CsiGAN: Robust Channel State Information-based Activity Recognition with GANs

Chunjing Xiao, Daojun Han, Yongsen Ma, and Zhiguang Qin

***Abstract*—As a cornerstone service for many Internet of Things applications, Channel State Information (CSI) based activity recognition has received immense attention over recent years. However, recognition performance of general approaches might significantly decrease when applying the trained model to the left-out user whose CSI data are not used for model training. To overcome this challenge, we propose a semi-supervised Generative Adversarial Network (GAN) for CSI-based activity recognition, CsiGAN. Based on general semi-supervised GANs, we mainly design three components for CsiGAN to meet the scenarios that unlabeled data form left-out users are very limited and enhance recognition performance. 1) We introduce a new complement generator, which can use limited unlabeled data to produce diverse fake samples for training a robust discriminator. 2) For the discriminator, we change the number of probability outputs from $k+1$ into $2k+1$ (here k is the number of categories), which can help obtain the correct decision boundary for each category. 3) Based on the introduced generator, we propose a manifold regularization, which can stabilize the learning process. The experiments suggest that CsiGAN attains significant gains compared to state-of-the-art methods.**

***Index Terms*—Internet of things, WiFi, channel state information, human activity recognition, GANs.**

I. Introduction

Human activity recognition plays an important role in human-computer interaction, and can support many emerging Internet of Things applications, such as smart homes, identification, health care, etc. And many human activity recognition systems have been proposed with different techniques, such as wearable sensors [1], [2], smart phones [3], [4], and cameras [5], [6]. Since human activities can bring about fluctuations of WiFi signals and these variations can be captured by Channel State Information (CSI) from commercial WiFi devices, CSI-based activity recognition has attracted great attention. In particular, it is regarded as a potential alternative method of wearable devices and cameras for human behavior recognition, because the latter can incur extra costs of equipment or invoke privacy concerns of security. By gathering CSI using the collection tool [7], a number of studies have investigated detection of different human behaviors, such as sign language [8], fall [9], gesture [10], keystroke [11], user identification [12], single and multiple human activity [13], [14].

For CSI-based activity recognition, one of the challenges is the performance degradation when applying the trained model to the user whose CSI data are not used for model training [15]. We call this user as the left-out user. The main reason is that individuals with different body characteristics and behavior habits will cause various fluctuations even when performing the same activity. And because there are not any data from left-out users for model training, the trained model cannot capture characteristics of left-out users. Hence the performance of the trained model might obviously decline when applying to left-out users. This problem is pointed out in a number of CSI-based activity recognition models, such as SignFi [8], FallDeFi [9] and CARM [16].

For practical applications, it is hard to collect labeled data of terminal (left-out) users. However, it is feasible to collect their unlabeled data. Because semi-supervised learning uses both unlabeled data and labeled data for model training, the corresponding models can capture characteristics of left-out users by using their unlabeled data, and further improve recognition performance for leave-one-subject-out validation. Among semi-supervised learning methods, GAN-based semi-supervised learning models are competitive with state-of-the-art methods for many fields, such as image classification [17] and haptic material recognition [18]. Hence, the semi-supervised GAN [19] is a potential solution to this problem.

While, like other semi-supervised learning approaches, semi-supervised GANs are typically applied to scenarios that unlabeled data are abundant [20], [21]. However, for CSI-based activity recognition, unlabeled data are generally limited. To enhance the satisfaction of terminal users, the device for CSI-based activity recognition needs to accurately recognize their activities as soon as possible after they buy and turn on the device. Therefore, only a little unlabeled data can be collected in a short time. At the same time, the study [22] shows decreasing the number of unlabeled data results in significant performance decline for semi-supervised learning. Hence, the direct application of semi-supervised GANs will suffer from the shortage of unlabeled data. Therefore, the semi-supervised GAN should be enhanced to address this

Manuscript received April 18, 2019; revised July 11, 2019; accepted August 07, 2019. This work is supported by the National Natural Science Foundation of China (No.61402151 and 61806074), Science and Technology Foundation of Henan Province of China (No.182102210238 and 162102410010). *(Corresponding author: Daojun Han.)*

Chunjing Xiao is with the Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 475004, China, and also with University of Electronic Science and Technology of China, Chengdu 610054, China (e-mail: chunjingxiao@gmail.com).

Daojun Han is with the Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 475004, China (e-mail: hdj@henu.edu.cn).

Yongsen Ma is with College of William & Mary, Williamsburg, VA 23187-8795, USA (e-mail: yma@cs.wm.edu).

Zhiguang Qin is with University of Electronic Science and Technology of China, Chengdu 610054, China.

shortage problem.

To this end, we propose a semi-supervised Generative Adversarial Network (GAN) for CSI-based activity recognition, CsiGAN. On the basis of the semi-supervised GAN [19], we introduce a new generator, and change the output and objective function of the discriminator, as well as propose a manifold regularization for CsiGAN. Specifically, the generators of CsiGAN are composed of the vanilla generator and complement generator. The vanilla generator is the same to the one in the semi-supervised GAN [19] and produces *vanilla fake examples* based on unlabeled samples. Because of limited unlabeled samples, the vanilla generator can only produce samples covering a part of categories. However, discriminators for semi-supervised learning actually need diverse fake samples without missing coverage [23]. Hence, vanilla fake examples are not adequate to train a robust discriminator. Therefore, besides the vanilla generator, we introduce a new generator from CycleGAN [24] as the complement generator. The CycleGAN can transfer real samples in the source domain to another while preserving the style of the target domain. Hence, regarding data of trained users and left-out users as source and target domains respectively, CycleGAN can transfer data of trained users to the one with the style of left-out users. Since data of trained users contain all kinds of categories, these generated samples can cover all the categories and meet the requirement of data diversity. Further, these generated samples are assigned labels according to their source data, and called *labeled fake samples*. As a result, both vanilla fake samples and labeled fake samples are fed into the discriminator.

Besides, we change the outputs of the discriminator from $k+1$ to $2k+1$. The discriminator of the semi-supervised GAN [19] outputs $k+1$ probabilities representing k real categories and one fake category. These outputs can encourage the discriminator to put category boundaries in low-density areas [23]. While, considering new added labeled fake samples, we further enhance the discriminator by adding k outputs representing k fake categories. In other words, labeled fake samples are forced to be put into specific categories in these new added k fake ones, instead of putting all the fake samples into one fake category. In this way, the CsiGAN discriminator has $2k+1$ outputs, which can help the discriminator obtain the correct decision boundary for each category. After the training process, the discriminator will serve as a classifier in the testing process.

To enforce classifier smoothness and stabilize the learning process, we further propose a manifold regularization for CsiGAN. The main idea of manifold is that the relevant subset of data, which comes from near points on the manifold, should be assigned similar labels. According to this idea, we design the specific manifold regularization which is more suitable for the situation with limited unlabeled samples. This term can further improve training stability as well as the final predictive performance.

We illustrate the effectiveness of CsiGAN based on data of fine-grained and coarse-grained behavior recognition, i.e., sign language recognition [8] and fall detection [9]. Experiment results show that when a small number of unlabeled data from the left-out user are used for model training, CsiGAN attains significant gains compared to state-of-the-art semi-supervised models. When there are not any unlabeled data, CsiGAN also obviously outperforms supervised baselines by regarding labeled data from training set as unlabeled data. Besides, we validate the main design choices of CsiGAN with ablation study, and experiments indicate that our designed components can efficiently enhance recognition performance.

We summarize the main contributions of this paper as follows:

- We propose a GAN-based semi-supervised learning model, CsiGAN, to address performance degradation of leave-one-subject-out validation for CSI-based activity recognition. To best of our knowledge, this is the first to apply GANs to CSI-based applications.
- Three components are proposed and incorporated into CsiGAN to deal with the shortage problem of unlabeled data and enhance recognition performance: the complement generator, loss term of k fake categories and new manifold regularization.
- Based on two human activity datasets, experiments in supervised and semi-supervised scenarios demonstrate that our model outperforms the state-of-the-art methods, and prove the effectiveness of our designed components.

II. Preliminaries

In this section, we present a preliminary overview of the problem and GANs for CSI-based activity recognition. And these will serve as background or key design ingredients of our CsiGAN framework.

A. Problem Statement

CSI can capture channel disturbances caused by human movements, and hence be used to recognize human activities by mapping channel distortion patterns to corresponding human activities. However, because different individuals generally have various body characteristics and behavior habits, the collected CSI traces may exhibit significant difference even when performing the same behavior. And recognition performance may severely decline when applying the trained model to the left-out user. For instance, the average accuracy of SignFi [8], a sign language recognition model, reaches 96.68% for 5-fold cross validation, however, it declines to 76.96% for leave-one-subject-out validation. The detection accuracy of CARM [16] decreases from 96.5% to 80% when the targeted user changes from the trained user to the left-out one. And similar trends also appear in the fall detection system, FallDeFi [9]. However, for practical application, the terminal user who buys devices for activity detection is usually the left-out user, because it is hard and costly to label the CSI traces of his activities and use that for training models. In fact, Wang *et al.* [15] point out that one of the important challenges for CSI-based activity detection is how to cope with individual differences and propose universal methods to enhance application on left-out users.

