

Article

Personalized Federated Learning Based on Bidirectional Knowledge Distillation for WiFi Gesture Recognition

Huan Geng ^{1,2} , Dongshang Deng ^{1,2} , Weidong Zhang ^{1,2} , Ping Ji ^{1,2,*} and Xuangou Wu ^{1,2} 

¹ School of Computer Science and Technology, Anhui University of Technology, Ma'anshan 243032, China; hgeng@ahut.edu.cn (H.G.); dsdeng@ahut.edu.cn (D.D.); wdzhang@ahut.edu.cn (W.Z.); wuxgou@ahut.edu.cn (X.W.)

² Anhui Engineering Research Center for Intelligent Applications and Security of Industrial Internet, Ma'anshan 243032, China

* Correspondence: jiping@ahut.edu.cn

Abstract: WiFi-based human gesture recognition has a wide range of applications in smart homes. Existing methods train gesture classification models by collecting large amounts of WiFi signal data in a centralized manner. However, centralized training faces challenges, including high communication overhead and the risk of data privacy leakage. Federated learning (FL) provides an opportunity to collaboratively train and share models without compromising data privacy. One of the main challenges FL faces is data that is non-Independent and Identically Distributed (non-IID) across clients. Specifically, in the gesture recognition scenario, since the transmission of WiFi signals is susceptible to cross-environment and cross-person interference, non-IID mainly manifests itself as a cross-domain problem. Cross-domain makes the knowledge learned between client models incompatible. Therefore, in the cross-domain scenario, effectively extracting and combining the knowledge learned by the client is a challenge. To solve this problem, we propose pFedBKD, a novel personalized federated learning scheme via bidirectional distillation. First, the knowledge that is beneficial to the client is extracted from the shared server model through knowledge distillation in the client, which helps train the personalized model of the client. Second, the server adaptively adjusts the aggregation weights according to the deviation between the shared model and the client's local model so that the server's shared model can extract more public knowledge. We conduct experiments on multiple open-source datasets. Experimental results show that our method is superior to existing methods and effectively alleviates the problem of reduced model recognition accuracy caused by cross-domain challenges.

Keywords: gesture recognition; personalized federated learning; knowledge distillation; cross-domain



Citation: Geng, H.; Deng, D.; Zhang, W.; Ji, P.; Wu, X. Personalized Federated Learning Based on Bidirectional Knowledge Distillation for WiFi Gesture Recognition.

Electronics **2023**, *12*, 5016. <https://doi.org/10.3390/electronics12245016>

Academic Editor: Christos J. Bouras

Received: 3 November 2023

Revised: 6 December 2023

Accepted: 12 December 2023

Published: 15 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the continuous development of the intelligent Internet of Things (IoT), WiFi-based gesture recognition is widely being applied in diverse domains such as smart homes [1,2], person recognition [3], and virtual reality [4,5]. Current WiFi-based gesture recognition methods require centralized user data collection to train gesture recognition models [6–10]. In addition, higher data volume requirements are required to achieve better performance of the model. However, collecting data is difficult, data transmission will consume a large amount of communication resources, and sharing data will cause concerns about user privacy [11–15].

Federated learning (FL) [16] emerged as a solution that only requires the client to upload model parameters instead of raw data. The classic FL system mainly consists of a server trusted by all parties and multiple clients. In each round of communication, the aggregated global model parameters are sent by the server to the clients participating in this round of updates. After receiving the parameters, the client performs training locally based on its private data. Then, the client participating in the update sends the

updated local model parameters to the server side and performs the aggregation operation on the server side. During the entire process, the server can only access the gradients or model parameters uploaded by participants, thus protecting the client's data privacy to a great extent.

However, one of the main challenges in FL is non-Independent and Identically Distributed (non-IID) [17–20] across client data. In WiFi-based gesture recognition scenarios, non-IID is mainly presented cross-domain: cross-environment [21,22] and cross-person [6,23]. Due to the multipath effect of Channel State Information (CSI) [24,25], cross-environment and cross-person lead to changes in the phase and amplitude of CSI signals, making the collected data features heterogeneous. Cross-domain not only makes the knowledge learned between client models incompatible, but also makes global and local knowledge incompatible. Ultimately, the accuracy of client-side testing of the global model after aggregation will be lower than before aggregation.

In response to the above cross-domain problems, existing research can be mainly divided into two different categories: traditional federated learning, which generates a unified model and personalized federated learning (PFL), which generates its personalized model for each client. Traditional federated learning, exemplified by approaches like FedProx [26] and SCAFFOLD [27], aims to generate a powerful global model that converges to the global optimum and is shared with all clients. On the other hand, PFL endeavours to train a personalized model tailored to perform effectively on each client, as demonstrated by methods such as FedFomo [28], Ditto [29] and pFedSD [30]. However, the methods mentioned above either address the incompatibility issue between local and global knowledge or exclusively concern the incompatibility problem among local knowledge without considering the two jointly.

To address the cross-domain problem, we propose a personalized federated learning approach based on bidirectional knowledge distillation for WiFi gesture recognition (pFedBKD). First, during the local training process of the client, we use knowledge distillation (KD) to extract the knowledge of the global model locally to absorb more beneficial local knowledge. The effects of incompatibility between global and local knowledge are mitigated. Second, we quantify the extent to which the local model deviates from the global distribution by uploading the difference between the labels produced locally by the global and local models. We assign greater weights to local models that exhibit lower deviations from the global distribution. This strategy helps alleviate the effect of incompatibility between local knowledge.

The main contributions of this paper can be summarized as follows:

- We propose pFedBKD to solve the incompatibility problem between knowledge caused by cross-domain challenges by using knowledge distillation on the client side and adjusting the weight of the aggregation model on the server side.
- In order to alleviate the knowledge incompatibility between the global model and the local model due to cross-domain challenges, we integrate global and local knowledge through the KD algorithm during the local training process to generate a personalized model. We then determine the aggregation weights in the aggregation process by evaluating the local and global bias degree using Jensen-Shannon (JS) divergence.
- We conduct extensive experiments on open-source datasets (Widar3.0 [6], and UT-HAR [31] datasets) in multiple cross-domain scenarios. Experimental results show that our proposed method is better than existing methods and improves model recognition accuracy.

