise, such as the static multipath profiles generated by the user's main body and furniture. In fact, our system is tested in a relatively static environment with nearly no interference like the signal changes caused by different domain factors. However, interference is ubiquitous and inevitable in real life, and how to resolve it is a well-known challenging problem [17], [45] for Wi-Fi-based sensing. Therefore, one of the most promising research fields is figuring out how to improve the robustness of wireless sensing in a dynamic environment [46]. In future work, we will try to incorporate some of the most advancements and look for breakthroughs in more complicated and dynamic scenarios.

Deep learning-based feature interpretability. Deep neural networks [14], [47] are shown to be powerful in extracting feature representation. However, these techniques have the drawback of relying on black boxes. The mapping correlation between the input and output is quite complicated. Although providing theoretical proof of the underlying working mechanism can obtain insight into why the model performs well, it is beyond the scope of this work. Recently, there are numerous strategies to remedy this issue [48], [49]. Adopting these methods to explain CARING from the black-box model is our future work.

Other sensing platforms. Although we focus mainly on Wi-Fi-based activity recognition to illustrate the performance of CARING, we believe CARING can be applied to radio frequency (RF) based sensing [3], acoustic sensing [50], [51], mmWave sensing [52], [53] and so on. We need further study on applying CARING to explore new applications based on commercial IoT devices. It would be interesting to see whether CARING can be used in conjunction with specialized sensing devices to improve performance.

10 Conclusion

We have presented CARING, a federated learning-based framework to support low-cost and high-accurate cross-domain Wi-Fi activity recognition. CARING enables a service provider to learn a shared global model using model weights obtained from large-scale deployments. The global model can then speed up the learning process when training a personalized local model at individual deployments. Because CARING does not require the exchange of raw Wi-Fi signal data, it enables collaborative machine learning while preserving user privacy. CARING implements a set of signal processing methods and learning strategies to develop a practical and robust federated learning system. Extensive evaluation performed on three public datasets shows that CARING delivers good and robust performance for cross-domain Wi-Fi sensing, especially in challenging situations where the global model is trained with imbalanced data samples.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (NSFC) under grant agreement 62272388, the NSFC A3 Foresight Program Grant under grant agreements 62061146001 and 62172332, and the Shaanxi International Science and Technology Cooperation Program (2023-GHZD-04 and 2022KW-11). The data and code associated with this paper are openly available at https://github.com/Eleanor-lmn/CARING

REFERENCES

[1] C. Dondzila and D. Garner, "Comparative accuracy of fitness tracking modalities in quantifying energy expenditure," *Journal of medical engineering & technology*, vol. 40, no. 6, pp. 325–329, 2016.

[2] F. Adib, H. Mao, Z. Kabelac, D. Katabi, and R. C. Miller, "Smart homes that monitor breathing and heart rate," in *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, 2015, pp. 837–846.

[3] C. Feng, J. Xiong, L. Chang, F. Wang, J. Wang, and D. Fang, "Rf-identity: Non-intrusive person identification based on commodity rfid devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 1, pp. 1–23, 2021.

[4] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with wi-fi," in *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2019, pp. 313–325.

[5] X. Chen, H. Li, C. Zhou, X. Liu, D. Wu, and G. Dudek, "Fido: Ubiquitous fine-grained wifi-based localization for unlabelled users via domain adaptation," in *Proceedings of The Web Conference 2020*, 2020, pp. 23–33.

[6] R. Gao, M. Zhang, J. Zhang, Y. Li, E. Yi, D. Wu, L. Wang, and D. Zhang, "Towards position-independent sensing for gesture recognition with wi-fi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 2, pp. 1–28, 2021.

[7] A. Virmani and M. Shahzad, "Position and orientation agnostic gesture recognition using wifi," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, 2017, pp. 252–264.

[8] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas *et al.*, "Towards environment independent device free human activity recognition," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 289–304.

[9] J. Zhang, Z. Tang, M. Li, D. Fang, P. Nurmi, and Z. Wang, "Crosssense: Towards cross-site and large-scale wifi sensing," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 305–320.