Considering practical application, although it is hard to gather labeled samples of left-out (terminal) users, it is feasible to collect their unlabeled data. When they turn on the device of CSI-based behavior recognition, the machine can record CSI

traces of their activities. Hence, these CSI traces, which are unlabeled, can easily be collected and used for model training.

By taking advantage of these unlabeled data, semi-supervised learning can be used to improve recognition accuracy for left-out users. But this case is different from general semi-supervised learning in two aspects. (1) In semi-supervised learning, the number of unlabeled examples is often much larger [20]. However, for our case, the number of unlabeled samples is usually very small. To enhance the satisfaction of terminal users, the device need accurately recognize their activities as soon as possible after they buy and turn on the device. Therefore, to obtain better performance, only a little unlabeled data can be collected in a short time. (2) For our problem, the unlabeled samples only cover a part of categories, instead of all kinds of samples, because it is hard to obtain some categories of samples even in a long time. For example, some general activities, such as walking and sitting down, are often performed by terminal users and their unlabeled data are easy to collect. But a few special activities, such as fall and loss of balance, may not be performed by terminal users for a long time. Especially for old men, it is dangerous to perform these activities. Hence, to obtain better performance, semi-supervised learning models need to be enhanced for dealing with these problems.

B. GANs for CSI-based activity recognition

Recently, semi-supervised learning methods using GANs have shown promising empirical success. The semi-supervised GAN [19] is built by extending the standard GAN [25]. Let x_l and x_u refer to the labeled sample and unlabeled sample individually. And x_l has the class label y. The semi-supervised model aims at training a classifier simultaneously exploring the labeled sample (x_l, y) and unlabeled sample x_u.

The standard GAN is composed of two components: the generator G and discriminator D. G takes as input a noise vector, randomly generated using an a-priori distribution ($z \in p_z$), and deterministically generates a fake sample $x_v = G(z)$. Then, the real sample x and fake sample x_v are fed into the discriminator D, and D estimates the probability that the input is drawn from real data distribution p_{data}. While, G aims to maximize the probability of the fake samples being classified as real. The overall GAN objective function can be written as:

$$\begin{aligned}\min_G \max_D \quad & \mathbb{E}_{x\sim p_{data(x)}}\left[log(D(x))\right] \\ & + \mathbb{E}_{z\sim p_{z(z)}}\left[log(1-D(G(z)))\right]\end{aligned} \tag{1}$$

To apply to semi-supervised learning, Salimans *et al.* [19] extend the discriminator D of the GAN by changing its input and output. Now D takes three items as input: the labeled sample (x_l, y), unlabeled sample x_u and fake sample x_v. And it outputs $(k+1)$ probabilities over classes (the first k categories for real samples and the $(k+1)$-th category for generated samples). The loss function of D becomes the combination of the supervised and unsupervised parts.

$$\begin{aligned}L_D = & -\mathbb{E}_{x_l,y\sim p_{data}(x_l,y)}\log[p_D(y|x_l, y<k+1)] \\ & -\mathbb{E}_{x_u\sim p_{data}(x_u)}\log[1-p_D(y=k+1|x_u)] \\ & -\mathbb{E}_{x_v\sim p_G(G(z))}\log[p_D(y=k+1|x_v)]\end{aligned} \tag{2}$$

where the first term is the supervised loss, and the last two are the unsupervised ones. And the loss function of G remains unchanged.

Although this semi-supervised GAN achieves remarkable improvement about classification performance. However, it fails to fully consider the situation that unlabeled samples are limited and only cover a part of categories. Hence, the semi-supervised GAN should be enhanced to address the shortage problem of unlabeled samples. As a result, the performance degradation problem of leave-one-subject-out validation becomes the one about how to adapt the semi-supervised GAN to deal with the shortage of unlabeled data. And, we try to propose new components which will be incorporated into the semi-supervised GAN to improve recognition performance for left-out users.

III. CsiGAN Framework

In this section, we present the CsiGAN framework for activity recognition using CSI. First, we illustrate the overview with the relationship of its components. Next, we provide the design of generator, discriminator and manifold regularization in this framework. And finally, we introduce the training algorithm of CsiGAN.

A. CsiGAN Components Overview

We try to use semi-supervised learning to address performance degradation for left-out users in CSI-based activity recognition. However, this case is different form general semi-supervised learning because unlabeled data form left-out users are very limited. Hence, we design a new semi-supervised learning model, CsiGAN. The CsiGAN framework mainly consists of two generators and one discriminator. The relationship of these components is illustrated in Figure 1.

The generators of CsiGAN are composed of the vanilla generator G_v and complement generator G_c. Because of limited unlabeled data, vanilla fake samples x_v produced by G_v can only cover a part of categories, which are not adequate to train a robust discriminator. Therefore, we introduce G_c to produce complement fake samples. G_c takes labeled data $x_{labeled}$ from trained users as input and produces fake samples x_c with the style of left-out users. Since $x_{labeled}$ contain all kinds of categories, x_c can cover all the categories and meet the requirement of data diversity. Furthermore, x_c are associated with labels $y' = y + k$ (here y are the labels of $x_{labeled}$ and k is the number of categories) to form labeled fake samples (x_c, y'). These labeled fake samples can effectively boost the discriminator performance.

For the discriminator, considering new added labeled fake samples, we add k probability outputs and the corresponding loss term in the objective function. As a result, the discriminator classifies inputs into $2k+1$ classes. Here, real labeled samples $x_{labeled}$ are forced to be classified into specific real classes $(1 \cdots k)$, labeled fake samples x_c into specific fake classes $(1' \cdots k')$, real unlabeled samples $x_{unlabeled}$ into any one of real classes $(1 \cdots k)$, and vanilla fake samples x_v into any one of fake classes $(1' \cdots k'{+}1)$. In this way, these outputs

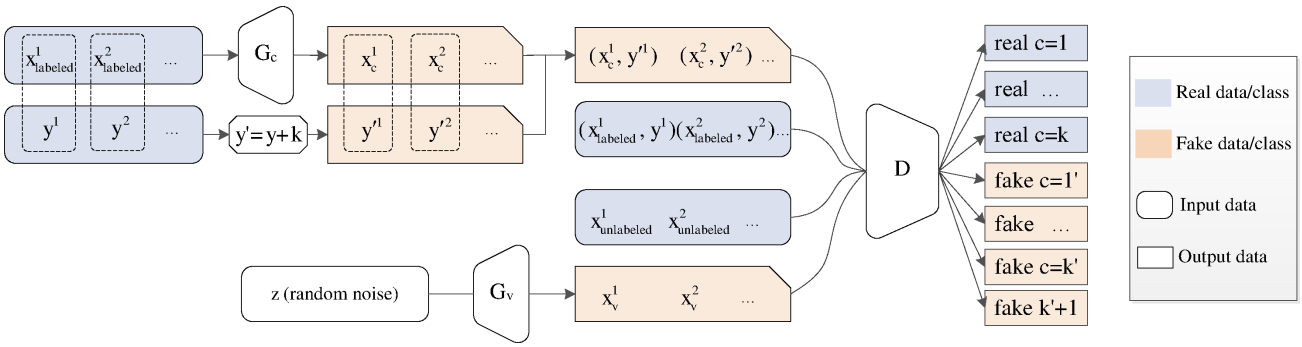


Fig. 1: CsiGAN Framework. Here G_c, G_v, (x_c, y') and x_v refer to the complement generator, the vanilla generator, labeled fake samples and vanilla fake samples individually. And y is the label of the real sample ($x_{labeled}$) and k is the number of categories. During the training process, $x_{labeled}$ is first transferred into fake one (x_c) but with the style of left-out users by G_c, and x_c is associated with the fake label y' to form (x_c, y'). Second, z is adopted to produce x_v by G_v. Third, (x_c, y'), $(x_{labeled}, y)$, $x_{labeled}$ and x_v are used to train the discriminator (D). In the testing process, D will serve as the classifier, which takes test data as input and outputs k probabilities over real classes.

can help the discriminator obtain the correct decision boundary for each category.

In addition, we propose a manifold regularization based on the introduced generator. This term is incorporated into the objective function of the discriminator/classifier and tries to force near points to be assigned similar labels. It can enforce classifier smoothness and further improve predictive performance.

In summary, CsiGAN is different from general semi-supervised GANs in three aspects: (1) the complement generator G_c is introduced into this model to produce complement fake samples. (2) The k probability outputs $(1' \cdots k')$ and corresponding loss term are added for the discriminator. (3) Based on the introduced generator, the manifold regularization is proposed to stabilize the learning process.