The subsequent sections of this work are structured in the following manner: Section 2 provides an overview of the existing research in the field. Section 3 provides an explanation for the motivation. Section 4 provides a clear and concise explanation of the problem formulation for PFL. The system design is outlined in Section 5. Section 6 displays the empirical findings. Ultimately, Section 7 serves as the final conclusion of this paper.

2. Related Work

Recently, FL has thrived as users pay more attention to the privacy and security of their data. However, gesture recognition scenarios often involve multiple users and devices, and the data between different clients is non-IID. Applying the federated averaging algorithm to build a global model cannot guarantee performance. Work on federated learning and WiFi-based action recognition technology is the main emphasis of this section. These kinds of work on federated learning are specifically divided into three types: FL in human activity recognition, federated knowledge distillation and model aggregation in FL.

2.1. Federated Learning in Human Activity Recognition

FL's benefits of protecting privacy and cutting communication costs have led to its widespread use in human activity recognition [32–35]. Sozinov et al. [33] apply FedAvg to human activity recognition for the first time. WiFederated [36] adopts the FL framework to scale the deployment of CSI-based WiFi sensing systems from multiple locations. The introduction of FL in human activity recognition will bring new challenges. Traditional data-centric methods cannot be used directly in FL scenarios because FL transmits model parameters rather than raw data. To address the above challenges, ClusterFL [37] reduces the impact of data heterogeneity through collaborative training between similar nodes. Nevertheless, these approaches ignore the contradiction between global and local knowledge by using the global model in place of the local model while updating the client.

2.2. Federated Learning and Knowledge Distillation

Multiple studies have investigated the integration of FL with KD to tackle the issue of data heterogeneity. By saving model parameters after the last round of local updates, pFedSD [30] extracts previous personalized knowledge into the local model through knowledge distillation in the new round of updates to accelerate recall. FedFTG [38] uses the local model as a discriminator and generates hard samples that can represent the local distribution through adversarial training. Then, KD is used to extract local knowledge to the aggregated model to improve the precision of the global model. pFedDF [39] presents a unified distillation technique for combining models, suitable for heterogeneous scenarios. Our approach addresses cross-domain challenges by reducing the incompatibility between local and global knowledge while mitigating the incompatibility among local knowledge.

2.3. Model Aggregation in Federated Learning

Model aggregation is crucial in combining knowledge from different client devices while minimizing communication costs and improving learning performance in FL. Chen et al. [40] propose a synchronous model learning strategy at the client level and a time-weighted aggregation method at the server side to reduce communication costs and improve FL accuracy. In order to promote similar client collaboration in the model aggregation process, FedAMP [41] introduces a novel federated attention messaging method. Rather than addressing cross-domain issues, the primary goal of these model aggregation solutions is to increase FL's communication efficiency.

3. Analysis and Motivations

In recent years, wireless devices have become an essential part of the IoT fields, such as smart homes, which can use wireless signals to identify user actions. By employing a sophisticated model architecture, the efficacy of the ultimate trained model will be enhanced as the number of users ready to provide private action data increases [42]. However, in real-life scenarios, most users are unwilling to share private data, or the labour and bandwidth costs required to collect data are too high. FL, in this instance, does not necessitate the client to upload sensitive information. Instead, it collectively trains the global model by transmitting parameters to the server. User privacy is ensured, and communication bandwidth is conserved.

Due to the differences in environments or users among different clients, traditional federated averaging (FedAvg) algorithms have mediocre performance [8]. In our trials, we assess the precision of both the local and global models to examine the correlation between these two models, as depicted in Figure 1.

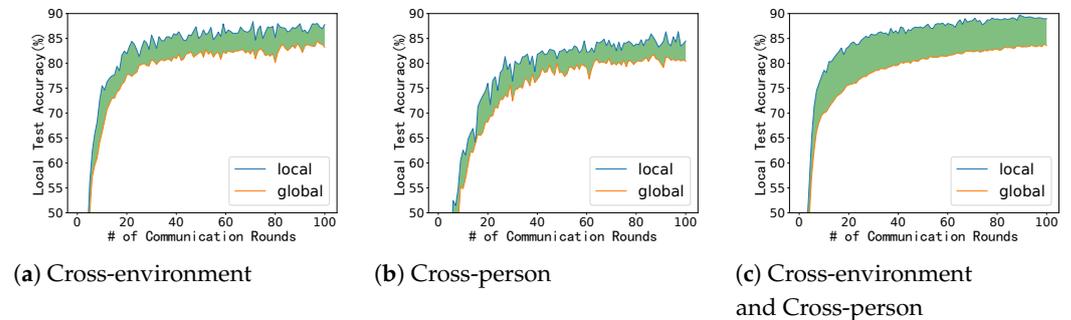


Figure 1. When FedAvg is applied to the Widar3.0 [6] dataset, you can observe the change in client-side test accuracy of the model before and after aggregation (green shaded area). Local models show higher accuracy when cross-environment, cross-person, and both exist.

Our experiments indicate that the precision of model aggregation falls when clients have challenges that span cross-domain. This phenomenon is caused by the incompatibility between global and local knowledge due to cross-domain issues [30]. This occurrence suggests that when the aggregation process takes place, historical local knowledge is abandoned, leading to a decline in the precision of the local model. Therefore, in a new communication round, the client must retrain the local model accuracy to the level of the previous round. The retraining process increases training time and may reduce the final model accuracy. When cross-domain problems become more severe, the phenomenon of accuracy decline becomes more prominent. From this, we judge that the precision of the global aggregation model decreases due to the incompatibility between local knowledge in the aggregation stage. At the same time, due to the incompatibility of global and local knowledge, the client's use of the global model for initialization in a new round of updates will lead to a decrease in accuracy.

Subsequently, we proceed with the training procedures of the classic FedAvg algorithm to conduct a more comprehensive examination of the factors contributing to the decrease in performance. For each communication cycle, a subset of clients denoted as rK , will be chosen to take part in the update, where r is a value between 0 and 1. The chosen client performs E local epochs. The participating client in the current iteration receives the updated model transmitted by the server and utilizes it as the initial reference for a subsequent training iteration. Server-sent models are aggregated from selected clients with different data distributions (such as different environments or users).

