FedHAR: Semi-Supervised Online Learning for Personalized Federated Human Activity Recognition

Hongzheng Yu, Zekai Chen, Xiao Zhang, Xu Chen, Fuzhen Zhuang, Hui Xiong, *IEEE Fellow* and Xiuzhen Cheng, *IEEE Fellow*

Abstract—The advancement of smartphone sensors and wearable devices has enabled a new paradigm for smart human activity recognition (HAR), which has a broad range of applications in healthcare and smart cities. However, there are four challenges, *privacy preservation*, *label scarcity*, *real-timing*, and *heterogeneity patterns*, to be addressed before HAR can be more applicable in real-world scenarios. To this end, in this paper, we propose a personalized federated HAR framework, named *FedHAR*, to overcome all the above obstacles. Specially, as federated learning, *FedHAR* performs distributed learning, which allows training data to be kept local to protect users' privacy. Also, for each client without activity labels, in *FedHAR*, we design an algorithm to compute unsupervised gradients under the *consistency training* proposition and an unsupervised gradient aggregation strategy is developed for overcoming the concept drift and convergence instability issues in online federated learning process. Finally, extensive experiments are conducted using two diverse real-world HAR datasets to show the advantages of *FedHAR* over state-of-the-art methods. In addition, when fine-tuning each unlabeled client, personalized *FedHAR* can achieve additional 10% improvement across all metrics on average.

Index Terms—Human activity recognition, Federated learning, Semi-supervised learning, Online learning

1 Introduction

Human activity recognition (HAR) aims to detect human physical activities in real-world scenarios, allowing intelligent systems to assist individuals with improving the quality of life in many areas such as healthcare, smart cities, etc. In recent years, HAR via smart sensing has drawn rapidly growing interests from both academia and industry [1] [2] [3]. On top of the powerful modern ubiquitous computing techniques, different sensors (e.g., accelerometers, gyroscope, etc.) embedded in individuals' smartphones or smart wearable devices are utilized to measure human activities in various domains such as medicinal services, digital entertainments, public security, and many other areas [4].

The sensor-based HAR has its own distinguishing characteristics and challenges: (1) *Privacy preservation.* The sensing data carries lots of users' privacy information, which are often highly sensitive. (2) *Label scarcity.* Labeled activity data is always limited. It is costly to obtain labeling feedback regarding sensing data, which could also impose a heavy burden on users. (3) *Real-timing.* The sensing measurements are generated consecutively, which can be considered as a real-time online data stream. The *online* signifies that the entire training data need not be stored in memory, and the model should instantly respond to adjustments along with the afresh generated sensor data. (4) *Heterogeneity patterns.* Different individuals' activity patterns are considerably heterogeneous and diverse due to either demographic or other inherent physical distinctions.

However, existing works on HAR only addressed a part of the issues above, and little effort attempts to tackle all the above challenges within a single framework. For instance, some deep learning techniques, such as recurrent neural network (RNN) based methods [1] [5] or convolution neural network (CNN) based methods [2] [6], which are usually trained offline in a centralized way. DeepSense [7] is a framework with architecture using both CNN and RNN to model relatedness among different sensors and capture long-term dependencies of the temporal sensing data. AttnSense [8] identified human activity from a multi-modality aspect by combining the attention mechanism with CNN-RNN architecture to fuse sensor data. Furthermore, [9] conducted HAR in a federated learning framework to address the *privacy preservation* concern and achieved relatively considerable performance compared with centralized models.

- *This work was supported by the National Key Research and Development Program of China under Grant No. 2021ZD0113602; the National Natural Science Foundation of China (No. 62176014, No. U20A20159, No. U1711265, No. 61972432); Shandong Provincial Natural Science Foundation of China under Grant ZR2021QF044; the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No.2017ZT07X355);the Pearl River Talent Recruitment Program (No.2017GC010465).*
- *Xiao Zhang, Hongzheng Yu, and Xiuzhen Cheng are with the School of Computer Science and Technology, Shandong University, Qingdao 266237, China. Email: xiaozhang@sdu.edu.cn, honnzhengyu@foxmail.com, xzcheng@sdu.edu.cn.*
- *Zekai Chen is with the Department of Computer Science, George Washington University, Washington, D.C., 20052, US. Email: zech_chan@gwu.edu.*
- *Fuzhen Zhuang is with Institute of Artificial Intelligence, and SKLSDE, School of Computer Science, Beihang University, Beijing 100191, China. Email: zhuangfuzhen@buaa.edu.cn.*
- *Xu Chen is with the School of Computer Science and Engineering, Sun Yat-sen University (SYSU), Guangzhou, China. Email: chenxu35@mail.sysu.edu.cn.*
- *Hui Xiong is with Artificial Intelligence Thrust, The Hong Kong University of Science and Technology, Guangzhou, China. Email: xionghui@ust.hk.*
- *The corresponding author is Xiao Zhang.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3136853, IEEE Transactions on Mobile Computing

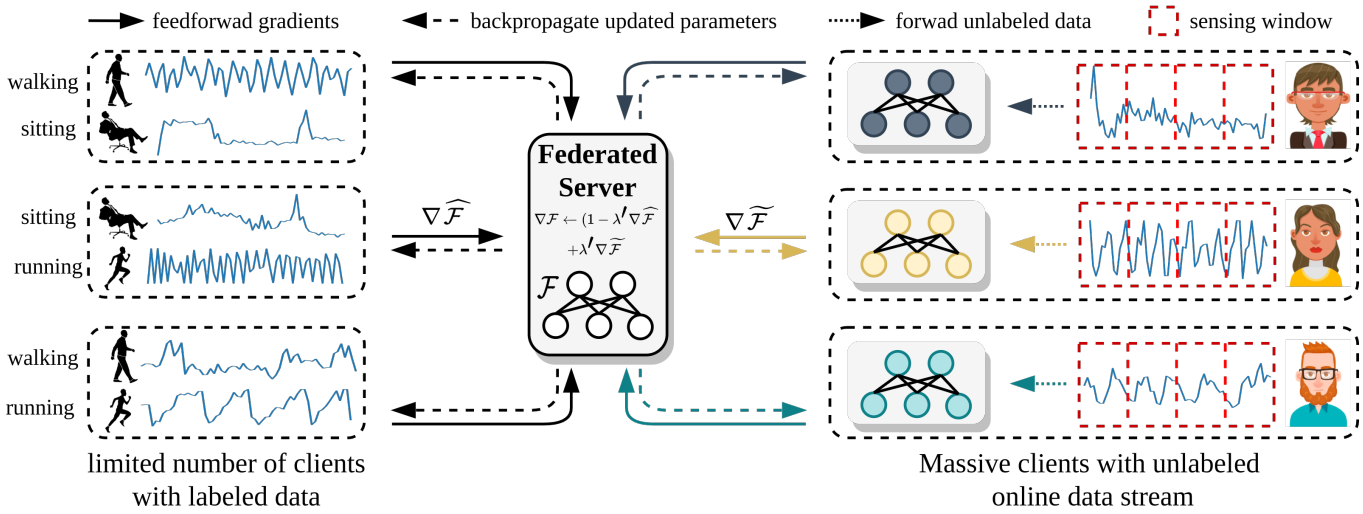


Fig. 1. The overview of proposed *FedHAR* framework. Generally, our method can be divided into the following steps: (1) Computing gradients from all the labeled clients and unlabeled clients; (2) A semi-supervised learning loss is designed to aggregate gradients of all the clients; (3) Backpropagating the updated parameters of model $\mathcal{F}$ to all clients from the server; (4) After the general model $\mathcal{F}$ is trained well, local model of each unlabeled client is further fine-tuned using *PerFedHAR* based on each client's private unlabeled real-time stream sensing data.

To overcome the *label scarcity* issue, RSAR [10] proposed a semi-supervised learning framework, in which $\ell_{2,1}$ minimization was used on loss function to improve recognition performance. Zeng et al. [11] presented semi-supervised CNNs to learn hidden features from labeled and unlabeled data for HAR. Moreover, literature [12] proposed a semi-supervised federated learning framework considering both *privacy preservation* problem and *scarce labels* problem, in which autoencoders were utilized for local models to learn representations and LSTM was for the global classifier. To address the *real-time* challenge, literature [13] utilized both CNN-extracted features and statistical features for HAR and limited the time series length up to $1s$ to make real-time HAR possible. Bhat et al. [14] proposed the first online HAR framework based on online reinforcement learning using policy gradient, which could train online to adapt to users. As for the *heterogeneity* challenge, Miu et al. [15] proposed an online active learning framework first to collect user-provided annotations and then bootstrap personalized HAR models. Nevertheless, none of the existing works addressed all the mentioned challenges on HAR in a holistic framework.