B. CsiGAN Generator

The CsiGAN generators are composed of the vanilla generator and complement generator. Since the vanilla generator does not always produce expected fake data, we introduce CycleGAN as the complement generator to produce diverse fake samples.

The vanilla generator usually adopts feature matching to produce fake data [19], [26]. The feature matching tries to force the first-order feature statistic of generated samples to approach the real ones. Unlabeled data are ordinarily used to represent the true distribution. However, for our case, unlabeled data are limited and only cover a part of categories, and consequently cannot represent the overall distribution of all the test set. Correspondingly, the vanilla generator cannot produce fake data covering all the categories. Also, the study [23] demonstrates that given the discriminator objective, the good semi-supervised GAN actually requires a complement generator, which can avoid collapsing with missing coverage and generate diverse bad samples. Therefore, vanilla fake samples produced by the vanilla generator are not adequate to train a robust discriminator.

Since the expected fake samples should be based on the distribution of the test data and no missing coverage [23], we introduce CycleGAN [24] as the complement generator for CsiGAN. CycleGAN can enable the transfer of sequential content from one domain to another while preserving the style of the targeted domain. For example, it can produce compelling image translation results, such as generating photorealistic images from impressionism paintings or zebras from horses.

To produce complement fake samples, we first select the same number of labeled samples with unlabeled samples, and regard labeled and unlabeled samples as the source and targeted domains individually. And these two kinds of samples are fed into CycleGAN to train the model. This model can transfer labeled samples into the new ones but with the style of unlabeled data. After the training, we feed all the labeled samples into the trained model to generate new samples with the style of unlabeled samples. Because labeled samples contain all kinds of categories, these generated samples will cover all the categories. Furthermore, the generated samples are assigned labels according to labels of corresponding labeled samples to form labeled fake samples. And both vanilla fake samples and labeled fake samples will be used to train the discriminator. The objective functions of the vanilla generator and complement generator are the same to the ones in [19] and [24] respectively.

C. CsiGAN Discriminator

In the semi-supervised GAN proposed by Salimans *et al.* [19], the discriminator outputs the probabilities over $k+1$ classes, where real samples are placed in the first k categories and produced fake samples are placed in the $(k+1)$-th category. This $k+1$ output objective brings about the remarkable performance increase, since the produced data stimulate the discriminator to put the category boundaries in low-density areas [23].

For CsiGAN, both vanilla fake samples and labeled fake samples are fed into the discriminator. Hence, to further clarify decision boundaries, we increase the discriminator outputs from $k+1$ to $2k+1$. Here, real samples are placed into the first k categories $(1 \sim k)$, labeled fake samples into the next k categories $(1' \sim k')$, and vanilla fake samples into any fake

category between $1'$ and $k'+1$. In this way, compared with the semi-supervised GAN [19] where all generated data are classified into one fake category, labeled fake samples are put into specific fake categories, which can enable the discriminator to learn the correct decision boundary for each category. Hence, the discriminator can obtain better classification performance.

Formally, let x_l and x_u represent the labeled and unlabeled samples individually, and x_c and x_v denote labeled fake samples and vanilla fake samples generated by the complement generator (G_c) and vanilla generator (G_v) respectively. As well as y refers to class labels of real samples, and y' means fake class labels for labeled fake samples and can be computed by $y' = y + k$. Hence, the objective function of the discriminator is:

$$\begin{aligned}\mathcal{L}_D = &-\mathbb{E}_{x_l,y\sim p_{data}(x_l,y)}\log[p_D(y|x_l, y < k+1)] \\ &- \gamma_1\mathbb{E}_{x_u\sim p_{data}(x_u)}\log[p_D(y < k+1|x_u)] \\ &- \gamma_2\mathbb{E}_{x_c,y'\sim p_G(G_c(x_l),y')}\log[p_D(y'|x_c, k < y' < 2k+1)] \\ &- \gamma_3\mathbb{E}_{x_v\sim p_G(G_v(z))}\log[p_D(y > k|x_v)]\end{aligned} \tag{3}$$

Where γ_1, γ_2 and γ_3 are parameters for adjusting the weights of unlabeled samples, labeled fake samples and vanilla fake samples individually. The first term of $\mathcal{L}_D$ accounts for labeled data in training set to be rightly placed in corresponding one of the first k categories. The second term is designed for unlabeled samples and tries to decrease probabilities of samples being put in fake categories. While the third loss term tries to put labeled fake samples to corresponding fake categories. The fourth encourages the discriminator to distinguish vanilla fake samples from true data.

D. Manifold Regularization

Inspired by studies [17], [26], we propose a manifold regularization for CsiGAN to enhance the performance. The general manifold regularization framework [27] assumes that the data lies in a low-dimension manifold $\mathcal{M}$, and further the classifier f is smooth on this manifold. Therefore, near data on this manifold should be assigned close labels. Algorithms according to this idea can enhance a classifier invariance to the perturbation on the manifold by penalizing its Laplacian norm $\|f\|_L^2 = \int_{x\in\mathcal{M}} \|\nabla_{\mathcal{M}} f(x)\|^2 \mathrm{d}\mathcal{P}_X(x)$. However, for deep neural networks, computing this norm is computationally prohibitive since it needs to compute the Hessian of models with a great number of parameters. And stochastic finite differences are usually adopted to approximate the Laplacian norm for computational efficiency [17], [26]. For example, on the basis of the study [21], Lecouat *et al.* [26] use $\|f(g(z)) - f(g(z) + \epsilon\bar{r}(z))\|_F$ to make approximation, where g refers to the GAN generator and $r(z)$ is an approximation of the manifold gradient at z, and achieve the better performance for semi-supervised learning on both CIFAR-10 and SVHN data. Their approach relies on the commonly held assumption about GANs that the generators of GANs can model the distribution about real samples.

However, for our case, the vanilla generator (G_v) cannot model the data distribution of the left-out user because the unlabeled data only cover a part of categories. Hence, we use the complement generator (G_c) for the manifold regularization. At the same time, the perturbation $G_c(x+\delta)$ and its source $G_c(x)$ should keep the same magnitude and belong to the same manifold to avoid over-smoothing and under-smoothing [26]. Thus, to efficiently control the perturbation, we propose the following approximation:

$$\begin{aligned}&\Omega(f) \approx \\ &\frac{1}{n}\sum_{i=1}^{n}\left\|f\left(G_c(x_l^i)\right) - f\left((1-\alpha)G_c(x_l^i) + \alpha G_c(x_l^i + \bar{\delta})\right)\right\|_F\end{aligned} \tag{4}$$

Where $\delta \sim N(0, I)$ is a tensor with the same shape of x_l. And α is used to adjust the weights of generated samples and random samples. This approximation tries to promote classifier invariance between the produced sample $G_c(x_l^i)$ and the corresponding perturbation $(1-\alpha)G_c(x_l^i) + \alpha G_c(x_l^i + \bar{\delta})$. And the perturbation can keep in the same magnitude of manifold gradients with $G_c(x_l^i)$ by combining the generated sample and random sample. Thus, this regularizer can enhance classifier invariance and further improve classification performance.

E. CsiGAN Training

To enhance classifier invariance, our proposed manifold regularization is incorporated into the discriminator. Therefore, after combining the manifold regularization loss in Equation 4 and the discriminator loss in Equation 3, the final loss function of the discriminator is formulated as:

$$L_D = \mathcal{L}_D + \Omega(f) \tag{5}$$

Furthermore, we can summarize the training process of CsiGAN. As shown in Algorithm 1, the training proceeds by iteratively updating the parameters of CycleGAN, the discriminator and vanilla generator. The algorithm firstly trains CycleGAN using both labeled and unlabeled samples (Line 3-7). To accelerate the training speed, CycleGAN is trained every M_{cycle} epoch (Line 3), instead of every epoch. The default settings and parameters in [24] are used for CycleGAN training (Line 4 and 5). And then the generator G of CycleGAN is saved as the complement generator G_c (Line 6) to generate labeled fake samples which can cover all the categories and have the style of the left-out user. After the training of CycleGAN, the fake samples x_c are generated by G_c and assigned labels y' to form labeled fake samples (x_c, y') (Line 8). Finally, the algorithm updates the discriminator (Line 10) and vanilla generator (Line 12) respectively, and the Adam algorithm [28] is adopted as the optimization method with the default hyper-parameters. When all the training is finished, the discriminator will be regarded as the classifier to perform classification on the test set.

IV. Experimental Evaluation

In this section, we evaluate the effectiveness of CsiGAN on two datasets under both semi-supervised and supervised scenarios. Also, the performance of the different design choices is compared. The data and code are available online[1].

[1]https://github.com/ChunjingXiao/CsiGAN

Algorithm 1 Optimization of CsiGAN via mini-batch SGD method

Input: Labeled data (x_l, y), unlabeled data (x_u), hyper-parameter γ_1, γ_2, γ_3, α, M_{cycle}, learning rate η, and training epochs N_{epochs}.