Hence, the most recent version of the model downloaded to the client exhibits superior accuracy in global distribution compared to its local distribution [26]. Figure 1 depicts the disparity in accuracy between the global model used for testing and the local model specific to the client following the most recent set of changes. The global model's performance has notably decreased following the latest round of aggregation compared to the accuracy of the model submitted by the customer in the prior round. The performance degradation is caused by the client using the downloaded global model as the starting point for a new round of updates. In real-world environments where cross-domain problems are more severe, this decrease in accuracy will become more apparent.

Assume that in t communication round, the update selects a client that has not been selected in previous rounds of updates. Then, the client will need more local updates to reach the previous performance level. However, the classic federated learning algorithm does not consider this, and the locally updated epochs are consistent. In order to obtain a generalized global model, the classic FedAvg algorithm discards some of the client's interests, especially in scenarios with severe cross-domain problems. Determining how to

balance generalization and personalization is a problem that PFL needs to solve. Using the global model in FedAvg to initialize the local model can enhance generalization capabilities, but it is challenging to adapt to different scenarios between clients. At the same time, due to the differences between local data distributions, if the aggregation stage can reduce the impact of different distribution differences, the global model’s accuracy after aggregation can be improved.

4. Preliminaries

4.1. Gesture Recognition Technology

In an orthogonal frequency division multiplexing (OFDM) system [43], a wireless sensing device collects a set of S subcarriers from each data packet, which consists of complex values. CSI is described as

$$Z(f_i, t) = |Z(f_i, t)|e^{j\angle Z(f_i, t)}, i \in [1, S] \tag{1}$$

where $|Z(f_i, t)|$ symbolizes the CSI’s amplitude of subcarrier f_i at the centre frequency and t timestamp, while $\angle Z(f_i, t)$ represents the phase of CSI.

Doppler Frequency Shift (DFS) [44] occurs due to the relative motion between the transmitter and receiver of a wireless device. According to the Carrier Amplitude and Phase Recovery Method (CARM), the fundamental reason for DFS is the variation in signal propagation paths. It is possible to represent the shift caused by the reflected signals as

$$f_D(t) = -\frac{1}{\bar{\lambda}} \frac{d}{dt}d(t) = -f \frac{d}{dt}\tau(t) \tag{2}$$

where $\bar{\lambda}$ represents the signal’s wavelength, f represents the subcarrier frequency, and $\tau(t)$ represents the time it takes for the signal to travel. Also, $d(t)$ represents the distance of the path where there is no direct line of sight.

DFS occurs when volunteers perform gestures that cause various body parts to move at different speeds, especially when the movements are relatively large. Assuming that the vector of velocity \vec{v} at each timestamp causes cumulative DFS, note that for k transmitter-receiver links, the DFS $F_D^k(\vec{v})$ can be calculated using the given formula.

$$F_D^k(\vec{v}) = c_x^k \vec{v}_x + c_y^k \vec{v}_y \tag{3}$$

The location of the transmitter-receiver link determines c_x^k and c_y^k . \vec{v}_x represents the user’s facial direction, \vec{v}_y represents the vertical direction [45]. c_x^k and c_y^k can be used to find the optimal solution of \vec{v}_x . \vec{v}_y can be separated from the measured DFS using CSI [6]. \vec{v}_x and \vec{v}_y can be used to represent the body velocity distribution (BVP). The same operations can be carried out by several users in various forms.

4.2. Personalized Federated Learning

Below, we first review the traditional FL objective function and then introduce the objective function of PFL. Our empirical observations of performance degradation of global models in PFL and our desire to reduce incompatibilities between local knowledge motivate us to provide pFedBKD.

We generate personalized models for different clients adapted to the client’s local data distribution as the ultimate goal. The dataset in this article consists of supervised WiFi-based gesture recognition data. Each client has its data set D_k , with a total of K clients (wireless devices). A piece of data consists of x input and y tags. All client data sets form $D = \bigcup_{k=1}^K D_k$. We first review the classic FL system, whose goal is to reduce the total experience loss:

$$\min_w F(w) = \sum_{k=1}^K \frac{|D_k|}{|D|} F_k(w) \tag{4}$$

The optimization function of client k is expressed as $F_k(w)$, where w represents the global model parameters.

Since the distribution of D_k is very different between different clients, a single global model w is challenging to guarantee to fit the local target $F_k(w)$ of each client. PFL generates a personalized model v_k for client k that is different from other clients and consistent with its local target.

$$\min_{\{v_1, v_2, \dots, v_K\}} \sum_{k=1}^K \frac{|D_k|}{|D|} F_k(v_k) \tag{5}$$

5. Our Solution and Algorithm Description

5.1. Overview

In this section, we divide pFedBKD into two parts to introduce the design of the pFedBKD framework in detail. The details of our method are summarized in Algorithms 1 and 2. Algorithm 1 describes the calculation process of the client, and Algorithm 2 describes the calculation process of the server. Figure 2 shows that the pFedBKD process is based on a client-server framework, where the wireless device is considered a client.

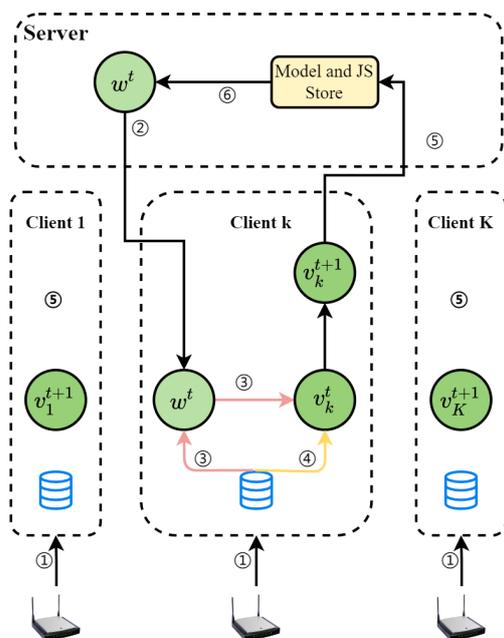


Figure 2. Description of pFedBKD workflow. Each communication round performs data training on the client side and model aggregation on the server side until the termination condition is met. The workflow consists of six steps: ① this is the collection and preprocessing part of CSI data; ② when communication starts, the server sends the initial model w_0 ; in subsequent updates, sends the aggregated updated global model w^t ; ③ trains the local model v_k^t on the client data set, extract the knowledge of w^t to the local through KD; ④ calculates v_k^{t+1} and w^t on the client data set JS value of w^t ; ⑤ the client uploads the updated parameter w^{t+1} and JS value to the server; ⑥ the collected JS values are normalized to generate aggregate weights, and then the aggregation model generates a global model w^{t+1} .