Along this line, we propose *FedHAR*, a personalized federated HAR framework based on semi-supervised online learning to overcome all the obstacles. The general overview of *FedHAR* is shown in Fig. 1. Federated learning [16] is a distributed machine learning framework that can keep training data local to protect users' privacy. As shown in the figure, clients are divided into two groups: a limited number of *clients with activity labels* and massive *clients without activity labels* but simultaneously generate the real-time sensing data stream. From the framework, the steps of *FedHAR* are exhibited as follows. (1) Computing supervised gradients from each labeled clients. For each client without activity labels, we design a novel algorithm to compute unsupervised gradients under the *consistency training* assumption. (2) A semi-supervised learning loss is designed to aggregate gradients from both labeled data and unlabeled data. Particularly, we design an *unsupervised gradients aggregation* strategy to overcome the *concept drift* and *convergence instability* issues in online learning. (3) Updated parameters were then broadcasted to all clients from the server. (4) After the prediction model finishes training, the updated local model of each unlabeled client is further personalized by aggregating overall supervised gradients and individual unsupervised gradients using each client's own online sensing stream based on *PerFedHAR*. Extensive experiments are conducted on two diverse real-world HAR datasets, *FedHAR* demonstrates its superiority over other state-of-the-arts. In addition, when fine-tuning each unlabeled client, *PerFedHAR* can achieve about additional **+10%** improvement across all metrics on two datasets on average. We have made our source code public on github.[1]

The **contributions** of our paper are summarized as follows:

- We proposed a general semi-supervised online learning framework for personalized federated human activity recognition (*FedHAR*). To the best of our knowledge, this is the first work addressing the *privacy preservation*, *label scarcity*, *Real-timing*, and *heterogeneity* challenges of HAR into a unified framework simultaneously.
- Within the unified framework, *FedHAR* can utilize only a small number of labeled clients with limited samples to train a federated HAR model with competitive performance along with massive real-time stream sensing data produced by unlabeled clients.
- A novel algorithm is designed to compute unsupervised gradients under the *consistency training* proposition based on temporal data characteristics of HAR tasks.

1. The code is available at https://github.com/fedhar/fedhar.git.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3136853, IEEE Transactions on Mobile Computing

- We design an *unsupervised gradients aggregation* strategy to overcome the *concept drift* and *convergence instability* problem in online learning. A semi-supervised learning loss is designed to aggregate gradients from all the label clients and unlabeled clients.
- Extensive experiments are conducted on two diverse real-world HAR datasets, *FedHAR* demonstrates its superiority over other state-of-the-arts. Additionally, when fine-tuning each unlabeled client, *PerFedHAR* can further gain superior performance improvement across all metrics on two datasets on average.

TABLE 1
Notations used in the paper.

τ	length of each sensing time window (seconds)
S	number of types of sensors within each device
K	number of devices within each client
N	number of clients with local labeled data
M	number of clients producing online data streams without labels
$\mathcal{X}$	labeled data from all N clients
$\widetilde{\mathcal{X}}$	unlabeled data streams from all M clients
$\mathcal{X}_{n,t}^{s,k}$	labeled time series measured by the s-th sensor in the k-th device from the n-th client within the t-th sensing window
$\widetilde{\mathcal{X}}_{m,t}^{s,k}$	unlabeled data stream measured by the s-th sensor in the k-th device from the m-th client within the t-th sensing window
$T^{s,k}$	number of recordings from the s-th sensor in the k-th device within a sensing window
d^s	dimensionality of sensing data from sensor s
$\mathcal{F}$	general global model stored in server

2 Related Work

2.1 Human activity recognition

In recent years, human activity recognition (HAR) based on smart sensing data has been successfully applied in many fields such as medical service and health [17], lifestyle researching [18] [19]. Existing methods usually utilized the deep learning techniques and achieved satisfactory performance [8] [20] [21] [22]. For example, [1] utilized ensembles of LSTM learners for HAR scenario to address the imbalanced datasets and the data quality problem. DeepSense [7] adopted CNN to learn the local interactions within each sensing modality and global interactions among different sensor inputs. In addition, RNN was utilized to learn the inter-interval relationships of the time series sensor data. AttnSense [8] combined the attention mechanism with CNN-RNN architecture from a multi-modality aspect to identify human activities.

2.2 Federated learning for HAR

Federated Learning is a distributed machine learning framework, in which a model can be trained across multiple devices holding local data without sharing them [23] [24] [25]. Generally, the parameters are aggregated in a central server with some widely used algorithms, such as FedAvg [16], FedSGD [26]. Literature [27] proposed personalized FedAvg based on the model-agnostic meta learning. The personalized clients must utilize labeled data for training, which cannot be applied in unlabeled data streams. Some methods has been proposed for HAR to protect users' privacy based on federated learning. For instance, Sozinov et al. [9] utilized two different models: polynomial logistic regression and deep neural networks in federated learning framework, which achieved acceptable performance compared with centralized algorithms. PMF [28] was a privacy preserving mobility prediction framework based on federated learning, in which a group optimization method was designed for training.

2.3 Semi-supervised learning for HAR

Since it is costly to obtain labeling activities regarding sensing data, some researchers have tried to apply the semi-supervised learning methods for HAR tasks. Zeng et al. [11] first introduced semi-supervised CNN for HAR, in which the CNN based encoder-decoder and convolutional ladder network were used to learn better high-level features. It can reduce more than 90% labeled data while ensuring the performance of model. Yao et al. [10] proposed a semi-supervised HAR algorithm based on manifold structure, which assumes that data in the path through high density regions on data manifold should have same label with higher probability. Zhao et al. [12] proposed a semi-supervised federated learning framework for HAR, in which auto-encoders were utilized for local models to learn representations and LSTM was for the global classifier.

2.4 Online learning for HAR

Because the generated sensing measurements can be considered as a real-time online data stream, online learning techniques have been applied for HAR [29] [30] [31] [32]. Ignatov et al. [13] utilized both CNN-extracted features and statistical features for HAR and limited the time series length to make real-time HAR possible. Miu et al. [15] constructed a online active learning framework to continuously monitor each individual's activities with annotations to bootstrap a personalised model. Bhat et al. [14] proposed the first online HAR framework based on online reinforcement learning using policy gradient, which could train online to adapt to users. In summary, existing works on HAR only addressed a part of the mentioned four challenges: *privacy preservation*, *label scarce*, *real-time*, and *heterogeneity challenges*. Therefore, the focus of our work is to address all the above challenges with one general solution.

3 Problem Description

In a typical federated learning setting consisting of one central cloud server and a total number of M clients, mobile devices collaboratively learn a shared prediction model through the server while keeping all clients' training data local. Under this circumstance, we consider a specific HAR scenario where each client consists of K different devices with each device containing S types of sensors embedded. Periodically, each sensor generates a sequence of measurements for a past time window that lasts τ seconds. Since the sensing frequency of different sensors in different devices might differ, the actual number of recording frames may also vary. Hence we use $T^{s,k}$ to denote the length of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3136853, IEEE Transactions on Mobile Computing

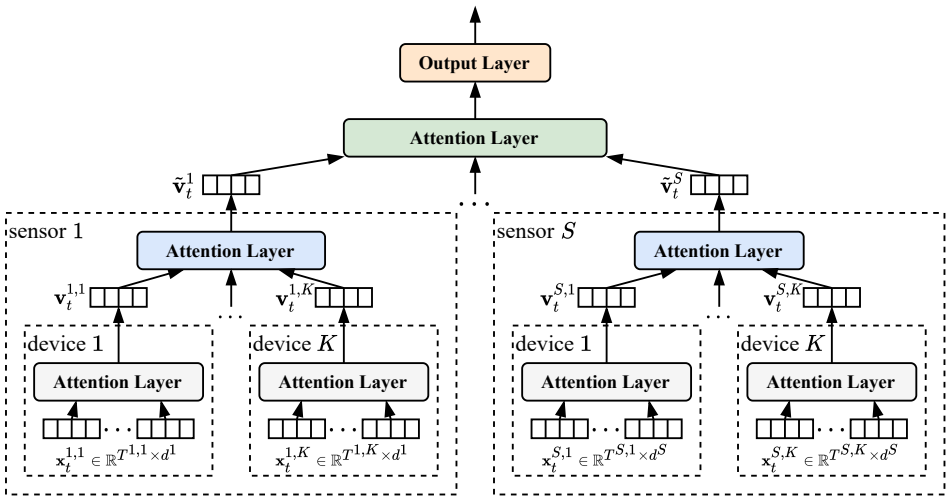


Fig. 2. The architecture of our proposed hierarchical attention architecture. It has 3 kinds of attention layers using to merge data in one time window, data in each devices and data from each types of sensor.

fragmented time series collected from the s-th sensor in k-th device (e.g., $T^{s,k} = 125$ if the sensing frequency is 50Hz with $\tau = 2.5s$). Considering only a small subset of clients can provide labeling feedback in most real-life situations, we let N ($N \ll M$) denote a small number of clients that able to provide a labeled local dataset $\mathcal{X} = \{\mathcal{X}_n\}_{1 \le n \le N}$ with corresponding labels $\mathcal{Y} = \{\mathcal{Y}_n\}_{1 \le n \le N}$; meanwhile, all M clients consecutively produce unlabeled data stream $\widetilde{\mathcal{X}} = \{\widetilde{\mathcal{X}}_m\}_{1 \le m \le M}$. Specifically, for labeled data $\mathcal{X}$, we denote $\mathcal{X}_{n,t}^{s,k} \in \mathbb{R}^{T^{s,k} \times d^s}$ to be the labeled raw time series of the t-th sensing window measured by the s-th sensor in k-th device for n-th client, where d^s represents the sensing dimensionality of sensor s (e.g., 3 for accelerometer, representing the acceleration in the axis of x, y, and z). We then denote $\mathcal{Y}_{n,t}$ as the corresponding labels. Similarly, $\widetilde{\mathcal{X}}_{m,t}^{s,k} \in \mathbb{R}^{T^{s,k} \times d^s}$ represents the online unlabeled sensing data of the t-th sensing window measured by the s-th type of sensor in k-th device on m-th client.