Initialize: CycleGAN with parameter θ_C, discriminator with parameter θ_D and generator with parameter θ_G.
for num_epoch = 0, ..., N_{epochs} **do**
  if $num\_epoch \% M_{cycle} == 0$ **then**
    Sample a batch of labeled data x_l and unlabeled data x_u.
    Update θ_C for training CycleGAN.
    Save G of CycleGAN as G_c.
  end if
  Generate labeled fake samples (x_c, y'), where $x_c = G_c(x_l)$ and $y' = y + k$ (y is the label of x_l).
  Sample a batch of data (x_c, y'), (x_l, y), x_u and x_v.
  Update θ_D by descending along the stochastic gradient on L_D.
  Sample a batch of unlabeled data x_u.
  update θ_G by descending along the stochastic gradient on L_G.
end for

A. Experiment Setup

We conduct the experiments on two CSI-based behavior recognition datasets. **SignFi data**: Ma *et al.* [8] collect thousands of CSI traces about sign language gestures, which are frequently used in daily life. And each of the users makes the sign gestures with each gesture repeated for 10 times in the lab environment. The data are collected by the Access Point (AP) with 3 external antennas and the Station (STA) with 1 internal antenna. Here the 802.11n CSI tool [7] provides 30 sub-carriers for each antenna pare, and there are 200 packets for each activity (sign gesture). Hence CSI values of each activity have the shape of $200 \times 30 \times 3$. This data can represent a kind of fine-grained action. **FallDeFi data**: Palipana *et al.* [9] gather hundreds of CSI traces about human activities, such as fall, walk, jump, pickup, sit down and stand up. Three volunteers aged between 27 to 30 years with different physiques perform the activities, and around 300 fall actions and 750 other activities are collected. The experiments are conducted in typical indoor environments consisting of two bedrooms, a corridor and the kitchen. For each activity, they collect 10,000 packets during 10 seconds (1000 packets per second), and correspondingly the shape of CSI values for one activity is $10,000 \times 30 \times 3$. To keep the same shape with SignFi data, we remove the first and last 1000 rows of each activity, because we observe that they represent human static status, and then we sample rows every 40 from the left 8000 rows. As a result, the final CSI values have the same shape with SignFi data: $200 \times 30 \times 3$. And our validation shows the classification performance using the data before and after removal almost keeps the same. Therefore, we use the data after removal for the experiments. This can represent a kind of coarse-grained action. The two kinds of data are used for evaluation, which can illustrate the potential for solving different tasks [29], [30].

For all the following experiments, we perform leave-one-subject-out validation. We adopt the average accuracy and F1-score as the metric for evaluation. Here the accuracy refers to the percentage of activities whose class labels are correctly classified. And the F1-score combines Recall and Precision with an equal weight. The models are optimized by Adam with learning rate 0.0003 and $\beta_1 = 0.5$, and the mini-batch size of data is 60. The hyper-parameters γ_1, γ_2, γ_3, α are empirically set 0.6, 0.2, 0.1 and 40 respectively. Training and testing are performed by a Windows desktop with an Intel Xeon E5-1603 CPU and NVIDIA GeForce GTX Titan X GPU.

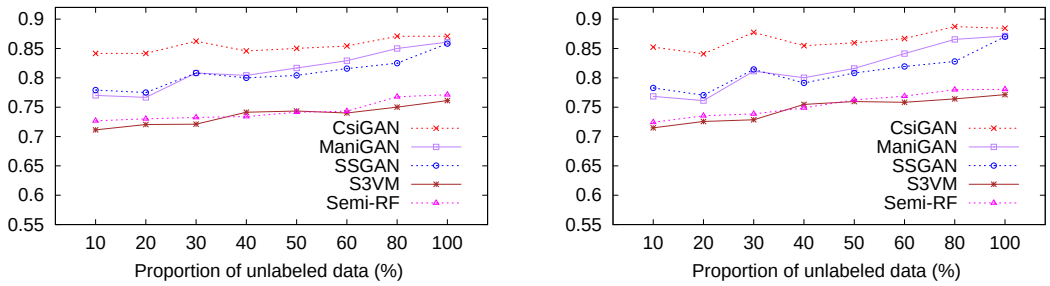
For implementation details about CsiGAN, Table I presents the specific network architectures of the vanilla generator (G) and discriminator (D). In this table, Conv, T-conv, BN, WN, FM and NiN stand for Convolution, Transposed-Convolution, Batch Normalization, Weight Normalization, Feature Maps and Network in Network, respectively. Here, G consists of layers 1-5 and D consists of layers 6-20. In the training process, the softmax function of the last layer in D outputs $2k + 1$ probabilities for k real categories and $k + 1$ fake categories. And in the testing process, it outputs k probabilities for k real categories. The architectures of the complement generator are the same to the one in [24].

TABLE I: Network architectures of the generator (G) and discriminator (D)

	NO.	Operation	Configuration
G	1	Input	1×100 (uniform noise)
	2	Dense	Unit=$5 \times 5 \times 512$ BN ReLU
	3	T-conv	Kernel=5×5 Stride=2×1 FM=256 BN ReLU
	4	T-conv	Kernel=5×5 Stride=2×3 FM=128 BN ReLU
	5	T-conv	Kernel=5×5 Stride=2×2 FM=3 WN Tanh
D	6	Input	$200 \times 30 \times 3$ (CSI values)
	7	Dropout	$p = 0.2$
	8	Conv	Kernel=3×3 FM=96 WN lReLU
	9	Conv	Kernel=3×3 FM=96 WN lReLU
	10	Conv	Kernel=3×3 Stride=5×2 FM=96 WN lReLU
	11	Dropout	$p = 0.5$
	12	Conv	Kernel=3×3 FM=192 WN lReLU
	13	Conv	Kernel=3×3 FM=192 WN lReLU
	14	Conv	Kernel=3×3 Stride=5×2 FM=192 WN IReLU
	15	Dropout	$p = 0.5$
	16	Conv	Kernel=3×3 FM=192 WN IReLU
	17	NiN	FM=192 WN IReLU
	18	NiN	FM=192 WN IReLU
	19	Pooling	FM=192 Global Pooling
	20	Dense	FM=100 WN Softmax

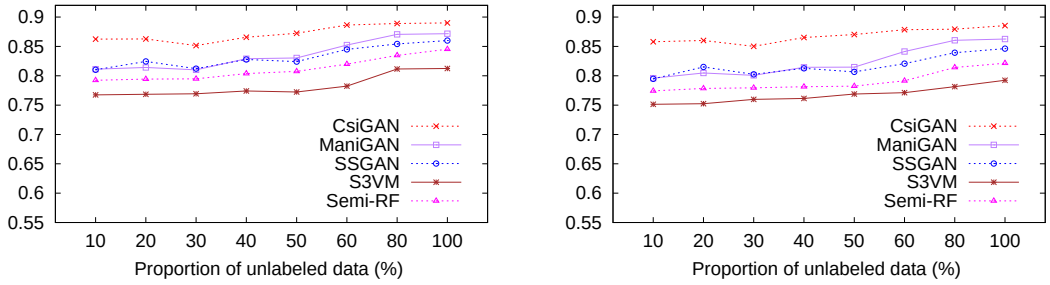
B. Performance of Semi-Supervised learning

We first illustrate the effectiveness of CsiGAN at leveraging the different number of categories of unlabeled data from the left-out user. This equals to semi-supervised learning. And we compare the performance of CsiGAN with four semi-supervised baselines. (1) **ManiGAN** [26]: A state-of-the-art GAN-based semi-supervised learning framework with manifold regularization. This model achieves better performance on CIFAR-10 and SVHN data than a number of GAN-based and non-GAN-based semi-supervised approaches, such as Local GAN [21], Bad GAN [23], Mean teachers [31] and Virtual Adversarial Training [32]. (2) **SSGAN** [19]: A popular GAN-based semi-supervised learning model. The implementation of



(a) Accuracy (b) F1 score

Fig. 2: The semi-supervised performance for SignFi data



(a) Accuracy (b) F1 score

Fig. 3: The semi-supervised performance for FallDeFi data

this model uses the same network architectures with CsiGAN, because the original architectures might not be suitable for dealing with CSI data and obtain worse performance on these two datasets. (3) **S3VM** [33]: a widely used non-deep learning-based semi-supervised method. S3VM cannot take the amplitude of CSI data as input, and thus we extract features using the approach in [9] to feed this model. (4) **Semi-RF** [34]: A semi-supervised learning algorithm that puts a self-training wrapper on the random forest classifier. Semi-RF takes the same input as the S3VM model.

Since terminal users usually want to obtain better performance as soon as possible, unlabeled data for training models can only be collected in a short time after users turn on devices. Therefore, the data generally only have a small number and cover a part of the categories. Hence, we mainly compare the performance when there are a few unlabeled samples from the left-out user. Specifically, we select one user as the left-out user, and the others as the trained ones. For this left-out user, we further evenly divide its data of each category into the unlabeled data set (which will be used for training the model with training set) and testing set. The data of the trained users are regarded as the training set.