[10] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[11] J. P. Albrecht, "How the gdpr will change the world," *Eur. Data Prot. L. Rev.*, vol. 2, p. 287, 2016.

[12] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[13] X. Ouyang, Z. Xie, J. Zhou, J. Huang, and G. Xing, "Clusterfl: a similarity-aware federated learning system for human activity recognition," in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, 2021, pp. 54–66.

[14] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 351–360.

[15] L. Guo, L. Wang, C. Lin, J. Liu, B. Lu, J. Fang, Z. Liu, Z. Shan, J. Yang, and S. Guo, "Wiar: A public dataset for wifi-based activity recognition," *IEEE Access*, vol. 7, pp. 154 935–154 945, 2019.

[16] "A survey on behavior recognition using wifi channel state information," https://github.com/ermongroup/Wifi_Activity_Recognition, 2017.

[17] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures," in *Proceedings of the 20th annual international conference on Mobile computing and networking*, 2014, pp. 617–628.

[18] F. Adib and D. Katabi, "See through walls with wifi!" in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, 2013, pp. 75–86.

[19] F. Zhang, K. Niu, J. Xiong, B. Jin, T. Gu, Y. Jiang, and D. Zhang, "Towards a diffraction-based sensing approach on human activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 1, pp. 1–25, 2019.

[20] R. H. Venkatnarayan, G. Page, and M. Shahzad, "Multi-user gesture recognition using wifi," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 2018, pp. 401–413.

[21] S. Tan and J. Yang, "Wifinger: Leveraging commodity wifi for fine-grained finger gesture recognition," in *Proceedings of the 17th ACM international symposium on mobile ad hoc networking and computing*, 2016, pp. 201–210.

[22] O. Zhang and K. Srinivasan, "Mudra: User-friendly fine-grained gesture recognition using wifi signals," in *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, 2016, pp. 83–96.

[23] X. Li, L. Chang, F. Song, J. Wang, X. Chen, Z. Tang, and Z. Wang, "Crossgr: accurate and low-cost cross-target gesture recognition using wi-fi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 1, pp. 1–23, 2021.

[24] Z. Gao, J. Xue, J. Zhang, and W. Xiao, "Ml-wigr: a meta-learning-based approach for cross-domain device-free gesture recognition," *Soft Computing*, vol. 26, no. 13, pp. 6145–6155, 2022.

[25] X. Ding, T. Jiang, Y. Zhong, Y. Huang, and Z. Li, "Wi-fi-based location-independent human activity recognition via meta learning," *Sensors*, vol. 21, no. 8, p. 2654, 2021.

[26] W. Zheng, L. Yan, C. Gou, and F.-Y. Wang, "Meta-learning meets the internet of things: Graph prototypical models for sensor-based human activity recognition," *Information Fusion*, vol. 80, pp. 1–22, 2022.

[27] J. Feng, C. Rong, F. Sun, D. Guo, and Y. Li, "Pmf: A privacy-preserving human mobility prediction framework via federated learning," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 1, pp. 1–21, 2020.

[28] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[29] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *Advances in neural information processing systems*, vol. 30, 2017.

[30] K. Sozinov, V. Vlassov, and S. Girdzijauskas, "Human activity recognition using federated learning," in *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*. IEEE, 2018, pp. 1103–1111.

[31] A. Li, J. Sun, P. Li, Y. Pu, H. Li, and Y. Chen, "Hermes: an efficient federated learning framework for heterogeneous mobile clients," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 420–437.

[32] S. M. Hernandez and E. Bulut, "Wifederated: Scalable wifi sensing using edge based federated learning," *IEEE Internet of Things Journal*, 2021.

[33] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 10, pp. 4229–4238, 2019.

[34] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized cross-silo federated learning on non-iid data." in *AAAI*, 2021, pp. 7865–7873.

[35] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[36] "Gridsearchcv," https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

[37] S. Zhai, Z. Tang, P. Nurmi, D. Fang, X. Chen, and Z. Wang, "Rise: Robust wireless sensing using probabilistic and statistical assessments," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 309–322.