Algorithm 1 pFedBKD Algorithm: Client**Input:** local training batch size B , learning rate η , local epochs E **Output:** client personalization models $\{v_k\}_{k \in [K]}$

```

1: procedure CLIENTUPLOAD( $k, w^t$ )
2:   Download the model from the server
3:   for  $i = 1, \dots, E$  do
4:      $\Phi_k(v_k^t) = F_k(v_k^t) + \lambda \mathcal{L}_{KL}(q(v_k^t) || q(w^t))$ 
5:      $v_k^t = v_k^t - \eta \nabla \Phi_k(v_k^t, w_t)$ 
6:   end for
7:   Store models  $v_k^t$  and  $w^t$ 
8:   Calculate JS divergence  $L_k^{t+1}$  ▷ Equation (8)
9:   return local model  $v_k^t$  and  $L_k^{t+1}$ 
10: end procedure

```

Algorithm 2 pFedBKD Algorithm: Server**Input:** communication rounds T , training participation ratio r **Output:** client personalization models $\{v_k\}_{k \in [K]}$

```

1: procedure SERVERAGGREGSYION()
2:   Initialize model  $w^0$ 
3:   for  $t = 1, \dots, T$  do
4:     Extract  $rK$  clients to form  $C^t$ 
5:      $C^t$  clients downloads global model parameters  $w^t$ 
6:     for each client  $k \in C^t$  do
7:        $v_k^{t+1}, L_k^{t+1} \leftarrow ClientUpload(k, w^t)$ 
8:     end for
9:     Compute aggregate weights  $\beta_t^k = \frac{1/L_k^{t+1}}{\sum_{k=1}^K 1/L_k^{t+1}}$ 
10:     $w^{t+1} = \sum_{k=1}^K \beta_t^k v_k^{t+1}$ 
11:  end for
12: end procedure

```

5.2. Client Model Training and Extract Global Knowledge to Local

When training clients, strike a balance between local and global knowledge. We refrain from replacing the client model with a global model to leverage global knowledge. Instead, we use KD to bring global knowledge to clients. To regulate the amount of mixing between the global and local knowledge, we employ a hyperparameter λ .

Therefore, the client's loss function is

$$\Phi_k(v_k^t) = F_k(v_k^t) + \lambda \mathcal{L}_{KL}(q(v_k^t) || q(w^t)) \quad (6)$$

Here, the degree of local participation in global knowledge is regulated by the hyperparameter λ . The function $F_k(\cdot)$ represents the loss of client k . \mathcal{L}_{KL} represents the KL divergence function between the global model prediction $q(w^t)$ and the current client local prediction $q(v_k^t)$.

In pFedBKD, client k updates client model parameters using function $\Phi_k(v_k^t)$ (instead of $F_k(v_k^t)$). The client model weights v_k are updated via stochastic gradient descent (SGD) as follows:

$$v_k^t = v_k^t - \eta \nabla \Phi_k(v_k^t, w_t) \quad (7)$$

where η represents the learning rate.

We will also calculate the Jensen-Shannon (JS) divergence between the updated client-side local model on its private training set and the predictions of the global model parameters.

$$\mathcal{L}_{JS}(q(v_k^{t+1})||q(w^t)) = \frac{1}{2}\mathcal{L}_{KL}(q(v_k^{t+1})||\frac{q(v_k^{t+1}) + q(w^t)}{2}) + \frac{1}{2}\mathcal{L}_{KL}(q(w^t)||\frac{q(v_k^{t+1}) + q(w^t)}{2}) \quad (8)$$

JS divergence is symmetric compared to KL divergence. Symmetry assumes that there are two probability distributions, P and Q; choosing P as the reference distribution and Q as the reference distribution will have the same result. The calculated JS divergence values and updated client model parameters are subsequently transferred to the server.

5.3. Calculate Aggregated Weights Based on Global and Local Biases

In the traditional FedAvg algorithm, the contribution of each client is measured by the amount of data (in general, the number of client data sets is equal in the experimental environment). Due to the data heterogeneity among the various clients, the data distribution of the client is inconsistent with the global data distribution. When the model weight is aggregated on the server side, a new way to measure the aggregated weight is needed. Due to the limitations of other methods, we use the JS divergence value mentioned in the previous section. The greater the JS value of a particular client, the greater the deviation between its data and global distribution. To mitigate the influence of the local deviation from the global distribution during aggregation, it should give a smaller weight, and the aggregation weight β_i^k is calculated as follows:

$$\beta_i^k = \frac{1/\mathcal{L}_{JS}(q(v_k^{t+1}), q(w^t))}{\sum_{k=1}^K 1/\mathcal{L}_{JS}(q(v_k^{t+1}), q(w^t))} \quad (9)$$

To create the next iteration of the global model w^{t+1} , each client submits the revised model parameters, which are then combined on the server side based on the aggregation weight:

$$w^{t+1} = \sum_{k=1}^K \beta_i^k v_k^{t+1} \quad (10)$$

6. Experiment Results

The following experiments showcase the efficacy of our approach and investigate the influence of alterations in hyperparameters $\{\lambda, \tau\}$ on performance.

6.1. Dataset

Our experiments used two open-source data sets, Widar3.0 [6] and UT-HAR [31]. We verified that our method performs well in cross-domain situations in Widar3.0. In order to improve the robustness and comprehensiveness of our method, we expanded the scope of the study. We used UT-HAR to verify the effectiveness of our method further.