The results with the different number of categories of unlabeled samples for the two tasks are shown in Figure 2 and 3. These two figures clearly show that the three GAN-based approaches, CsiGAN, ManiGAN and SSGAN, always outperform the two traditional semi-supervised learning models. This is primarily attributed to their architectures which can efficiently cope with the multimodal CSI data in activity recognition. While, among these three GAN-based frameworks, our proposed CsiGAN further obviously improves the recognition performance, especially for the tiny proportion of unlabeled data. Besides, the differences between accuracy of CsiGAN and the other two GAN-based methods are decreasing steadily as the number of unlabeled samples grows. Although the fluctuation in 30% of unlabeled data arises due to the instability of GANs [35], [36] and inadequate unlabeled data for SignFi data. Still, the overall trend is obvious. When 10% of unlabeled data are used, the accuracy of CsiGAN is 9% and 6% higher than the best baseline (ManiGAN) for SignFi data and FallDeFi data individually. Here, this ratio means that unlabeled data only cover 10% of categories. This suggests that CsiGAN can take advantage of limited unlabeled data to generate diverse fake samples for enhancing discriminator training, and further obtain decent improvement. While, for 100% of unlabeled data, the performance of ManiGAN and SSGAN tends to approach that of CsiGAN. This is because when there are plenty of unlabeled data, the generators of ManiGAN and SSGAN also can generate diverse fake samples which can benefit classifier training. And both methods acquire good performance. Nevertheless, these two methods consistently exhibit an inferior performance compared to CsiGAN for the two data sets. These results strongly suggest that CsiGAN can effectively improve recognition performance compared with the baselines, especially for the scenario with limited unlabeled samples.

C. Performance of Supervised Learning

When the terminal user firstly turns on the device for activity recognition using CSI, there are not any labeled or unlabeled data about the terminal (left-out) user. Hence, here we evaluate the performance of CsiGAN without any data from the left-out user. To meet the requirement of CsiGAN, we use a part of labeled data from the training set as unlabeled data. This experiment only uses labeled samples to train the model, which

equals to fully supervised learning. Hence, besides the semi-supervised baselines mentioned in the previous section, we also compare the effectiveness of CsiGAN with supervised classifiers, including a few neural network-based and conventional methods. (1) **CNN** [8]: A 9-layer CNN for sign gesture classification using CSI data. This network architecture is well tuned for CSI-based sign gesture recognition, and achieves about 97% accuracy for 5-fold cross validation. While, when applying to leave-one-subject-out validation, the average accuracy decreases to around 77%. (2) **LSTM** [37]: A long short term memory (LSTM) extension of RNN for activity recognition. It obtains better accuracy than Random Forest and Hidden Markov Model [37]. (3) **SVM** [9]: A widely used non-neural network approach. After carefully data pre-processing and feature extracting, this model has a 93% accuracy for fall detection, and significantly outperforms other detection methods, such as RTFall [38] and CARM [16]. However, the leave-one-subject-out validation only obtains an average accuracy of 80%. We extract features using the same approach in [9] for this model. (4) **RF**: The random forest supervised classifier. RF is frequently used for activity recognition [13], [39]. And it takes the same input as the SVM model.

The experimental data are the same as the previous section except for the unlabeled data. Here a number of labeled samples randomly selected from the training set are treated as unlabeled data. The results of the two datasets are provided in Figure 4 and 5.

The first observation is that three GAN-based methods attain important gains for both datasets. This is particularly true for SignFi data, where there are a large number of samples and categories. While, CsiGAN outperforms the other two GAN-based mothods, ManiGAN and SSGAN. This indicates that although GAN-based approaches are generally designed for semi-supervised learning tasks, they are still effective for scenarios without unlabeled data, especially for CsiGAN which obtains the best performance. Second, CsiGAN achieves significantly higher accuracy and F1-score than two recent supervised learning approaches, CNN [8] and SVM [9]. For SignFi data, the accuracy of CNN is about 77%, which is almost the same to the result of leave-one-subject-out validation using CNN in [8]. While, the accuracy of CsiGAN reaches more than 84%, which is around 9% higher than that of CNN. Similarly, for FallDeFi data, the accuracy of CsiGAN, 86%, is also significantly higher than that of SVM in [9], 80%. The above results indicate that, compared with supervised and semi-supervised learning baselines, CsiGAN can efficiently improve the recognition performance even though there are not any data from left-out users.

D. Impact of Design Choices

There are a few design choices within the CsiGAN structure. In this section, we evaluate these design choices against baseline models generated by deleting one design component from the CsiGAN model. We study the effect of considering different choices: (1) **noCycle**: This model does not consider labeled fake samples generated by the complement generator G_c, and correspondingly its loss term and new added k outputs are removed. (2) **diffOut**: This model still includes the complement generator and manifold regularization, but the discriminator outputs $k+1$ classes. It means that both vanilla fake samples and labeled fake samples are forced to be put into the $(k+1)$-th class. (3) **noMani**: This model deletes the manifold regularization in the loss function of the discriminator. (4) **noThree**: This model removes all of our proposed components: labeled fake samples generated by G_c, new added k probability outputs and manifold regularization. (5) **full**: The model fully incorporates all the components, which is our proposed CsiGAN.

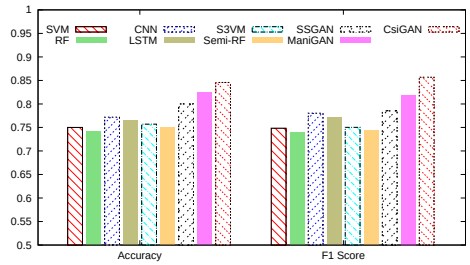


Fig. 4: The supervised performance for SignFi data

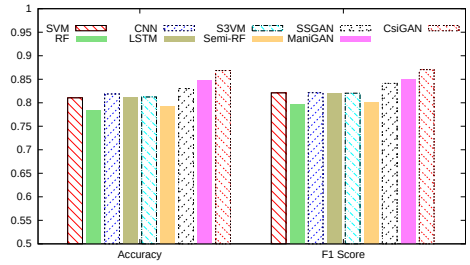


Fig. 5: The supervised performance for FallDeFi data

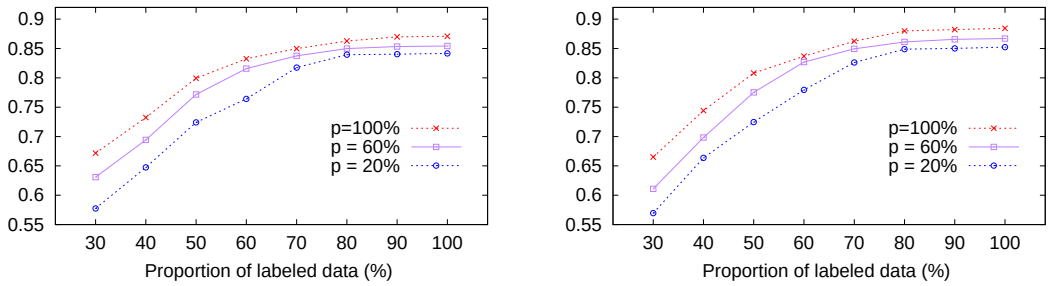
The experimental results using the data in the previous section are presented in Table II with the best results highlighted in boldface. As seen, when removing any component, the models suffer from performance degradation. For deleting labeled fake samples generated by G_c, the accuracy of *noCycle* declines around 2% and 3% for SignFi data and FallDeFi data respectively. And for removing new added k probability outputs (*diffOut*) and manifold regularization (*noMani*), there also exists distinct performance degradation compared with *full*. Besides, without any component we proposed, *noThree* performs worse than other models, declining by 7% compared to *full* for SignFi data. And similar trends also appear for FallDeFi data. These results show that the components we proposed can effectively facilitate the recognition performance.