Therefore, our primary goal is to learn a general deep neural network for HAR in a federated learning fashion with limited labeled data but copious unlabeled data. Our second goal is to strengthen user-specific prediction accuracy personalized by the way different individuals act through semi-supervised online learning. Namely, instead of immediately applying the generally optimized global function $\mathcal{F}$ on all clients, we further customize an optimized model $\mathcal{F}'_m$ for any m-th client, where $m \in \{1, \cdots, M\}$.

4 Federated Online Semi-supervised Learning Framework

In this section, we introduce our federated semi-supervised online learning framework on HAR in a two-stage fashion based on a hierarchical attention neural network architecture. Specifically, we first learn a general prediction model on the server using both labeled data and unlabeled data stream through our novel federated semi-supervised online learning algorithm **FedHAR**. Then we propose the second-stage local personalized strategy **PerFedHAR** to further enhance the prediction accuracy for individual clients.

4.1 FedHAR algorithm

The core of this idea is to apply semi-supervised online learning on fragmented time series in a federated learning fashion. Considering each client might be equipped with various sensors across multiple devices, we thus devise the general model architecture from the perspective of feature fusion. Inspired by [8], we first introduce a hierarchical attention architecture for each client to fuse various measurements collected from different sensor across devices for optimal representation learning, as shown in Fig. 2.

4.1.1 Hierarchical attention architecture

Generally, we devise three levels of attention layers for the alignment of different level features. For clarification, we let $\mathbf{x}_t^{s,k} \in \mathbb{R}^{T^{s,k} \times d^s}$ be the original input temporal data collected during t-th sensing window of s-th sensor in k-th device from any potential client. Each attention layer genuinely consists of (1) one linear layer to align the sensing dimensionality (from d^s to d) and (2) an attention layer to model higher-level representations. For instance, the **input-level** attention layers attend to model higher-level representations from original input $\mathbf{x}_t^{s,k}$ along the chronological dimension as:

$$\mathbf{v}_t^{s,k} = \sum_{j=1}^{T^{s,k}} \alpha_j \mathbf{x}_{t,j}^{s,k} w_1, \tag{1}$$

where $\mathbf{x}_{t,j}^{s,k} \in \mathbb{R}^{d^s}$ represents the sensing value at j-th timestamp, $\mathbf{v}_t^{s,k} \in \mathbb{R}^d$ denotes the output representation, $w_1 \in \mathbb{R}^{d^s \times d}$ is the weight matrix of the first linear layer, while α_j is the attention score which measures the contribution of each sensing value. The attention score is computed as following:

$$\alpha_j = \frac{exp(\phi(\mathbf{x}_{t,j}^{s,k} w_2))}{\sum_j exp(\phi(\mathbf{x}_{t,j}^{s,k} w_2))}, \tag{2}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3136853, IEEE Transactions on Mobile Computing

where $\phi(\cdot)$ denotes a nonlinear activation function we utilize, $w_2 \in \mathbb{R}^{d^s \times 1}$ is the weight matrix of learnable parameters. The **device-level** and **sensor-level** attention layers then utilize the same attention mechanism to incorporate more advanced representations of each level from bottom to up, making it a hierarchical attention learning architecture. Through feature fusing, attention layers can also eliminate the difference in input sequences' length across different sensing windows, caused by various sensors' different sensing frequency.

Algorithm 1: FedHAR algorithm

Input: Local labeled data $\mathcal{X}$ with respective labels $\mathcal{Y}$ from N clients; real-time unlabeled data stream $\widetilde{\mathcal{X}}$ hitherto generated by all M clients; M_b randomly selected clients with unlabeled data in one training iteration; the initial weight of unsupervised loss λ; a hyper-parameter r_λ to balance the supervised and unsupervised loss; agg rounds of aggregation regarding the unsupervised gradients

Output: $\mathcal{F}$: a generalized prediction model stored in sever

Let $\mathcal{X}_n$ be the labeled local data from n-th client and $\widetilde{\mathcal{X}}_m$ be the unlabeled data stream provided by m-th client
randomly initialize neural network $\mathcal{F}$
$\lambda \leftarrow 0$; iteration step $r \leftarrow 0$
while *Training* **do**
  send $\mathcal{F}$ from server to all clients
  randomly sample M_b clients out of all M clients that produce unlabeled data stream and denote this subset as $\mathcal{B}_M$
  for $\forall n \in \{1, \cdots, N\}$ ***in parallel*** **do**
    $\nabla\widehat{\mathcal{F}}_n \leftarrow$ compute supervised gradients using $\{\mathcal{X}_n, \mathcal{Y}_n\}$
    upload $\nabla\widehat{\mathcal{F}}_n$ to the server
  end
  for $\forall m \in \mathcal{B}_M$ ***in parallel*** **do**
    for $a \leftarrow 1, \cdots, agg$ **do**
      $\nabla\widetilde{\mathcal{F}}_m^a \leftarrow$ compute unsupervised gradients on $\widetilde{\mathcal{X}}_m$ using Alg. 2
      upload $\nabla\widehat{\mathcal{F}}_m^a$ to the server
    end
    $\nabla\widetilde{\mathcal{F}}_m \leftarrow \frac{1}{agg}\sum_{a=1}^{agg}\widetilde{\mathcal{F}}_m^a$
  end
  $\nabla\widehat{\mathcal{F}} \leftarrow \frac{1}{N}\sum_{n=1}^{N}\nabla\widehat{\mathcal{F}}_n$; $\nabla\widetilde{\mathcal{F}} \leftarrow \frac{1}{M_b}\sum_{m\in\mathcal{B}_M}\nabla\widetilde{\mathcal{F}}_m$
  if $r \le r_\lambda$ **then**
    $\lambda' \leftarrow \frac{r}{r_\lambda} \times \lambda$
  else
    $\lambda' \leftarrow \lambda$
  end
  $\nabla\mathcal{F} \leftarrow (1-\lambda')\nabla\widehat{\mathcal{F}} + \lambda'\nabla\widetilde{\mathcal{F}}$
  update $\mathcal{F}$ using **SGD**
  $r \leftarrow r + 1$
end
return $\mathcal{F}$

4.1.2 Semi-supervised online learning

Based on the hierarchical attention network architecture, we propose a semi-supervised learning strategy for online human activity recognition tasks, and we now introduce this method.

Algorithm 2: Unsupervised Gradients Computation

Input: Unlabeled data stream $\widetilde{\mathcal{X}}_m$ collected from m-th client; time of sensing window τ; $\tau_s$as the communication interval between client and server; current model $\mathcal{F}$ on client m

Output: $\nabla\widetilde{\mathcal{F}}_m$: computed gradients with respect to unsupervised loss on m-th client

we denote $\widetilde{\mathcal{X}}_{m,0}$ as data stream measured within last sensing window saved in client m
for $t \leftarrow 1, \cdots, \lceil\frac{\tau_s}{\tau}\rceil$ **do**
  $\widetilde{\mathcal{X}}_{m,t}$ denotes the online unlabeled sequences within t-th sensing window on client m
  $[\widetilde{\mathcal{Y}}_{m,t-1}, \widetilde{\mathcal{Y}}_{m,t}] \leftarrow \mathcal{F}(\widetilde{\mathcal{X}}_{m,t-1}, \widetilde{\mathcal{X}}_{m,t})$
  $\nabla\widetilde{\mathcal{F}}_m \leftarrow \frac{1}{t} \times \frac{\partial MSE(\widetilde{\mathcal{Y}}_{m,t-1}, \widetilde{\mathcal{Y}}_{m,t})}{\partial\mathcal{F}} + \frac{t-1}{t}\nabla\widetilde{\mathcal{F}}_m$
end
save $\widetilde{\mathcal{X}}_{m,t}$ to be used for the next round on client m
return $\nabla\widetilde{\mathcal{F}}_m$

Algorithm 3: PerFedHAR algorithm for personalization

Input: Local labeled data $\{\mathcal{X}, \mathcal{Y}\}$ from N clients; unlabeled data stream $\widetilde{\mathcal{X}}$; let $\mathcal{R}_M$ denote a subset of M_r clients in need of personalizing; the initial weight of unsupervised loss λ; the general prediction model $\mathcal{F}$ in sever

Output: $\{\mathcal{F}'_m\}_{m\in\mathcal{R}_M}$: a set of personalized neural networks for clients in need

$\forall m \in \mathcal{R}_M$, $\mathcal{F}'_m \leftarrow \mathcal{F}$ (from server to local)
while *Training* **do**
  for $\forall m \in \mathcal{R}_M$ ***in parallel*** **do**
    for $\forall n \in \{1, \cdots, N\}$ ***in parallel*** **do**
      $\nabla\widehat{\mathcal{F}}'_n \leftarrow$ compute supervised gradients using $\{\mathcal{X}_n, \mathcal{Y}_n\}$
    end
    $\nabla\widehat{\mathcal{F}}'_m \leftarrow \frac{1}{N}\sum_{n=1}^{N}\widehat{\mathcal{F}}'_n$
    $\nabla\widetilde{\mathcal{F}}'_m \leftarrow$ compute unsupervised gradients with $\widetilde{\mathcal{X}}_m$ using Alg. 2
    $\nabla\mathcal{F}'_m \leftarrow (1-\lambda)\nabla\widehat{\mathcal{F}}' + \lambda\nabla\widetilde{\mathcal{F}}'_m$
    update $\mathcal{F}'_m$ using **SGD** with gradients $\nabla\mathcal{F}'_m$
  end
end
return $\{\mathcal{F}'_m\}_{m\in\mathcal{R}_M}$

4.1.2.1 Consistency Training on Human Activity.:

Previous works utilize **consistency regularization** and apply data augmentation to semi-supervised learning by leveraging the idea that a classifier should output the same class distribution for an unlabeled example even after it has been augmented. This approach makes intuitive sense since a good model should be robust to small noise injected in an input. However, under this circumstance, different augmentation strategies may result in different influences, making the whole training phase hard to control. Also, it is not straightforward to design an appropriate augmentation procedure for time series data. Thus, we design the following learning scheme to be in line with the task characteristics.