Widar3.0 [6] contains WiFi data from 17 users and three environments. In Widar3.0, not all users have taken corresponding actions in each environment. Therefore, to better verify various cross-domain challenges, we use six actions, namely push and pull, scan, tap, slide, draw O and zigzag. Widar3.0 makes detailed annotations for each data user and environment. Therefore, we can easily set up various cross-domain (including cross-person and cross-environment) experiments to verify our solution.

UT-HAR [31] is a CSI dataset including six users and seven actions. We use all its data in our experiments. In UT-HAR, we use the Dirichlet distribution [46] to control the degree of heterogeneity, recorded as $Dir(\rho)$. The smaller the ρ , the more heterogeneous the setting.

6.2. Settings

In this article, whenever Widar3.0 appears in the comparison experiment, the experimental data set uses Widar3.0. Otherwise, the data set used is UTHAR. In pFedBKD, we first set up a complex scenario to simulate the natural environment. That is, there is heterogeneity between different client environments, and users also have heterogeneity.

Moreover, we test the performance of our method in cross-environment and cross-person situations, respectively.

We keep the parameter settings consistent with standard work. Unless otherwise stated, each client runs for five epochs per round by default and communicates for 100 rounds. The ratio of training set to test set in the client is 7:3. We use the Adam optimizer in Widar3.0, setting the weight decay to 0, the momentum to 0.9, and the learning rate to 0.001. In the UT-HAR experiment, we used the SGD optimizer with a weight decay 1×10^{-6} , and a learning rate of 0.01.

In Widar3.0 for each task, the training batch size is $B = 128$, while in UT-HAR, the training batch size is $B = 64$. μ_1 is selected for FedProx from {0.001, 0.01, 0.1, 1}. Ditto controls the hyperparameter μ_2 of local update, and we choose among {0.01, 0.1, 0.5, 1, 2}. We choose λ from {0.1, 0.3, 0.5} for our technique, and we modify the temperature τ from {1, 10}.

Our method is implemented with Python 3.8. The deep learning library of torch v.1.11.0 is used for coding and implementation, and TensorFlow v.2.3 is used as the backend for the model. The hardware platform we used was a desktop with a 3.40 GHz AMD Ryzen 7 5800 eight-core processor CPU, 48 GB RAM, and an NVIDIA GeForce RTX 3070 GPU.

6.3. Model Structure

This article uses two model structures corresponding to the data sets Widar3.0 and UT-HAR, respectively. Regarding Widar3.0: first, there are two 3×3 convolution blocks for extracting spatial features. Then, there is a 2×2 max pooling layer for feature dimensionality reduction, followed by two fully connected layers. We utilize a gated recurrent unit (GRU) to extract dynamic features from BVP since it is time series data. Compared with long short-term memory (LSTM) [47], GRU can achieve comparable results using fewer parameters. To prevent overfitting, we introduce a dropout layer. To obtain the prediction result, we employ a fully connected layer as a classifier. Lastly, input the softmax classifier. Regarding UT-HAR: here, we use the classic Multilayer Perceptron (MLP), which consists of three fully connected layers. Use the fully connected layer to extract data features, pass through the hidden layer, and finally output the classification prediction result through the output layer.

6.4. Benchmark

We compare pFedBKD with four baselines. We briefly introduce these baselines as follows.

- Widar3.0 [6]: the technique amalgamates the data from all users involved in the training process.
- FedAvg [16]: upload all client model parameters and average to generate a global model.
- FedProx [26]: using l_2 regularization during local updates, the client's update direction is prevented from deviating too much from the global distribution.
- pFedSD [30]: by saving model parameters after the last round of local updates, it extracts previous personalized knowledge into the local model through knowledge distillation in the new round of updates to accelerate recall.
- Diito [29]: by regularizing the global model parameters, a personalized model is generated for the client to improve the fairness and robustness of FL.

6.5. Experimental Analysis

In cases when there are differences between environments and users, each client stores a user's data, and there are cross-environment and cross-person exchanges between clients. There are 21 clients in total. The performance of our method is still better than other solutions when both cross-domain factors exist, as shown in Figure 3. Our method is more effective in this scenario, and the accuracy rate can reach 0.9, which is far better than other methods. The performance of FedAvg is not as good as that of Widar3.0. FedAvg exhibits

suboptimal performance in the presence of significant variation in data between clients. Moreover, the performance of FedProx is not good. Simply adding near-end items on the client side cannot handle this cross-domain problem well. Although pFedSD and Ditto have achieved some results, it still has a particular gap with our method because it ignores the incompatibility between local knowledge.

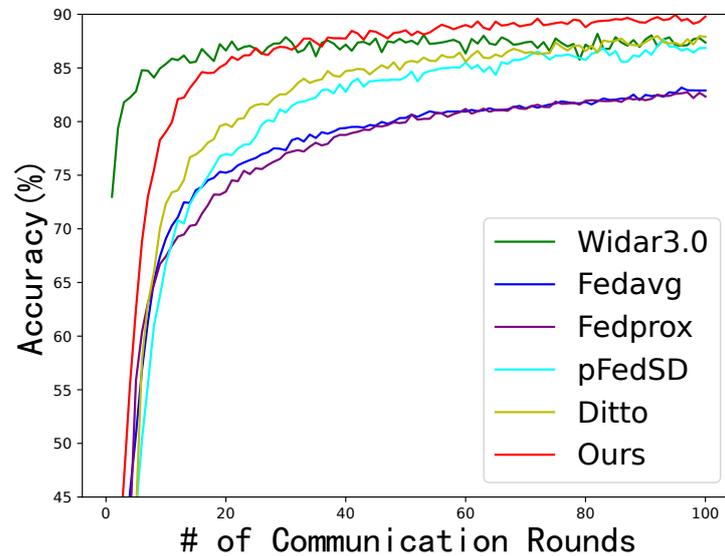


Figure 3. Comparison between pFedBKD and other methods in the case of differences in users and environments.