TABLE II: The performance of different design choices

	DataSet	noCycle	diffOut	noMani	noThree	Full
SignFi	Accuracy	82.56	83.75	81.67	77.50	**84.17**
	F1 Score	82.57	83.14	82.65	77.05	**84.09**
FallDeFi	Accuracy	83.24	84.73	83.47	82.44	**86.27**
	F1 Score	83.47	85.13	82.98	81.50	**86.01**

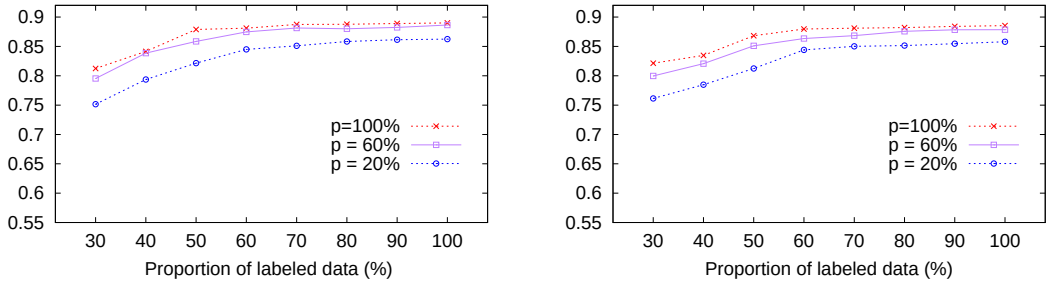
E. Impact of Labeled Data Size

We have shown that CsiGAN can utilize unlabeled data efficiently. It can consistently achieve the best performance



(a) Accuracy (b) F1 score

Fig. 6: The performance with different size of labeled data for SignFi data



(a) Accuracy (b) F1 score

Fig. 7: The performance with different size of labeled data for FallDeFi data

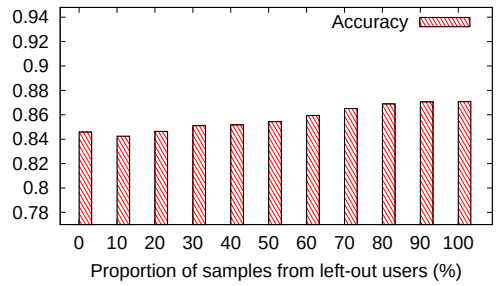
on the two datasets with different proportions of unlabeled data. However, it is not clear that whether CsiGAN can utilize labeled data efficiently. Next, we conduct a series of experiments on performing semi-supervised learning with different proportions of labeled data to explore the effect of labeled data size on the CsiGAN performance. For these experiments, we select p = [20, 60, 100]% of all the unlabeled samples as unlabeled data, and select q% of all the labeled samples as labeled data. When q = 100%, the experiments are fully semi-supervised learning, which are the same as those in Section IV-B.

As shown in Figure 6 and Figure 7, for all the p values, CsiGAN achieves increasing accuracy and F1 score with the rise of the labeled data size for both datasets, which indicates that the labeled data size has an important impact on recognition performance for CsiGAN. However, the growth rates of the two datasets are quite different. For SignFi data, the difference between accuracy of 80% and 30% of labeled data is more than 19% when p = 100%. However, the difference is only 8% for FallDeFi data. At the same time, for SignFi data, when the data size reaches 80% of labeled data, the performance keeps stable. While, for FallDeFi data, 60% of labeled data already acquires similar accuracy and F1 score with 100% of data. The reason behind this is because the quantity of labeled samples per category per user has a great difference for the two datasets. In fact, there are 10 and 21 labeled samples per category for SignFi data and FallDeFi data individually. Hence, the same ratio means the various number of labeled data for these two datasets, which further leads to different performance. These results suggest that CsiGAN needs a certain amount of training data to achieve better performance. However, the number of labeled data, such as 20 per category, is easily affordable by human labeling.

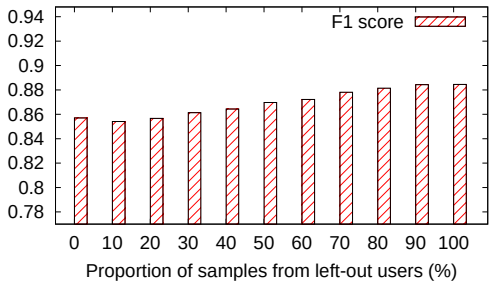
F. Impact of Unlabeled Data Distribution

In CsiGAN, the complement generator tries to transfer labeled data into fake samples with the style of unlabeled data. Before this transfer, the complement generator needs to be trained based on labeled data and unlabeled data. Hence the difference between distributions of labeled data and unlabeled data can influence quality of fake samples produced by the complement generator. Therefore, here we evaluate the impact of this difference on recognition performance. For these experiments, the unlabeled data are composed of h% of samples from left-out users and (1 - h)% of samples from trained users. When h = 0%, all the unlabeled data are from trained users and have the same distribution with labeled data, and experiments equal to supervised learning. When h = 100%, unlabeled data are from left-out users and have different distribution with labeled data, and that become semi-supervised learning.

The results for different h values are shown in Figure 8 for SignFi data. The figures show that both accuracy and F1 score steadily increase with the rise of the h value. When h = 0%, there are not any unlabeled data from left-out users during the training process. Hence, labeled fake samples produced by the complement generator cannot capture any characteristic of the left-out user, and correspondingly cannot cover the entire distribution of test data. Therefore, it obtain the worst performance among all the h values. While, with the rise of the h value, the distribution of the generated samples gradually approaches the distribution of test data from the left-out user, and accordingly the recognition performance keeps increasing. FalldeFi data, not shown here, also exhibits the same trends. Hence, the distribution of unlabeled data can influence recognition accuracy of CsiGAN, and to obtain better performance unlabeled data should keep the same distribution with test data from left-out users.



(a) Accuracy



(b) F1 score

Fig. 8: The performance with different distribution of unlabeled data

G. Visual Analysis of CSI Data

Here, we conduct visual inspection about real samples, labeled fake samples and vanilla fake samples. Since all the data exhibit similar trends, we take a few samples from SignFi data as an example to conduct analysis. Figure 9 depicts the CSI amplitudes of one subcarrier as a function of time. The samples in this figure, including real data and fake data, are extracted from experiments of the semi-supervised scenario with 100% unlabeled data. And all these samples belong to the same activity.

In these figures, the four ones on the left represent real samples. Here, Figure 9(a) and 9(d) refer to two real samples performed by user A at different time, and similarly, Figure 9(b) and 9(e) are for user B. These four figures show that, for the same activity, fluctuations of CSI amplitudes for two samples performed by the same user are quite similar. For example, waveforms of the two samples of user A, shown in Figure 9(a) and 9(d), keep swinging together with the close rhythm. It is still true for user B, shown in Figure 9(b) and 9(e). On the other hand, fluctuations for two samples performed by the different users, such as the sample 1 of user A in Figure 9(a) and the sample 1 of user B in Figure 9(b), exhibit many differences. Hence, these results indicate that different individuals can cause distinct fluctuations of CSI traces even when performing the same activity. This may be an important reason why recognition performance significantly decreases for leave-one-subject-out validation.

Besides real samples, two figures on the right illustrate fake samples. The fake sample in Figure 9(c) is produced by the complement generator, which tries to transfer the source data (labeled data from trained users) in Figure 9(a) into the new one but with the style of the targeted data (unlabeled data from left-out users) in Figure 9(b). And the same relationship can apply to Figure 9(f), 9(d) and 9(e). As observed in the figures, the waveform in Figure 9(c) has the characteristic of both the waveforms in Figure 9(a) and 9(b). For example, for the first 25 packets, the waveform in Figure 9(c) is quite similar to Figure 9(a). On the other hand, the two troughs in Figure 9(c) near the 50th and 150th packet can well match the ones in Figure 9(b). And similar trends can be found for Figure 9(f). Therefore, samples produced by the complement generator can be regarded as the superposition of both source and targeted samples. And to some extent they have the characteristics of targeted data. These results indicate that our introduced complement generator is effective to produce fake samples with the style of left-out users.

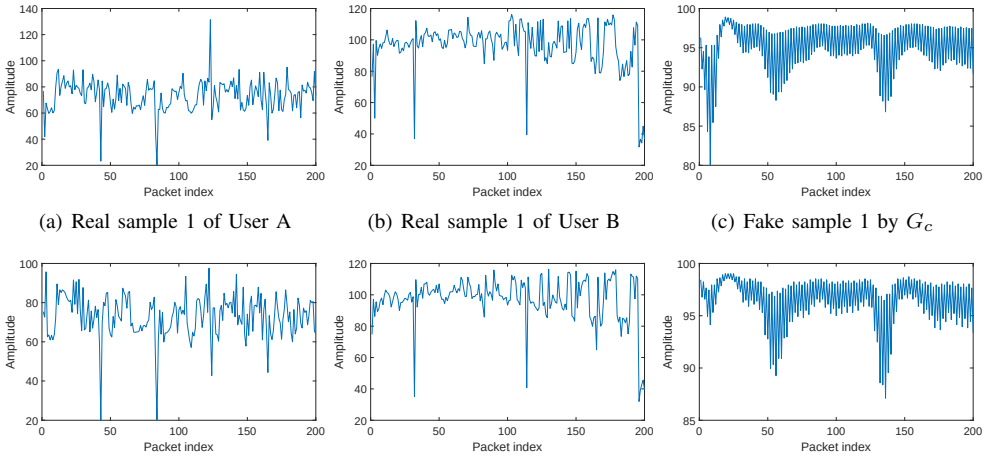
To compare the difference between functions of the complement generator and vanilla generator, we also illustrate CSI amplitudes of two fake samples produced by the vanilla generator. As shown in Figure 10, their amplitudes are relatively distorted, compared with samples of the complement generator in Figure 9(c) and 9(f). These results indicate that, for CSI-based activity recognition, although the crude samples produced by the vanilla generator can experimentally enhance recognition performance, there is still large room for improvement. And our introduced complement generator can be an effective way to produce refined fake samples for better performance.