Considering temporal data characteristics of tangible human activity, we argue that the time series should always reflect an individual's coherent and unified physical status within a relatively short time interval. Thus, we expect two

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3136853, IEEE Transactions on Mobile Computing

neighboring online series to obtain similar representations. In general, for any two adjacent sensing windows, we add a constraint to keep the model's outputs regarding these two stream sequences as close as possible. Specifically, we denote $\widetilde{\mathbf{x}}_t, \widetilde{\mathbf{x}}_{t+1}$ as two unlabeled sequences measured within t-th and $(t+1)$-th sensing window respectively from any randomly selected client while $\widetilde{\mathbf{y}}_t, \widetilde{\mathbf{y}}_{t+1}$ denote the corresponding model outputs. In this case, we apply MSE (mean square error or L_2 norm) as the divergence distance between two probability distributions $\widetilde{\mathbf{y}}_t, \widetilde{\mathbf{y}}_{t+1}$ to implement the consistency training on human activity (see Alg. 2). Thus, our loss objective for the unsupervised part is as following:

$$\mathcal{L}_{unsup} = \mathcal{D}[p(\widetilde{\mathbf{y}}_t|\widetilde{\mathbf{x}}_t, \mathcal{F}), p(\widetilde{\mathbf{y}}_{t+1}|\widetilde{\mathbf{x}}_{t+1}, \mathcal{F})], \tag{3}$$

where $\mathcal{D}$ is the divergence distance metric between $\widetilde{\mathbf{y}}_t$ and $\widetilde{\mathbf{y}}_{t+1}$ and we select mean-square error in this study.

4.1.2.2 Why this works?: The fact is that not all neighboring sensing windows essentially reflect the same action type yet most of the adjacent windows do. Therefore, we theoretically analyze why the consistency training can actually work through eliminating the negative effect and improve the model performance as the model keeps learning.

Theorem 4.1. *Suppose there are t windows of sensing measurements in total, and we assume there is one specific sensing window belongs to one particular action j with probability $\mathcal{P}_j$ while other windows consistently reflect another specific action type. Let $\mathcal{P}_\mathcal{A}$ denote the overall probability of inconsistent adjacent windows could happen during training, which is given as:*

$$\mathcal{P}_\mathcal{A} = \sum_j \mathcal{P}_j^2 (1-\mathcal{P}_j)^{t-2},$$

Consequently, this probability has an upper bound of $\mathcal{O}(\epsilon)$ with $t = \mathcal{O}(C/\epsilon)$, where C is an empirical estimation of the number of all possible action types that an individual could own in real-life.

Proof. If there is only one specific sensing window that denotes to a different action j, then it is easy to get:

$$\mathcal{P}_\mathcal{A} = \sum_j \mathcal{P}_j^2 (1-\mathcal{P}_j)^{t-2},$$

since there are two adjacent windows that represent different actions. To find the upper bound of this probability, we need to find out the maximum value of $\sum_j \mathcal{P}_j^2(1-\mathcal{P}_j)^{t-2}$. We then define the following optimization function:

$$\min_{\mathcal{P}} -\sum_j \mathcal{P}_j^2 (1-\mathcal{P}_j)^{t-2}$$
$$s.t. \sum_j \mathcal{P}_j = 1,$$

The problem is a convex optimization problem and we construct its Lagrangian dual function:

$$\sum_j \mathcal{P}_j^2 (1-\mathcal{P}_j)^{t-2} - \lambda(\sum_j \mathcal{P}_j - 1),$$

Using the KKT condition, we can take derivatives to $\mathcal{P}_j$ and set it to zero. Then we have

$$\lambda = (\mathcal{P}_j(2-t\mathcal{P}_j))(1-\mathcal{P}_j)^{t-3},$$

Hence $\mathcal{P}_i = \mathcal{P}_j$ for any $i \neq j$. Suppose $\mathcal{P}_j = \frac{1}{C}$, where C is the total number of all possible actions, then we have

$$\mathcal{P}_\mathcal{A} \leq (1-\frac{1}{C})^t = \exp^{t\log(1-\frac{1}{C})} \leq \exp^{-t/C},$$

If we let $t = \mathcal{O}(C/\epsilon)$, we have $\mathcal{P}_\mathcal{A} = \mathcal{O}(\epsilon)$.

From the theorem, we can see that this empirical number C governs the potential model performance. However, we argue this C is often a trivial value in most real-life scenarios since the basic types of movements are relatively limited for individuals. Therefore, as the total sensing time grows, the negative effect brought by inconsistent training samples can be gradually dominated by positive transfer. Additionally, it answers the question **why hierarchical attention instead of RNN?** The conventional recurrent neural network is suitable for dealing with long-sequence temporal data due to its internal memory mechanism. However, we argue that RNN might not work in this study because (1) sequential data of each sensing window is not long enough for RNN to model the temporal dependencies; (2) more vitally, the inconsistency error could be infinitely magnified due to the cumulative product of matrices.

4.1.2.3 Unsupervised Gradients Aggregation.: As we adopt a federated learning paradigm, all clients would then compute the gradient descent corresponding to the given supervised and unsupervised objectives and merely upload the updated gradients to the cloud server using encrypted communication while keeping all training data local. However, as discussed in section 1, it is common for individuals to maintain a particular action state within a relatively long duration. Therefore these homogeneous sensing data representations can easily dominate the learning scheme over other human activity varieties, known as **concept drift**. Moreover, if we only compute unsupervised gradients over few consecutive samples, it may eject extra noisy signals into the process of stochastic gradient descent, also known as **convergence instability**.

Hence we devise an **unsupervised gradients aggregation** strategy to overcome these obstacles. Specifically, during each online training iteration, we randomly select M_b clients that continuously produce an online data stream, and we denote the set of clients as $\mathcal{B}_M$. Since the number of unlabeled clients is usually vast, it can significantly improve training efficiency and reduce computation and communication cost compared with requiring gradients from all clients. For $\forall m \in \mathcal{B}_M$, we compute multiple rounds of unsupervised gradients and address the average of aggregated gradients as the final update that the server model utilizes (see Alg. 1):

$$\nabla\widetilde{\mathcal{F}}_m = \frac{1}{agg}\sum_{a=1}^{agg} \nabla\widetilde{\mathcal{F}}_m^a, \tag{4}$$

where $\nabla\widetilde{\mathcal{F}}_m^a$ represents computed unsupervised gradients immediately transferred to the server in the a-th communication with m-th client, and agg denotes the total rounds of aggregation, which also denotes the communication rounds. In practice, the time interval τ_s between every communication is much longer than the length of each sensing window τ due to the expensive communication cost. We then utilize an average of computed unsupervised gradients

using online stream data across each sensing window to be the ultimate gradient for a single aggregation (see Alg. 1).

4.1.2.4 Training Strategy.: To encourage better convergence, instead of optimizing over both supervised and unsupervised objectives simultaneously in the early training stage, we develop a simple yet effective strategy to gradually enlarge unsupervised gradients and anneal the supervised gradients across training iterations inspired by curriculum learning [33]. Specifically, by setting the maximum weight of unsupervised gradients, we gradually enlarge the coefficient for unsupervised gradients before the learning iteration reaches r_λ (see Alg. 1) using $\lambda' \leftarrow \frac{r}{r_\lambda} \times \lambda$. After the clients send the updated gradients back to server, the server uses SGD with gradient $\nabla \mathcal{F}$ backpropagated to update the whole neural network.

4.2 Online semi-supervised personalizing

As the FedHAR scheme can develop a common output for all clients, it is not adaptive for each individual. In particular, in the heterogeneous settings where the underlying data distribution of clients is not identical, the resulted global model obtained by minimizing the average loss could perform arbitrarily poorly once applied to the local dataset of each specific client [34]. In other words, a generalized solution is yet insufficient due to the lack of personalization. Therefore, we further propose a second-stage personalized semi-supervised online learning strategy PerFedHAR to tackle this challenge (see Alg. 3).