When there are differences in the environment: Table 1 displays a comparison of different approaches for the same users in different rooms. Furthermore, it is apparent that the precision of the majority of techniques increases with the growing number of individuals involved. As the overall user participation in training diminishes, specifically when the total number of datasets falls, the performance of all approaches also declines. In the case of significant client heterogeneity, the performance performance between FedProx and FedAvg is lower because they ignore the needs of client personalization. Due to the cross-environment advantage of the BVP dataset, most methods are inferior to the training results of Widar3.0, but our method still has advantages. In Widar3.0, there are only three of the same users in room 1 and room 2, namely user 1, user 2, and user 3. To mitigate the impact of user variations, we just utilize these three individuals for experimental purposes. Since the training does not involve many user data, the accuracy of all methods is low in this case.

Table 1. On the data set Widar3.0, the accuracy of different methods is compared when there are cross-environment differences. The client belongs to room 1 or room 2.

User Num	Widar3.0	FedAvg	FedProx	pFedSD	Ditto	Ours
Users = 1	0.805	0.788	0.806	0.832	0.823	0.833
Users = 2	0.844	0.804	0.831	0.851	0.838	0.854
Users = 3	0.851	0.823	0.839	0.855	0.852	0.862

When there are differences in users: the accuracy of pFedBKD in Table 2 can exceed 0.85 in room 1 and reach 0.89 when the number of users reaches 12. Overall, pFedBKD has better performance than other methods. The benefit of pFedBKD diminishes as the level of data heterogeneity diminishes. Conversely, as the level of variation in data rises, the advantages of pFedBKD over alternative methods become increasingly apparent. We can see that the performance of FedAvg and FedProx is average in the experiment. When the number of users participating in training is four and eight, the performance of pFedSD is second only

to ours and better than that of other methods. When the number of participants in training reaches 12, Ditto's performance exceeds pFedSD. However, there is still a gap between it and our method. Our method considers the impact of mitigating the incompatibility of global knowledge and local knowledge on the client side and the impact between local knowledge on the server, which makes our method perform better.

Table 2. On the data set Widar3.0, the accuracy of each method is compared when there are cross-person differences. All clients' environments belong to room 1.

User Num	Widar3.0	FedAvg	FedProx	pFedSD	Ditto	Ours
Users = 4	0.819	0.812	0.784	0.845	0.841	0.854
Users = 8	0.853	0.816	0.836	0.87	0.867	0.874
Users = 12	0.856	0.799	0.80	0.852	0.869	0.882

Table 3 shows that the performance of all methods exceeds 0.78. As ρ increases, that is, when data heterogeneity between clients decreases, the performance of all methods improves. According to Table 3, we can observe that the performance improvement of FedProx is limited because it only cares about global performance. The performance of pFedSD and Ditto is similar. FedProx only cares about the global model's performance and ignores the local model's personalised needs. pFedSD and Ditto only consider the incompatibility between the global and local models on the client side. In short, our method still maintains the lead under various ρ values.

Table 3. Accuracy comparison under different ρ values in UT-HAR.

$Dir(\rho)$	FedAvg	FedProx	pFedSD	Ditto	Ours
$Dir(0.01)$	0.783	0.797	0.822	0.819	0.845
$Dir(0.05)$	0.866	0.869	0.893	0.886	0.928
$Dir(0.1)$	0.916	0.889	0.936	0.939	0.954

In Figure 4, we demonstrate the variation of performance under different hyperparameters (coefficient λ and temperature τ). We tested various values of τ from the set {1, 5, 10} and λ from the set {0.1, 0.3, 0.5, 0.7}. Generally, $\tau = 1$ yielded better results. Lower temperature values indicate that negative labels were given less attention during distillation.

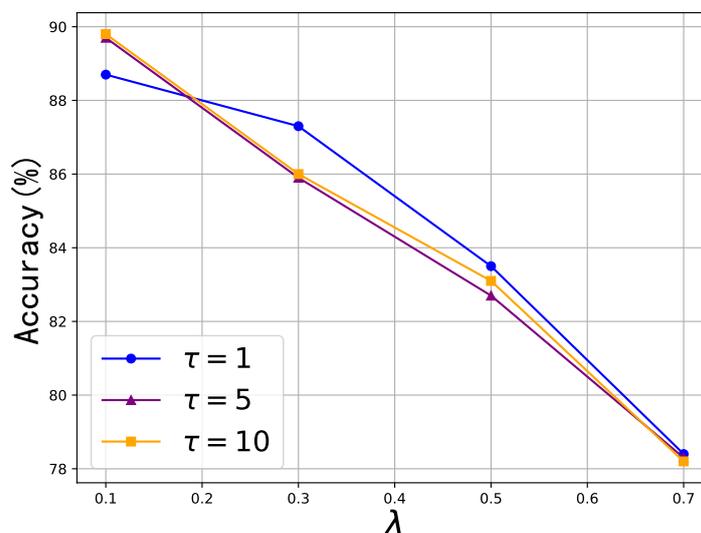


Figure 4. Effects of λ and τ on dataset Widar3.0.

The λ parameter determines the impact of global knowledge on local training, and we maintained a constant value for it throughout our studies. The graph in Figure 4 demonstrates that an increase in the value of λ resulted in a decline in performance. Hence, selecting a modest value for λ is crucial to attain the highest level of accuracy. We also contemplated dynamically modifying the parameter λ by assigning distinct values to clients, contingent upon the global and local model's performance in local test sets.

7. Conclusions

We proposed pFedBKD, an action recognition method based on PFL. To solve the adverse effects of the previous PFL method, pFedBKD designed a method that did not directly use the global model for initialization but indirectly utilized the global knowledge by using KD, avoiding the problem of knowledge forgetting. In the aggregation, the server assigned weights according to the degree of local deviation from the global values, significantly mitigating the influence of data heterogeneity. The experimental results unequivocally demonstrated that pFedBKD effectively mitigated the problem of model accuracy degradation stemming from the heterogeneity of data features among clients.

Author Contributions: Conceptualization, H.G., D.D. and P.J.; methodology, H.G.; software, H.G.; validation, H.G. and W.Z.; formal analysis, H.G.; investigation, H.G.; resources, W.Z.; data curation, H.G.; writing—original draft preparation, H.G.; writing—review and editing, H.G. and D.D.; visualization, H.G.; supervision, X.W.; project administration, X.W.; funding acquisition, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Nature Science Foundation of China, grant number 62172003; the University Synergy Innovation Program of Anhui Province, grant number GXXT-2022-051; the Natural Science Foundation of Anhui Provincial Education Department, grant number YJS20210350.