[38] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Device-free human activity recognition using commercial wifi devices," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1118–1131, 2017.

[39] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using wifi signals," in *Proceedings of the 21st annual international conference on mobile computing and networking*, 2015, pp. 90–102.

[40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

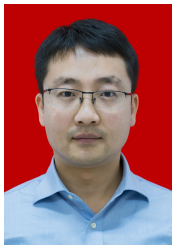
[41] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[43] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," 2016.
[44] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Widar3.0 dataset," http://tns.thss.tsinghua.edu.cn/widar3.0/, 2019.
[45] Y. Zeng, D. Wu, J. Xiong, J. Liu, Z. Liu, and D. Zhang, "Multi-sense: Enabling multi-person respiration sensing with commodity wifi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 3, pp. 1–29, 2020.
[46] B. Xie and J. Xiong, "Combating interference for long range lora sensing," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 69–81.
[47] C. Cummins, P. Petoumenos, Z. Wang, and H. Leather, "End-to-end deep learning of optimization heuristics," in *2017 26th International Conference on Parallel Architectures and Compilation Techniques (PACT)*. IEEE, 2017, pp. 219–232.
[48] A. Dosovitskiy and T. Brox, "Inverting visual representations with convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4829–4837.
[49] O. Bastani, C. Kim, and H. Bastani, "Interpreting blackbox models via model extraction," *arXiv preprint arXiv:1705.08504*, 2017.
[50] L. Wang, X. Zhang, Y. Jiang, Y. Zhang, C. Xu, R. Gao, and D. Zhang, "Watching your phone's back: Gesture recognition by sensing acoustical structure-borne propagation," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 2, pp. 1–26, 2021.
[51] C. CAI, H. PU, P. WANG, Z. CHEN, and J. LUO, "We hear your pace: Passive acoustic localization of multiple walking persons," 2021.
[52] P. S. Santhalingam *et al.*, "mmasl: Environment-independent asl gesture recognition using 60 ghz millimeter-wave signals," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 1, 2020.
[53] C. Wu *et al.*, "msense: Towards mobile material sensing with a single millimeter-wave radio," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 3, 2020.

Xinyi Li received a B.E degree in Internet of Things Engineering from Northwest University in 2017. She is currently working toward a PhD degree in Software Engineering at the School of Information Science and Technology at Northwest University. Her research interests include ubiquitous computing, wireless sensing and mobile computing.

Fengyi Song received a B.E degree in Internet of Things Engineering from Northwest University in 2020. She is currently working toward an M.S. degree in Computer Application Technology at the School of Information Science and Technology, Northwest University. Her research interests include ubiquitous computing, wireless sensing and mobile computing.

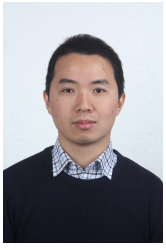
Mina Luo received her B.E. degree in Software Engineering from Northwest University in 2021. She is currently working toward an M.S. degree in Software Engineering at the School of Information Science and Technology, Northwest University. Her research interests include wireless sensing and mobile computing.



Kang Li received a PhD degree in computer software from Northwest University, Xi'an, China, in 2013. He is currently an associate professor with the School of Computer Science and Technology, at Northwest University. His current research interests include intelligent information processing and visualization.

Liqiong Chang received a Ph.D. degree in computer application technology from Northwest University, Xi'an, China, in 2017. She is currently a lecturer at the School of Information Science and Technology, Northwest University, Xi'an. Her current research interests include IoT localization, wireless sensing, and deep learning.

Xiaojiang Chen received a PhD degree in computer software and theory from Northwest University, Xi'an, China, in 2010. He is currently a professor at the School of Information Science and Technology, at Northwest University. His current research interests include RF-based sensing and performance issues in the internet of things.



Zheng Wang is Professor of intelligent software technology at the University of Leeds, UK. His research areas include systems optimization and applied machine learning.