Considering the issue of convergence instability and concept drift that training with unlabeled data only might bring, we again utilize the supervised gradients from clients with labeled local data in order to provide a good starting point for the personalized neural network. Thus, our personalized strategy is still in a fashion of semi-supervised learning. In detail, for one specific unlabeled data client m, we first initialize the personalized model using downloaded general model $\mathcal{F}'_m \leftarrow \mathcal{F}$ from the server and then start training. Unlike the first-stage FedHAR, not all clients have personalized needs. We then denote $\mathcal{R}_M$ as a subset of M_r clients requiring further personalized service. During each training iteration, gradient descent's principal scheme is similar to FedHAR, except that these clients convey gradients to sever only once rather than agg times. The detailed algorithm is shown in Alg. 3.

5 Experiments

In this section, we demonstrate the effectiveness of our proposed approach by conducting extensive experiments on two real-world HAR datasets. First, we compare our FedHAR framework with a wide range of state-of-the-arts to evaluate the effectiveness of our novel federated semi-supervised online training algorithm. Second, we show that our personalized training strategy PerFedHAR can further bring significant improvements based on the global model.

5.1 Dataset description

The experiments are conducted on two real-world datasets:

RealWorld[2] [35] A sensor-based HAR dataset collected from 15 probands. It contained 8 kinds of activities, including *running, standing, lying, sitting, walking, jumping, climbing stairs down* and *up*. Every person wore sensing devices on 7 body parts, including *chest, one forearm, head, shin, one thigh, one upper arm* and *waist*. Three sensors of each device were chosen in our experiment, including *accelerometer, gyroscope,* and *magnetometer*.

HAR-UCI[3] [36]. This is a public sensor-based HAR dataset published in UC Irvine Machine Learning Repository (UCI). It was built from the recordings of 30 subjects performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors. Those 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (walking, walking upstairs, walking downstairs, sitting, standing, laying) wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, 3-axial linear acceleration and 3-axial angular velocity were captured at a constant rate of 50Hz.

In **RealWorld**, the total sensing time for each client is 90 minutes with each time window lasts 2.5 seconds. So for each client, there are 2118 windows (or samples) of measurements available in total. For **HAR-UCI** dataset, the total sensing time for each client is 8 minutes with each sensing window also lasts 2.5 seconds. In total, there are 187 sensing windows (or samples) available to use for each client.

5.2 Experimental setup

We evaluate our FedHAR algorithm under various experimental settings where we attempt different numbers of clients with local labeled data and total available sensing time. Expressly, we set the number of clients with local labeled data N to $3, 4, 5$ with different total sensing time as $3000, 4000, 5000$ seconds separately for **RealWorld** dataset. For **HAR-UCI** dataset, N is chosen as $4, 6, 8$ and total sensing time is set to $250, 350, 450$ seconds, separately. In all, we conduct experiments under nine different settings in total on both datasets. We adopt the Adam optimizer with a learning rate of $1e^{-3}$ by default. Besides, we set the maximum unsupervised loss weight λ in FedHAR to 0.2 with r_λ set as 400, the communication interval between server and client τ_s is set to 60 seconds. We set the aggregation rounds agg to 20, the number of randomly selected clients M_b without labels to 5 for **RealWorld** dataset, and 9 for **HAR-UCI** dataset. Also, the batch size for labeled data is chosen as 128. In order to prevent over-fitting, dropout strategy and L_2 norm are also applied in the proposed FedHAR model training.

When personalizing the general FedHAR model, we select N as 5 and total sensing time as 3000 seconds for **RealWorld** dataset, while N as 8 and total sensing time as 450 seconds for **HAR-UCI**.

Online Settings: Unlabeled clients are trained with online learning way, whose training data are from real-time

2. It can be download inhttps://sensor.informatik.uni-mannheim.de/#dataset_realworld

3. It can be download in https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones#.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3136853, IEEE Transactions on Mobile Computing

TABLE 2
Prediction results on RealWorld dataset with best model performance in bold and second-best results with underlines. $*-*$ denotes the number of clients with labeled data $-$ the total sensing time (seconds) of each client.

Model	Metric	3-1200	3-1600	3-2000	4-1200	4-1600	4-2000	5-1200	5-1600	5-2000
HAR$_{sup}$	Accuracy	42.36	54.00	53.25	54.29	52.81	47.14	64.65	62.39	55.82
	F1-Score	40.63	50.16	48.88	50.30	48.91	42.68	61.11	59.44	51.22
DeepSense	Accuracy	45.33	52.63	53.02	57.31	59.71	61.85	58.94	59.14	61.44
	F1-Score	38.20	46.58	46.86	52.20	54.20	57.21	53.83	52.99	55.19
Pseudo-label	Accuracy	51.16	56.47	51.99	43.84	52.41	59.40	64.38	61.54	58.53
	F1-Score	47.29	53.15	47.50	38.19	48.03	54.95	61.76	59.09	54.23
Mean Teacher	Accuracy	34.11	32.62	37.21	39.41	36.93	34.83	43.40	33.88	31.89
	F1-Score	26.55	23.88	28.77	27.50	28.59	28.94	35.47	25.98	23.78
ICT	Accuracy	32.10	37.30	37.66	48.50	52.27	42.02	36.06	41.74	44.49
	F1-Score	27.41	31.24	32.77	42.03	49.04	37.50	31.43	37.04	41.46
FedAvg	Accuracy	41.36	42.36	53.53	46.74	48.34	53.40	49.56	55.77	50.68
	F1-Score	34.12	34.17	49.33	40.24	41.72	46.59	43.91	49.61	44.00
FedProx	Accuracy	45.23	48.62	48.15	56.98	52.68	56.14	56.10	61.14	62.93
	F1-Score	39.59	43.64	42.02	51.90	48.12	51.78	51.25	56.05	56.84
FedHAR$_{w/o\ agg}$	Accuracy	54.87	56.66	53.41	62.42	58.21	58.24	60.54	63.85	55.50
	F1-Score	51.69	52.85	49.61	58.79	53.47	53.69	56.91	60.43	53.62
FedHAR	Accuracy	**55.33**	**63.67**	**56.02**	**63.47**	**61.63**	**65.31**	**65.10**	**64.11**	**62.95**
	F1-Score	**51.80**	**60.48**	**50.81**	**59.39**	**57.75**	**62.77**	**62.44**	**61.10**	**57.70**

TABLE 3
Prediction results on HAR-UCI dataset with best model performance in bold and second-best results with underlines. $*-*$ denotes the number of clients with labeled data $-$ the total sensing time (seconds) of each client.

Model	Metric	4-100	4-140	4-180	6-100	6-140	6-180	8-100	8-140	8-180
HAR$_{sup}$	Accuracy	69.71	72.45	70.98	71.61	69.70	73.53	73.97	74.77	75.05
	F1-Score	67.40	69.92	68.61	69.22	67.48	71.11	72.10	72.55	73.13
DeepSense	Accuracy	72.11	71.57	70.16	75.76	75.98	73.33	77.56	75.39	74.24
	F1-Score	68.94	69.60	66.52	73.62	72.87	70.97	75.48	73.76	72.40
Pseudo-label	Accuracy	74.96	57.63	73.04	72.88	73.08	71.30	79.37	64.73	74.64
	F1-Score	72.05	49.54	70.41	70.93	70.51	68.21	78.10	61.62	72.74
Mean Teacher	Accuracy	63.12	44.75	38.25	47.36	56.87	58.28	40.61	41.59	57.98
	F1-Score	57.92	32.81	24.63	37.16	48.07	51.85	27.86	30.02	47.85
ICT	Accuracy	69.55	71.66	69.41	69.28	66.23	67.88	75.72	74.04	77.48
	F1-Score	67.45	68.33	66.60	67.66	63.13	65.51	74.56	72.13	76.49
FedAvg	Accuracy	59.28	62.39	67.47	71.45	64.31	64.66	68.97	67.69	68.23
	F1-Score	54.08	57.39	64.55	68.24	59.89	58.74	63.93	64.58	63.41
FedPorx	Accuracy	62.03	69.34	57.03	64.43	66.43	69.20	73.75	70.35	68.74
	F1-Score	58.61	67.22	49.94	60.03	62.05	65.48	71.55	65.78	65.87
FedHAR$_{w/o\ agg}$	Accuracy	70.68	**74.95**	75.35	72.07	73.02	73.94	75.73	70.57	77.23
	F1-Score	68.44	**72.36**	73.11	69.34	71.08	71.78	72.96	67.67	75.64
FedHAR	Accuracy	**76.32**	74.32	**78.68**	**81.51**	**82.33**	**77.21**	**82.61**	**76.48**	**80.74**
	F1-Score	**74.69**	71.98	**76.59**	**79.97**	**81.28**	**75.27**	**81.62**	**74.78**	**79.34**

data streams. The duration of the real-time activities is from τ(the length of time window, 2.5s in default) to 10 minutes. The data stream values in one time window can be missing. The data streams are processed using Fourier transform to convert time domain information into frequency information.