V. Related Work

This work is mainly related to two research areas: CSI-based activity recognition and GAN-based semi-supervised learning. Next, we will present an overview of the most closely related works in each area, and highlight the major differences between our study and these works.

CSI-based activity recognition. The studies on CSI-based activity recognition can be divided into two main genres according to recognition methods: template matching-based and classification model-based approaches. For the former, specific activities are identified by matching their characteristics with the pre-constructed activity profiles. For instance, Wang *et al.* [16] built a human activity recognition system, CARM, which estimates the correlation between CSI dynamics and human activities, and recognizes a given activity by matching it to the best-fit profile based on this correlation. Wang *et al.* [40] presented E-eyes to identify human activities by calculating Dynamic Time Warping (DTW) distances between CSI traces and comparing that with the given threshold. Ali *et al.* [41] proposed a keystroke recognition system, which classifies keystrokes by computing DTW distance between different typing gestures. Virmani *et al.* [42] explored the connection of the CSI feature and gesture position and orientation, and translated CSI measures to related virtual samples for recognizing gestures. Xiao *et al.* [43] exploited features containing both time and frequency information, and adopted DTW to calculate the distance between two feature vectors for exercise activity recognition.

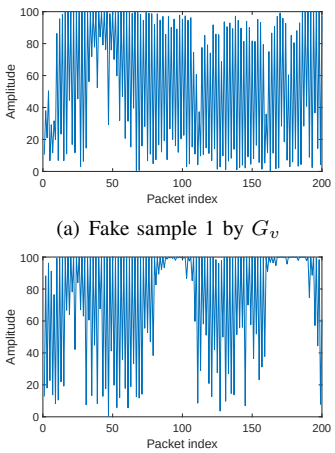
For classification model-based approaches, some researchers focus on extracting effective statistical features from time and frequency domains, and then employ standard classification models, such as SVM and Random Forest, to classify



(a) Real sample 1 of User A (b) Real sample 1 of User B (c) Fake sample 1 by G_c

(d) Real sample 2 of User A (e) Real sample 2 of User B (f) Fake sample 2 by G_c

Fig. 9: CSI amplitudes for real samples and fake samples generated by the complement generator (G_c) as a function of time



(a) Fake sample 1 by G_v

(b) Fake sample 2 by G_v

Fig. 10: Samples produced by the vanilla generator (G_v)

activities. For example, Palipana *et al.* [9] designed a fall detection system, FallDeFi, and they used the Short-Time Fourier Transform to extract time-frequency features and a sequential forward selection algorithm to single out features, and achieved higher accuracy using SVM. Yang *et al.* [44] proposed a CSI-based activity detection framework integrating the WiFi device and cloud server. And they extracted features of activities via Class Estimated Basis Space Singular Value Decomposition (CSVD) and conducted the recognition by Nonnegative Matrix Factorization (NMF). Han *et al.* [45] exploited seven features from the time domain, such as median absolute deviation, velocity of signal change and signal entropy, and employed SVM to detect the fall activity. Wu *et al.* [46] presented a human activity recognition system, TW-See, to meet scenarios of the WiFi signals through the wall. They proposed an opposite robust PCA approach to obtain the correlation between activities and CSI values, and then extracted the eight features and used a BP neural network to recognize human activities.

Other studies adopt deep learning approaches to classify human activities. Since the raw amplitude and phase of CSI data can be fed to these models, it is not required to extract features and reduce dimension. Hence, these studies mainly build different deep learning networks for various activity recognition. For examples, Ma *et al.* [8] proposed a sign language recognition system, SignFi. In this system, after removing the noise and recovering the CSI change, CSI data are fed to a 9-layer convolutional neural network model, which is more suitable to the scenario with a large number of sign gestures. To deal with spatial diversity, Wang *et al.* [47] exploited characteristics of many spatial dimensions from multiple antennas pairs, and built a deep learning framework combining CNN and LSTM to conduct classification. Gao *et al.* [48] transformed CSI data from multiple channels into radio images, and developed a deep learning-based image processing framework for activity recognition. Feng *et al.* [49] proposed a deep learning framework based on LSTM model for activity detection, which achieves higher accuracy and efficiency than the two non-deep learning baselines. To alleviated the influence of environmental dynamics, Zou *et al.* [50] proposed an adversarial domain adaptation scheme to reduce the domain discrepancy between the original environment and new environment. This scheme is combined with the convolutional neural network trained in the original environment to detect gestures. For the scenario that people usually deploy multiple access points (APs), Li *et al.* [51] transformed the CSI data of multiple APs into heatmaps, which are fed into their specially designed deep network to conduct activity recognition.

The above researches mainly focus on the cross-validation and achieve expected performance for different activity recognition. However, these models fail to consider individual differences, and their recognition performance may severely decline when applying to leave-one-subject-out validation, such as SignFi [8], FallDeFi [9] and CARM [16]. Here, we focus on the problem of performance degradation for leave-one-subject-out validation, which has not been fully considered. And we adopt GANs to address this task, which have been rarely used for CSI-based applications.

GAN-based semi-supervised learning. Adopting GANs for semi-supervised learning is first applied to the field of image classification. For example, Kumar *et al.* [17] aimed at enforcing invariance for the classifier of semi-supervised learning tasks by estimating tangent space to the data manifold via GANs. Salimans *et al.* [19] used the GAN discriminator as the classifier, and modified that to output $k+1$ probability values representing k true categories and one fake category. The model is suggested to perform well by adopting feature matching to produce fake data. Qi *et al.* [21] presented a localized GAN for semi-supervised learning tasks. In this model, local generators are introduced to parameterize various local manifold geometry. Dai *et al.* [23] found that the distribution of generated samples cannot match the real data distribution perfectly, and a preferred generator should produce complement samples in the feature space for GAN-based semi-supervised learning. And they proposed several regularization terms for the loss function of the generator and discriminator to deal with this problem. Lecouat *et al.* [26] presented a manifold regularization term for the semi-supervised GAN. The term tries to approximate the variant of the Laplacian norm and is easily calculated based on the GAN. And Li *et*

al. [52] introduced an additional classifier to cope with the problem arose from the two-player design of general GANs under the scenario of semi-supervised learning.

Most recently, besides image classification, a few researchers extended GAN-based semi-supervised learning for other applications, such as gene expression inference [53], graphs [54], Internet of Things [29]. Specifically, since traditional models of gene expression inference generally formulate it in a completely supervised manner, Dizaji *et al.* [53] proposed a novel semi-supervised framework to make use of unlabeled data. This framework consists of a GAN network and an inference network, and both networks have collaborative relation, such that prediction of the former enhances the training of generators, and the generated data are utilized to improve the learning of the latter. Ding *et al.* [54] presented a new competitive game between the generator and classifier for semi-supervised learning on graphs. Under this equilibrium, the generator produces fake data in low-density areas between subgraphs, and the classifier fully considers the density property of subgraphs. Yao *et al.* [29] proposed a semi-supervised deep-learning framework, SenseGAN, for Internet of Things applications. This framework separates the functionalities of discriminator and classifier into two neural networks, and designs specific generator and discriminator structures for handling multimodal sensing inputs, and stabilizes the adversarial training process by introducing Wasserstein metric.

These GAN-based semi-supervised learning approaches achieve a remarkable improvement for different applications. While they aim to deal with the situation that there are plenty of unlabeled samples from all the users. However, we try to address the application when there are limited unlabeled data from the left-out user. And we propose three novel components to deal with this problem.

VI. Conclusions and future work

In this paper, we investigated the performance degradation problem of leave-one-subject-out validation for CSI-based activity recognition, and proposed a robust GAN-based activity recognition framework, CsiGAN. In this model, we introduced a new generator to produce complement fake samples, which can effectively boot performance of the discriminator/classifier. And we changed the output and objective function of the discriminator to place the classification decision boundary for each category in low-density areas. In addition, we also proposed a manifold regularization to enhance classification performance. Based on fine-grained and course-grained human activity data, experimental results demonstrate that our proposed CsiGAN outperforms state-of-the-art approaches under both semi-supervised and supervised scenarios.

There are several limitations about our proposed model, which can become fruitful directions of further investigation. First, currently CycleGAN is not very stable when a test sample looks unusual compared to training samples. Hence, we are interested in adopting improved CycleGAN for CsiGAN to enhance recognition performance. Second, CsiGAN is only evaluated for training and testing under the same experimental environments. Thus, more evaluation is needed to understand the feasibility in the context of different deployed scenarios. Third, the learning process of CsiGAN is computationally intensive. Therefore, more studies are needed to enable online adaptive learning with streaming data on lightweight devices.

References

[1] K. Yatani and K. N. Truong, "Bodyscope: A wearable acoustic sensor for activity recognition," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 341–350.