Data Availability Statement: The Widar3.0 data set used in this study comes from Tsinghua University. The link is <http://tns.thss.tsinghua.edu.cn/widar3.0/>, and the access date is 2 September 2023; The link of the UT-HAR data set is <https://github.com/xyanchen/WiFi-CSI-Sensing-Benchmark>, and the access date is 2 September 2023.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zou, H.; Zhou, Y.; Yang, J.; Jiang, H.; Xie, L.; Spanos, C.J. WiFi-enabled device-free gesture recognition for smart home automation. In Proceedings of the 2018 IEEE 14th International Conference on Control and Automation (ICCA), Anchorage, AK, USA, 12–15 June 2018; pp. 476–481.
2. Zhang, B.B.; Zhang, D.; Li, Y.; Hu, Y.; Chen, Y. Unsupervised Domain Adaptation for RF-based Gesture Recognition. *IEEE Internet Things J.* **2023**, *10*, 21026–21038. [CrossRef]
3. Shahzad, M.; Zhang, S. Augmenting user identification with WiFi based gesture recognition. *Proc. Acm Interactive Mobile Wearable Ubiquitous Technol.* **2018**, *2*, 1–27. [CrossRef]
4. Fu, Z.; Xu, J.; Zhu, Z.; Liu, A.X.; Sun, X. Writing in the air with WiFi signals for virtual reality devices. *IEEE Trans. Mob. Comput.* **2018**, *18*, 473–484. [CrossRef]
5. Polo, A.; Capra, F.; Lusa, S.; Rocca, P.; Salas-Sánchez, A.Á.; Salucci, M. Machine Learning-based Inversion of Wireless Signals for Real-Time Gesture Recognition. In Proceedings of the 2023 12th International Conference on Modern Circuits and Systems Technologies (MOCASST), Athens, Greece, 28–30 June 2023; pp. 1–3.
6. Zheng, Y.; Zhang, Y.; Qian, K.; Zhang, G.; Liu, Y.; Wu, C.; Yang, Z. Zero-Effort Cross-Domain Gesture Recognition with Wi-Fi. In Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, New York, NY, USA, 17–21 June 2019; pp. 313–325.
7. Xiao, C.; Lei, Y.; Ma, Y.; Zhou, F.; Qin, Z. DeepSeg: Deep-learning-based activity segmentation framework for activity recognition using WiFi. *IEEE Internet Things J.* **2021**, *8*, 5669–5681. [CrossRef]
8. Li, X.; Song, F.; Luo, M.; Li, K.; Chang, L.; Chen, X.; Wang, Z. Caring: Towards Collaborative and Cross-domain Wi-Fi Sensing. *IEEE Trans. Mob. Comput.* **2023**, *2023*, 1–14. [CrossRef]
9. Liu, S.; Chen, Z.; Wu, M.; Wang, H.; Xing, B.; Chen, L. Generalizing Wireless Cross-Multiple-Factor Gesture Recognition to Unseen Domains. *IEEE Trans. Mob. Comput.* **2023**, *2023*, 1–14. [CrossRef]
10. Zhang, C.; Jiao, W. ImgFi: A High Accuracy and Lightweight Human Activity Recognition Framework Using CSI Image. *IEEE Sens. J.* **2023**, *23*, 21966–21977. [CrossRef]