Platform. All models are trained on Nvida P100 16GB GPU in Ubuntu 20.04x64 with 8-core Intel CPU and 64GB RAM. Python 3.8 and Pytorch 1.7.1 are adopted in the experiments.

Architecture and Parameters. As described in Section 4.1, our model is a hierarchical attention-based neural network. In the first type of attention layer used to weight time series data, it contains 3 linear layer (# of neurons = 64, 128, 256). Each linear layer is with a LeakReLu activation function and a dropout layer (dropout=0.2). In the second type of attention layer used to weight $\{v_t^{s,k}\}$, it contains a linear layer (64 neurons), a LeakReLu activation function and a dropout layer (dropout=0.2). In the third type of attention layer used to weight $\{\overline{v}_t^s\}$, it contains a linear layer (32 neurons), a LeakReLu activation function and a dropout layer (dropout=0.2). In the output layer, it contains a linear layer (16 neurons), a LeakReLu activation, a Dropout layer (dropout=0.4) and a linear layer whose number of neurons

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3136853, IEEE Transactions on Mobile Computing

TABLE 4
Performance comparison between FedHAR and PerFedHAR on each unlabeled client under RealWorld dataset.

Model	Metric	Client															Average
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
FedHAR	Accuracy	60.94	77.81	57.19	51.46	58.33	36.51	86.67	67.76	65.99	65.89	90.42	90.31	89.22	91.04	91.77	72.09
	F1-Score	57.50	72.85	51.85	48.50	53.93	27.15	86.35	64.60	60.75	61.27	90.21	90.06	88.53	91.10	91.62	69.08
PerFedHAR	Accuracy	82.40	78.75	75.89	61.46	65.31	46.61	91.41	89.01	70.36	84.06	95.31	96.25	96.46	95.63	96.46	81.69
	F1-Score	82.01	74.52	74.77	60.71	57.43	39.59	91.36	88.92	66.70	83.60	95.33	96.26	96.45	95.63	96.47	79.98
$\Delta(\%)\uparrow$	Accuracy	**21.46**	**0.94**	**18.70**	**10.00**	**6.98**	**10.10**	**4.74**	**21.25**	**4.37**	**18.17**	**4.89**	**5.94**	**7.24**	**4.59**	**4.69**	**9.60**
	F1-Score	**24.51**	**1.67**	**22.92**	**12.21**	**3.50**	**12.44**	**5.01**	**24.32**	**5.95**	**22.33**	**5.12**	**6.20**	**7.92**	**4.53**	**4.85**	**10.90**

TABLE 5
Performance comparison between FedHAR and PerFedHAR on each unlabeled client under HAR-UCI dataset.

Model	Metric	Client															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
FedHAR	Accuracy	79.65	90.62	80.83	76.39	90.97	61.25	90.28	62.85	87.50	82.29	80.49	83.47	74.44	89.17	96.67	90.28
	F1-Score	76.80	90.50	78.93	73.85	90.76	57.56	89.75	61.80	86.80	80.36	78.15	82.55	69.29	89.03	96.67	89.95
PerFedHAR	Accuracy	96.04	96.32	91.25	83.82	98.75	79.03	97.71	73.06	98.33	95.97	89.44	89.65	92.71	97.29	98.33	99.86
	F1-Score	96.04	96.28	91.08	83.54	98.75	79.03	97.71	69.08	98.33	95.98	89.62	89.32	92.65	97.30	98.34	99.86
$\Delta(\%)\uparrow$	Accuracy	**16.39**	**5.70**	**10.42**	**7.43**	**7.78**	**17.78**	**7.43**	**10.21**	**10.83**	**13.68**	**8.95**	**6.18**	**18.27**	**8.12**	**1.66**	**9.58**
	F1-Score	**19.24**	**5.78**	**12.15**	**9.69**	**7.99**	**21.47**	**7.96**	**7.28**	**11.53**	**15.62**	**11.47**	**6.77**	**23.36**	**8.27**	**1.67**	**9.91**

Model	Metric	Client														
		17	18	19	20	21	22	23	24	25	26	27	28	29	30	Average
FedHAR	Accuracy	79.86	79.72	68.61	75.07	77.08	78.82	85.76	87.22	88.12	87.92	83.19	73.33	84.31	96.32	82.08
	F1-Score	77.40	78.59	68.00	74.82	76.41	77.59	85.24	86.63	86.77	87.16	82.65	72.97	83.65	96.32	80.90
PerFedHAR	Accuracy	84.10	91.46	71.94	91.60	95.00	87.85	94.79	99.51	100.00	91.67	99.79	95.56	99.58	100.00	92.68
	F1-Score	82.00	91.41	68.89	91.47	95.01	87.81	94.74	99.51	100.00	91.12	99.79	95.50	99.58	100.00	92.32
$\Delta(\%)\uparrow$	Accuracy	**4.24**	**11.74**	**3.33**	**16.53**	**17.92**	**9.03**	**9.03**	**12.29**	**11.88**	**3.75**	**16.60**	**22.23**	**15.27**	**3.68**	**10.60**
	F1-Score	**4.60**	**12.82**	**0.89**	**16.65**	**18.60**	**10.22**	**9.50**	**12.88**	**13.23**	**3.96**	**17.14**	**22.53**	**15.93**	**3.68**	**11.43**

equals to number of activities.

5.3 Baselines

We compare FedHAR with other widely used algorithms, including:

Pseudo Label [37] For unlabeled data sample, a pseudo label vector would be predicted, and the confidence is computed via dividing the maximum value by the summation. If the confidence is larger than a threshold, the pseudo label can be used as ground truth to calculate the gradients.

Mean Teacher [38] A widely used semi-supervised learning algorithm keeps a teacher model and a student model with the same structure. Within labeled data clients, the gradient is calculated by the weighted summation of the supervised loss and the knowledge distillation loss. For unlabeled data clients, only distillation loss is used to calculate the gradient.

ICT [39] The interpolation consistency training algorithm is also adopted for semi-supervised learning. Similarly, a teacher model and a student model are used for calculating the loss under the assumption that the prediction at an interpolation of unlabeled samples should be consistent with the interpolation of the predictions at those samples.

DeepSense [7]A widely used architecture based on CNN and RNN to capture relations among different sensors on HAR tasks. In each round, only those labeled data clients participate in training the framework.

HAR$_{sup}$ In each training round, only the labeled data is used to our proposed FedHAR architecture without the unlabeled clients, meaning it is a supervised training only process.

FedHAR$_{w/o\ agg}$ We use our FedHAR algorithm for training but set $agg = 1$ instead of using multiple aggregations. It is to observe the influence caused by the problem of convergence instability and concept drift in online semi-supervised learning.

FedAvg [16] One widely used aggregation algorithm in federated learning. Comparing with the proposed aggregation algorithm shown in Alg. 1 and Alg. 2, FedAvg uploaded the local model parameters to the server instead of the gradients. Except the aggregation algorithm, the other settings are all the same as *FedHAR*.

FedProx [40] A variation of *FedAvg* algorithm, in which a new hyper-parameter μ was introduced to control the difference of model parameters between the before and after the updating in each round.

HAR$_{sup}$, *Pseudo-label*, *Mean Teacher*, *ICT*, *FedHAR-agg*$_{w/o\ agg}$ all use the same model architecture with *FedHAR*. About *Pseudo-label*, we set the threshold to make pseudo labeling for one sample as 0.9. About *Mean Teacher*, we set the weight of semi-supervised loss as 0.1. Coefficient of exponential moving average (EMA) is set as 0 if number of rounds is less than 200, as 0.999 if umber of rounds is larger than 1000, otherwise as 0.99. About *ICT*, we set the weight of semi-supervised loss as 0.1. The setting of EMA is same with *Mean Teacher*. About *DeepSense*, in *RealWorld* dataset, we set the length of interval in one time window ($\tau = 2.5s$) as 0.5s and channel size as 32 for all CNN layers. For the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3136853, IEEE Transactions on Mobile Computing

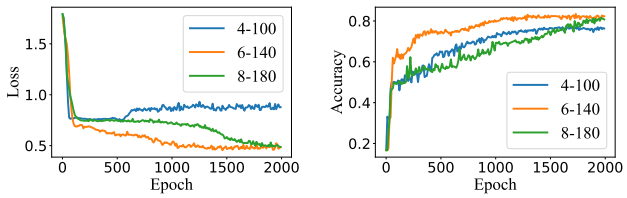
data in a time interval in each sensor, it first pass a 2-D CNN layer with kernel=(3,5) and stride=(1,1), two 1-D CNN layers with kernel=5 and stride=3 in turn. Then, generated features of each sensor are stacked and then pass a 2-D CNN layer with kernel=(9,5) and stride=(1,1), a 1-D CNN layer with kernel=8 and stride=5, a 1-D CNN layer with kernel=6 and stride=3. Next, generated feature concatenate with time window size τ and pass two GRU layers with output shape 128 and 64 respectively. Finally, labels are predicted based on a linear output layer. For *DeepSense* in *HAR-UCI* dataset, the channel size is set as 16 for all CNN layers. The generated features of each sensor are stacked and then pass a 2-D CNN layer with kernel=(2,5) and stride=(1,1). The shape of GRU layer is set as 64 and 16 respectively. Otherwise the same with the setting in *RealWorld* dataset.