[2] M. Z. Uddin and M. M. Hassan, "Activity recognition for cognitive assistance using body sensors data and deep convolutional neural network," *IEEE Sensors Journal*, pp. 1–8, 2019.

[3] J. Wannenburg and R. Malekian, "Physical activity recognition from smartphone accelerometer data for user context awareness sensing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 12, pp. 3142–3149, 2017.

[4] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," *Future Generation Computer Systems*, vol. 81, no. 4, pp. 307–313, 2018.

[5] R. Bodor, B. Jackson, N. Papanikolopoulos, and H. Tracking, "Vision-based human tracking and activity recognition," in *Proceedings of the 11th Mediterranean Conference on Control and Automation*, 2003, pp. 18–20.

[6] A. Jalal, Y.-H. Kim, Y.-J. Kim, S. Kamal, and D. Kim, "Robust human activity recognition from depth video using spatiotemporal multi-fused features," *Pattern Recognition*, vol. 61, no. C, pp. 295–308, 2017.

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

[8] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "Signfi: Sign language recognition using wifi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, pp. 1–21, 2018.

[9] S. Palipana, D. Rojas, P. Agrawal, and D. Pesch, "Falldefi: Ubiquitous fall detection using commodity wi-fi devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–25, 2018.

[10] H. Abdelnasser, K. A. Harras, and M. Youssef, "A ubiquitous wifi-based fine-grained gesture recognition system," *IEEE Transactions on Mobile Computing*, vol. 4, no. 11, pp. 1–14, 2018.

[11] 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.

[12] M. Shahzad and S. Zhang, "Augmenting user identification with wifi based gesture recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–27, 2018.

[13] H. Zou, Y. Zhou, R. Arghandeh, and C. J. Spanos, "Multiple kernel semi-representation learning with its application to device-free human activity recognition," *IEEE Internet of Things Journal*, pp. 1–11, 2019.

[14] C. Feng, S. Arshad, S. Zhou, D. Cao, and Y. Liu, "Wi-multi: A three-phase system for multiple human activity recognition with commercial wifi devices," *IEEE Internet of Things Journal*, pp. 1–12, 2019.

[15] Z. Wang, B. Guo, Z. Yu, and X. Zhou, "Wi-fi csi-based behavior recognition: From signals and actions to activities," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 109–115, 2018.

[16] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of wifi signal based human activity recognition," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015, pp. 65–76.

[17] A. Kumar, P. Sattigeri, and P. T. Fletcher, "Semi-supervised learning with gans: Manifold invariance with improved inference," in *The 31st Annual Conference on Neural Information Processing Systems*, 2017, pp. 5535–5545.

[18] Z. Erickson, S. Chernova, and C. C. Kemp, "Semi-supervised haptic material recognition for robots using generative adversarial networks," in *Proceedings of the Annual Conference on Robot Learning*, 2017, pp. 157–166.

[19] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 2234–2242.

[20] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3581–3589.

[21] G.-J. Qi, L. Zhang, H. Hu, M. Edraki, J. Wang, and X.-S. Hua, "Global versus localized generative adversarial nets," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1517–1525.
[22] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," in *Advances in Neural Information Processing Systems*, 2018, pp. 3235–3246.
[23] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. Salakhutdinov, "Good semi-supervised learning that requires a bad gan," in *The 31st Annual Conference on Neural Information Processing Systems*, 2017, pp. 6511–6521.
[24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2242–2251.
[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.
[26] B. Lecouat, C. S. Foo, H. Zenati, and V. R. Chandrasekhar, "Semi-supervised learning with gans: Revisiting manifold regularization," in *Proceedings of the International Conference on Learning Representations*, 2018.
[27] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal The Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.
[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, 2015.
[29] S. Yao, Y. Zhao, H. Shao, C. Zhang, A. Zhang, S. Hu, D. Liu, S. Liu, L. Su, and T. Abdelzaher, "Sensegan: Enabling deep learning for internet of things with a semi-supervised framework," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–21, 2018.
[30] X. Chen and Y. Wang, "Predicting resting-state functional connectivity with efficient structural connectivity," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 6, pp. 1079–1088, 2018.
[31] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 1195–1204.
[32] T. Miyato, S. Maeda, S. Ishii, and M. Koyama, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2018.
[33] S. Ding, Z. Zhu, and X. Zhang, "An overview on semi-supervised support vector machine," *Neural Computing and Applications*, vol. 28, no. 5, pp. 969–978, 2017.
[34] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study," *Knowledge and Information Systems*, vol. 42, no. 2, pp. 245–284, 2015.
[35] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.
[36] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F. Wang, "Generative adversarial networks: introduction and outlook," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588–598, 2017.
[37] S. Yousefi, H. Narui, S. Dayal, S. Ermon, and S. Valaee, "A survey on behavior recognition using wifi channel state information," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 98–104, 2017.
[38] H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li, "Rt-fall: A real-time and contactless fall detection system with commodity wifi devices," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 511–526, 2017.
[39] Z. Qin, Y. Zhang, S. Meng, Z. Qin, and K.-K. R. Choo, "Imaging and fusing time series for wearable sensor-based human activity recognition," *Information Fusion*, vol. 53, pp. 80–87, 2020.
[40] 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.
[41] 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.
[42] 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.
[43] F. Xiao, J. Chen, X. H. Xie, L. Gui, J. L. Sun, and W. Ruchuan, "Seare: A system for exercise activity recognition and quality evaluation based on green sensing," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–10, 2018.
[44] J. Yang, H. Zou, H. Jiang, and L. Xie, "Device-free occupant activity sensing using wifi-enabled iot devices for smart homes," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3991–4002, 2018.
[45] C. Han, K. Wu, Y. Wang, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," in *IEEE Conference on Computer Communications*, 2014, pp. 271–279.
[46] X. Wu, Z. Chu, P. Yang, C. Xiang, X. Zheng, and W. Huang, "Tw-see: Human activity recognition through the wall with commodity wi-fi devices," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 306–319, 2019.
[47] F. Wang, W. Gong, and J. Liu, "On spatial diversity in wifi-based human activity recognition: A deep learning based approach," *IEEE Internet of Things Journal*, pp. 1–1, 2018.
[48] Q. Gao, J. Wang, X. Ma, X. Feng, and H. Wang, "Csi-based device-free wireless localization and activity recognition using radio image features," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 346–10 356, 2017.
[49] C. Feng, S. Arshad, R. Yu, and Y. Liu, "Evaluation and improvement of activity detection systems with recurrent neural network," in *2018 IEEE International Conference on Communications*, 2018, pp. 1–6.
[50] H. Zou, J. Yang, Y. Zhou, L. Xie, and C. J. Spanos, "Robust wifi-enabled device-free gesture recognition via unsupervised adversarial domain adaptation," in *The 27th International Conference on Computer Communication and Networks*, 2018, pp. 1–8.
[51] H. Li, K. Ota, M. Dong, and M. Guo, "Learning human activities through wi-fi channel state information with multiple access points," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 124–129, 2018.
[52] C. Li, K. Xu, J. Zhu, and B. Zhang, "Triple generative adversarial nets," in *The 31st Annual Conference on Neural Information Processing Systems*, 2017, pp. 4089–4099.
[53] K. G. Dizaji, X. Wang, and H. Huang, "Semi-supervised generative adversarial network for gene expression inference," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 1435–1444.
[54] M. Ding, J. Tang, and J. Zhang, "Semi-supervised learning on graphs with generative adversarial nets," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 913–922.

Chunjing Xiao is currently an associate professor at School of computer and information engineering in Henan University. He received his Ph.D. degree from University of Electronic Science and Technology of China. He was a visiting scholar in the Department of Electrical Engineering and Computer Science at Northwestern University. His research interests include online social networks, information retrieval, machine learning, wireless networks and Internet of Things.

Daojun Han is currently an associate professor at School of computer and information engineering in Henan University. He received his Ph.D. degree from Sun Yat-sen University. He is also a major member of the Institute of Data and Knowledge Engineering, Henan, China. His research interests include information retrieval, machine learning, access control technology and network security.

Yongsen Ma received the B.S. degree in control science and engineering from Shandong University, Jinan, China, and the M.S. degree in control science and engineering from Shanghai Jiao Tong University, Shanghai, China and is currently pursuing the Ph.D. degree at the Department of Computer Science, College of William and Mary, Williamsburg, VA, USA. He is currently a member of the LENS Research Group, where he is advised by Dr. G. Zhou. He was a Research Assistant with Intel, Shanghai, China. His current research interests include wireless networking, ubiquitous sensing, and mobile systems.

Zhiguang Qin is the full professor of the School of Information and Software Engineering in University of Electronic Science and Technology of China (UESTC), where he is also Director of the Key Laboratory of New Computer Application Technology and Director of UESTC-IBM Technology Center. His research interests include medical image processing, computer networking, information security, cryptography, information management, intelligent traffic, electronic commerce, distribution, and middleware.