11. Truex, S.; Baracaldo, N.; Anwar, A.; Steinke, T.; Ludwig, H.; Zhang, R.; Zhou, Y. A Hybrid Approach to Privacy-Preserving Federated Learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, New York, NY, USA, 15 November 2019; pp. 1–11.
12. Mothukuri, V.; Parizi, R.M.; Pouriyeh, S.; Huang, Y.; Dehghantanha, A.; Srivastava, G. A survey on security and privacy of federated learning. *Future Gener. Comput. Syst.* **2021**, *115*, 619–640. [[CrossRef](#)]
13. Zhang, T.; Xu, C.; Shen, J.; Kuang, X.; Grieco, L.A. How to Disturb Network Reconnaissance: A Moving Target Defense Approach Based on Deep Reinforcement Learning. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 5735–5748. [[CrossRef](#)]
14. Zhang, T.; Xu, C.; Zou, P.; Tian, H.; Kuang, X.; Yang, S.; Zhong, L.; Niyato, D. How to mitigate DDOS intelligently in SD-IOV: A moving target defense approach. *IEEE Trans. Ind. Inform.* **2023**, *19*, 1097–1106. [[CrossRef](#)]
15. Niknam, S.; Dhillon, H.S.; Reed, J.H. Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Commun. Mag.* **2020**, *58*, 46–51. [[CrossRef](#)]
16. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.y. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the Machine Learning Research, Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Seattle, WA, USA, 20–22 April 2017; Volume 54, pp. 1273–1282.
17. Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; Chandra, V. Federated learning with non-iid data. *arXiv* **2018**, arXiv:1806.00582.
18. Sattler, F.; Wiedemann, S.; Müller, K.R.; Samek, W. Robust and communication-efficient federated learning from non-iid data. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3400–3413. [[CrossRef](#)] [[PubMed](#)]
19. Wang, H.; Kaplan, Z.; Niu, D.; Li, B. Optimizing federated learning on non-iid data with reinforcement learning. In Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications, Virtual Conference, 6–9 July 2020; pp. 1698–1707.
20. Zhu, H.; Xu, J.; Liu, S.; Jin, Y. Federated learning on non-IID data: A survey. *Neurocomputing* **2021**, *465*, 371–390. [[CrossRef](#)]
21. Zhang, Y.; Chen, Y.; Wang, Y.; Liu, Q.; Cheng, A. CSI-based human activity recognition with graph few-shot learning. *IEEE Internet Things J.* **2022**, *9*, 4139–4151. [[CrossRef](#)]
22. Zhang, X.; Tang, C.; Yin, K.; Ni, Q. WiFi-based cross-domain gesture recognition via modified prototypical networks. *IEEE Internet Things J.* **2021**, *9*, 8584–8596. [[CrossRef](#)]
23. Xiao, R.; Liu, J.; Han, J.; Ren, K. OneFi: One-Shot Recognition for Unseen Gesture via COTS WiFi. In Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems, New York, NY, USA, 15–17 November 2021; pp. 206–219.
24. Wu, K.; Xiao, J.; Yi, Y.; Chen, D.; Luo, X.; Ni, L.M. CSI-based indoor localization. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *24*, 1300–1309. [[CrossRef](#)]
25. Yang, Z.; Zhou, Z.; Liu, Y. From RSSI to CSI: Indoor localization via channel response. *ACM Comput. Surv. CSUR* **2013**, *46*, 1–32. [[CrossRef](#)]
26. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated Optimization in Heterogeneous Networks. In *Proceedings of Machine Learning and Systems*; Dhillon, I., Papailiopoulos, D., Sze, V., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 2, pp. 429–450.
27. Karimireddy, S.P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; Suresh, A.T. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In Proceedings of the 37th International Conference on Machine Learning, Online, 13–18 July 2020; Volume 119, pp. 5132–5143.
28. Zhang, M.; Sapra, K.; Fidler, S.; Yeung, S.; Alvarez, J.M. Personalized Federated Learning with First Order Model Optimization. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021; pp. 1–17.
29. Li, T.; Hu, S.; Beirami, A.; Smith, V. Ditto: Fair and robust federated learning through personalization. In Proceedings of the International Conference on Machine Learning, Virtual Event, 18–24 July 2021; pp. 6357–6368.
30. Jin, H.; Bai, D.; Yao, D.; Dai, Y.; Gu, L.; Yu, C.; Sun, L. Personalized edge intelligence via federated self-knowledge distillation. *IEEE Trans. Parallel Distrib. Syst.* **2023**, *34*, 567–580. [[CrossRef](#)]
31. Yang, J.; Chen, X.; Zou, H.; Lu, C.X.; Wang, D.; Sun, S.; Xie, L. SenseFi: A library and benchmark on deep-learning-empowered WiFi human sensing. *Patterns* **2023**, *4*, 1–22. [[CrossRef](#)]
32. Chen, Y.; Qin, X.; Wang, J.; Yu, C.; Gao, W. FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare. *IEEE Intell. Syst.* **2020**, *35*, 83–93. [[CrossRef](#)]
33. Sozinov, K.; Vlassov, V.; Girdzijauskas, S. Human activity recognition using federated learning. In Proceedings of the 2018 IEEE International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), Melbourne, Australia, 11–13 December 2018; pp. 1103–1111.
34. Bettini, C.; Civitarese, G.; Presotto, R. Personalized semi-supervised federated learning for human activity recognition. *arXiv* **2021**, arXiv:2104.08094.
35. Xiao, Z.; Xu, X.; Xing, H.; Song, F.; Wang, X.; Zhao, B. A federated learning system with enhanced feature extraction for human activity recognition. *Knowl. Based Syst.* **2021**, *229*, 107338. [[CrossRef](#)]
36. Hernandez, S.M.; Bulut, E. WiFederated: Scalable WiFi Sensing Using Edge-Based Federated Learning. *IEEE Internet Things J.* **2022**, *9*, 12628–12640. [[CrossRef](#)]
37. Ouyang, X.; Xie, Z.; Zhou, J.; Huang, J.; Xing, G. 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, New York, NY, USA, 24 June–2 July 2021; pp. 54–66.

38. Zhang, L.; Shen, L.; Ding, L.; Tao, D.; Duan, L.Y. Fine-Tuning Global Model via Data-Free Knowledge Distillation for Non-IID Federated Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 10174–10183.
39. Zhang, J.; Guo, S.; Ma, X.; Wang, H.; Xu, W.; Wu, F. Parameterized Knowledge Transfer for Personalized Federated Learning. In *Proceedings of the Advances in Neural Information Processing Systems*; Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2021; Volume 34, pp. 10092–10104.
40. Chen, Y.; Sun, X.; Jin, Y. Communication-Efficient Federated Deep Learning with Layerwise Asynchronous Model Update and Temporally Weighted Aggregation. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 4229–4238. [[CrossRef](#)] [[PubMed](#)]
41. Huang, Y.; Chu, L.; Zhou, Z.; Wang, L.; Liu, J.; Pei, J.; Zhang, Y. Personalized Cross-Silo Federated Learning on Non-IID Data. *Proc. Aaai Conf. Artif. Intell.* **2021**, *35*, 7865–7873. [[CrossRef](#)]
42. Jiang, W.; Miao, C.; Ma, F.; Yao, S.; Wang, Y.; Yuan, Y.; Xue, H.; Song, C.; Ma, X.; Koutsonikolas, D.; et al. Towards Environment Independent Device Free Human Activity Recognition. In Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, New Delhi, India, 29 October–2 November 2018; pp. 289–304.
43. Armada, A.G. Understanding the effects of phase noise in orthogonal frequency division multiplexing (OFDM). *IEEE Trans. Broadcast.* **2001**, *47*, 153–159. [[CrossRef](#)]
44. Li, X.; Li, S.; Zhang, D.; Xiong, J.; Wang, Y.; Mei, H. Dynamic-MUSIC: Accurate Device-Free Indoor Localization. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Heidelberg, Germany, 12–16 September 2016; pp. 196–207.
45. Qian, K.; Wu, C.; Yang, Z.; Liu, Y.; Jamieson, K. Widar: Decimeter-Level Passive Tracking via Velocity Monitoring with Commodity Wi-Fi. In Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Chennai, India, 10–14 July 2017; pp. 1–10.
46. Lin, T.; Kong, L.; Stich, S.U.; Jaggi, M. Ensemble distillation for robust model fusion in federated learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 2351–2363.
47. Zhang, T.; Xu, C.; Lian, Y.; Tian, H.; Kang, J.; Kuang, X.; Niyato, D. When Moving Target Defense Meets Attack Prediction in Digital Twins: A Convolutional and Hierarchical Reinforcement Learning Approach. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 3293–3305. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.