5.4 Numerical results

5.4.1 General FedHAR

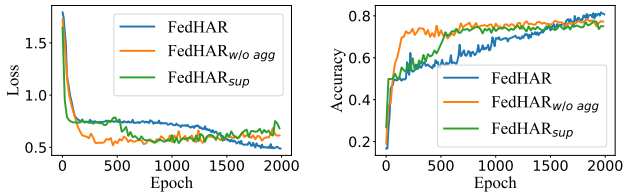
Table 2 and Table 3 denote the experimental results for general FedHAR algorithm and all the other baselines on **RealWorld** dataset and **HAR-UCI** dataset respectively. Particularly, we have few following observations:

- The proposed *FedHAR* can achieve the best prediction performance under each setting on both two datasets.
- Comparing FedHAR with the semi-supervised baselines, the effectiveness of the proposed semi-supervised algorithm in FedHAR can be proved. Among the semi-supervised baselines, *Mean Teacher* performs worst under two datasets. The dropout is used to constrain the predictions of the same samples to be consistent, which would aggravate over-fitting. About the *Pseudo Label* method, once the wrongly predicted label is used to train the model, the mistake would be accumulated and influence the model's performance. *ICT* is under the assumption that the probability that two randomly selected samples belong to different activities should be higher. However, in real-world online setting, activities in two adjacent time windows are more likely to be same. Therefore, the performance of *ICT* is relatively worse.
- The total number of labeled samples in Table 2 is much larger than that in Table 3, however, the overall performance is worse. This is because that the clients in **RealWorld** dataset are much more heterogeneous and the activity patterns are more diversified.
- FedHAR performs better than both FedAvg and FedProx, which demonstrates the effectiveness of the proposed gradients aggregation strategy. This is because in both FedAvg and FedProx, each client needs to update the parameters for several epochs using local data, and then uploads the updated parameters to the server for aggregation, which would lead to concept drift when facing real-time online data streams. Furthermore, FedProx could perform better than FedAvg. This is due to the fact that FedProx introduces one hyper-parameter μ to constrain the change of model parameters in each round, which would help the model keep learned knowledge in online setting.



(a) Loss of FedHAR under different experimental settings. (b) Accuracy of FedHAR under different experimental settings.

Fig. 3. Convergence comparison of the proposed FedHAR under different experimental settings on HAR-UCI dataset.



(a) Loss of different methods. (b) Accuracy of different methods.

Fig. 4. Convergence comparison between FedHAR with other methods under setting *8-180* on HAR-UCI dataset.

5.4.2 Ablation Study

Benefits of online setting. Comparing *FedHAR* with *FedHAR*$_{w/o\ agg}$, we can find that *FedHAR* can perform obviously better (e.g. *3-4000, 5-5000* in Table 2 and *4-250, 6-250* in Table 3), or their performance are quite close (e.g. *3-3000, 4-3000* in Table 2 and *4-350, 4-450* in Table 3). First, averaging gradients in multiple communications can help solve the problem of unstable convergence instability and concept drift in online learning. Furthermore, although gradient in one communication usually involve one action, gradients from a large of unlabeled data clients may involve all activity and sampling one labeled data batch is set to class-balance in practice, which can prevent concept drift to a certain extent.

Benefits of semi-supervised setting. The proposed *FedHAR* can perform better compared with two supervised baselines *HAR*$_{sup}$ and *DeepSense*. It is due to the fact that we leverage unlabeled data samples from all the unlabeled clients to improve the performance jointly. Because of the heterogeneity of different individuals' activity patterns and the small number of labeled clients, the two trained baselines have poor generalization on unlabeled data clients. In addition, *DeepSense* performs better than *HAR*$_{sup}$, this is because of the complex CNN and RNN architecture with heavy number of parameters, which is not appropriate in online HAR setting.

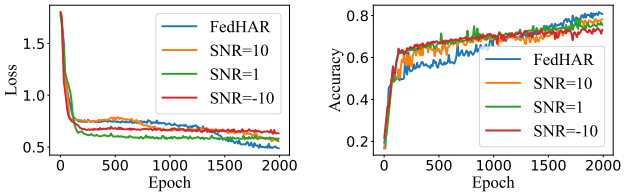
5.4.3 Personalized FedHAR

Table 4 denotes the performance results on each unlabeled client under **RealWorld** dataset. As shown in Table 5, we have some following observations: (1) In average, *PerFedHAR* all achieves over **+10%** improvement across all metrics on two datasets. It shows that personalizing based on the proposed semi-supervised learning under each unlabeled client can make a significant improvement based on general

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3136853, IEEE Transactions on Mobile Computing

TABLE 6
Total size of gradients/parameters transmitted in the training stage (GB). We use the amount of floating point operations (FLOPS) to measure the computational cost of models on edge devices (TFLOPS).

Dataset	RealWorld		UCI-HAR	
Communication costs (GB)	**Upload**	**Download**	**Upload**	**Download**
	32.59	48.99	15.46	27.36
Computational costs (TFLOPS)	**Labeled**	**Unlabeled**	**Labeled**	**Unlabeled**
	107.87	404.51	38.67	326.31



(a) Loss of FedHAR with different SNRs. (b) Accuracy of FedHAR with different SNRs.

Fig. 5. Performance comparisons of the proposed FedHAR with different SNRs(*10, 1, -10*) under setting *8-180* on HAR-UCI dataset.

FedHAR. (2) The performance improvements of some unlabeled clients are much more significant, for instance, the performance of *client 1, 3, 8, 10* improves about **+20%** on Accuacy and F1-Score in Table 4, the performance of *client 1, 6, 13, 20, 21* improves about **+15%** on two metrics as shown in Table 5. It is due to fact that only using personal data to fine-tune each unlabeled client can eliminate the heterogeneity of clients. The learned model can adapt to the personalized activity patterns well. (3) However, the performance of some unlabeled clients can only achieve limited improvement, for example, *client 2, 9* on **RealWorld** dataset and *client 15, 19* on dataset. This is because the activity patterns in the above unlabeled clients are quite different from those of the other clients. The initialized model based on general FedHAR does not learning enough knowledge on activity patterns of the above unlabeled clients.

5.4.4 Convergence analysis

We show the convergence time of proposed *FedHAR* under different experimental settings (e.g., *4-100, 6-140, 8-180*) on **HAR-UCI** dataset in Fig. 3. As shown in the result, with the increase of the number of labeled clients and labeled samples, *FedHAR* required more time to converge. We also compared the convergence time of *FedHAR* with other baselines ($FedHAR_{w/o\ agg}$ and HAR_{sup}) under experiment setting *8-180* on **HAR-UCI** dataset in Fig. 4. As depicted in the figures, HAR_{sup} converges fastest, while *FedHAR* needs most time. This is because HAR_{sup} just utilized the limited number of labeled samples for training. However, based on the real-time online sensing data without labels, *FedHAR* could update the model continuously to improve performance.

5.4.5 Communication cost and computation cost analysis

We conduct experiments under setting *5-2000* on **RealWorld** dataset and setting *8-180* on **HAR-UCI** dataset respectively to demonstrate the computation cost and the communication cost of the proposed *FedHAR*, whose results are shown in Table 6. The amount of floating point operations (FLOPS) is used to measure the computational cost of models on edge devices. We also show the total size of gradients transmitted in the training stage (communication costs).

5.4.6 Communication error analysis

According to literature [41], communication errors could prevent federated learning systems converge. Thus we simulate communication errors through assuming gradients would be transmitted under different signal noise ratios (SNRs). We conduct experiments to compare the proposed *FedHAR* with different SNRs under setting *8-180* on **HAR-UCI** dataset. As shown in Fig. 5, with the decrease of SNR, the performance becomes worse and worse due to the increasing noises. However, the performance of *FedHAR* can still keep in a relatively high level, which demonstrate the robustness of our proposed method when facing with communication errors.

5.4.7 Security analysis

The IoT devices in federated learning systems are vulnerable to attacks from malicious clients, thus robust federated learning has drawn more and more researchers' attentions [42] [43]. In our work, we leverage one Byzantine-robust aggregation rule [44] for defenses against malicious clients and introduce differential privacy (DP) for preventing data recovery attacks.

In order to demonstrate the damages of malicious clients and the effectiveness of the defenses like Median aggregation methods, we conduct experiments with *2, 4, 6, 8, 10* malicious clients under setting *8-180* on **HAR-UCI** dataset. The uploaded gradients from each malicious client were generated by standard Gaussian distribution. As shown in Table 7 with the increase of the number of malicious clients, the performance of *FedHAR* drops sharply. However, with Median defenses, the performance of *FedHAR* could be improved. For instance, when there are *4* malicious clients, the accuracy of *FedHAR* drops *26.18%*. In comparison, *FedHAR* with Median defenses achieve significant improvement up to *75.66%*.

As the differential privacy (DP) has been widely used for preventing data recovery attacks [45] [46], we conduct extensive experiments to test the performance of proposed *FedHAR* under the DP protected gradients. Similar with literature [47], we add Laplace noise to the gradients of each client. The experimental results under different settings on both **RealWorld** dataset and **HAR-UCI** dataset are shown in Table 8, 9. As shown in the results, both accuracy and F1-score of HAR_{DP} drops comparing with *FedHAR* due to the added noises. However, the performance of HAR_{DP} can still keep in a relatively high level, which is acceptable in real-world scenarios.

In summary, in the proposed *FedHAR* framework, we tried defenses against malicious clients and data recovery attacks using Median methods and DP respectively, which demonstrate the effectiveness. However, robust federated learning is an important yet challenging topic, which is worth our more efforts.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3136853, IEEE Transactions on Mobile Computing

TABLE 7
Numeric results of FedHAR, FedHAR$_{mal}$ (number of malicious clients are *2, 4, 6, 8, 10*), and FedHAR$_{med}$ with Median defenses.

	FedHAR	**FedHAR$_{mal}$**					**FedHAR$_{med}$**				
	-	2	4	6	8	10	2	4	6	8	10
Accuracy	80.74	55.35	54.56	54.49	54.85	37.88	77.84	75.66	73.00	67.70	65.64
F1-Score	79.34	47.04	43.69	43.80	47.57	24.30	75.78	73.45	69.99	65.31	62.16

TABLE 8
Performance comparison between FedHAR and FedHAR$_{DP}$ on RealWorld dataset. $*-*$ denotes the number of clients with labeled data $-$ the total sensing time (seconds) of each client.

	Metirc	**3-1200**	**3-1600**	**3-2000**	**4-1200**	**4-1600**	**4-2000**	**5-1200**	**5-1600**	**5-2000**
FedHAR$_{DP}$	Accuracy	53.01	60.03	55.19	60.37	60.68	62.10	60.31	59.64	59.77
	F1-Score	50.07	56.14	49.19	55.93	57.26	58.91	59.28	57.26	57.00
FedHAR	Accuracy	55.33	63.67	56.02	63.47	61.63	65.31	65.10	64.11	62.95
	F1-Score	51.80	60.48	50.81	59.39	57.75	62.77	62.44	61.10	57.70
$\mathbf{\Delta}(\%)\downarrow$	Accuracy	2.32	3.64	0.83	3.10	0.95	3.21	4.79	4.47	3.18
	F1-Score	1.73	4.34	1.62	3.46	0.49	3.86	3.16	3.84	0.70

TABLE 9
Performance comparison between FedHAR and FedHAR-DP on HAR-UCI dataset. $*-*$ denotes the number of clients with labeled data $-$ the total sensing time (seconds) of each client.

	Metric	**4-100**	**4-140**	**4-180**	**6-100**	**6-140**	**6-180**	**8-100**	**8-140**	**8-180**
FedHAR$_{DP}$	Accuracy	72.31	72.41	76.78	76.56	77.77	76.97	82.02	75.26	77.54
	F1-Score	70.59	69.87	74.82	74.02	73.08	74.63	80.70	72.27	76.19
FedHAR	Accuracy	76.32	74.32	78.68	81.51	82.33	77.21	82.61	76.48	80.74
	F1-Score	74.69	71.98	76.59	79.97	81.28	75.27	81.62	74.78	79.34
$\mathbf{\Delta}(\%)\downarrow$	Accuracy	4.01	1.91	1.90	4.95	4.56	0.24	0.59	1.22	3.20
	F1-Score	4.10	2.11	1.77	5.95	8.20	0.64	0.92	2.51	3.15

5.4.8 Hyper parameters analysis

To investigate the influence of different hyper parameters that effect the performance of *FedHAR*, we conduct experiments on following parameter variants: weight of semi-supervised loss, number of aggregation rounds of semi-supervised gradients, number of randomly selected unlabeled clients in one round, the choosing of the loss function.

Can the model rely principally on the unsupervised learning? We set the weight of unsupervised loss λ in Alg. 1 from 0 to 0.9 to investigate the performance change. As shown in Fig. 6(a), with the increase of λ, model performance gradually improves and achieves the best when λ reaches 0.2; then the performance drops rapidly as the unsupervised gradient descent dominates the whole training. We argue that supervised learning is indispensable as it can provide a stable global-view across the whole training phase.

How do aggregation rounds affect model performance? We set $agg = 1, 10, 20, 40, 60$ in Alg. 1, and the results are shown in Fig. 6(b). As depicted in the figure, with the increase of **agg**, both accuracy and F1-Score increase first and then decrease. As we mentioned in Sec. 4.1, small values of agg would cause unstable convergence and concept drift. However, too large values of agg mean that too many gradients would be collected from clients in one round, which would introduce more noises and mislead the training process.

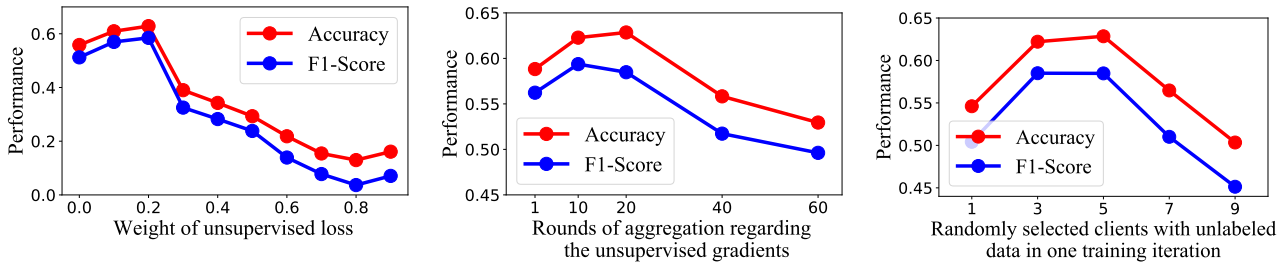
How many unlabeled clients should be included in one iteration? We set $M_b = 1, 3, 5, 7, 9$ in Alg. 1, and the results are shown in Fig. 6(c). Similarly, both accuracy and F1-Score increase first and then decreases with the increase of M_b. If only a few unlabeled clients are sampled in one round, differences in activity patterns will make gradient descent unstable. Each client would generate gradients in different directions or even opposite directions caused by entirely different activity patterns in the same round. Averaging too many gradients would lose much information and influence the performance.

Why to choose MSE as the consistency training loss function? We choose the mean-square error (MSE) as the divergence distance metric empirically, which is based on extensive experiments. As shown in Fig. 7, the model can achieve superior performance improvement compared with other measures, such as KL-divergence or cosine-similarity, with the increase of epochs. This might be due to the fact that the gradients calculated based on the MSE loss could be relatively smaller than other baselines, which could update the model continuously when facing with real-time online sensing data. However, how to proof this theoretically would be another challenge in the future work.

6 Conclusion

Human activities recognition (HAR) based on multi-modality sensor data was an important yet challenging

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3136853, IEEE Transactions on Mobile Computing

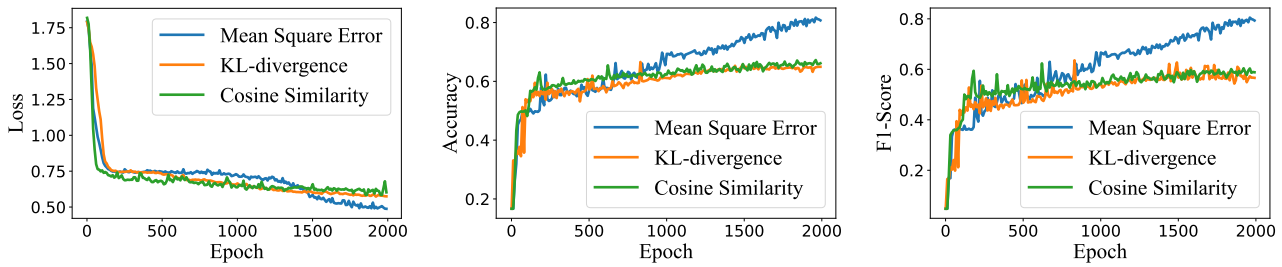


(a) Model performance against the weight of unsupervised loss.

(b) Model performance against the total rounds of aggregation regarding the unsupervised learning.

(c) Model performance against the number of randomly selected clients in one iteration.

Fig. 6. Parameter analysis on **RealWorld** dataset with $N = 5$, $M = 10$ and total sensing time of 5000 seconds.



(a) Loss under different metrics.

(b) Accuracy under different metrics.

(c) F1-Score under different metrics.

Fig. 7. Performance comparisons about using different metrics as the consistency training loss function under setting *8-180* on HAR-UCI dataset.

task. To overcome the *privacy preservation*, *label scarce*, *real-time*, and *heterogeneity* challenges, we proposed *FedHAR*, a personalized federated HAR framework based on semi-supervised online learning. *FedHAR* first introduced hierarchical attention architecture for the alignment of different level features. Then a semi-supervised online learning strategy was proposed for online HAR tasks, including a novel algorithm for computing unsupervised gradients under the *consistency training* proposition, an *unsupervised gradients aggregation* strategy to overcome the *concept drift* and *convergence instability* problem in online learning, and a semi-supervised learning loss to aggregate gradients from all the labels clients and unlabeled clients. As demonstrated in the experiments, the proposed *FedHAR* outperformed the state-of-the-art baselines on two public datasets. Moreover, when fine-tuning each unlabeled client, *PerFedHAR* can achieve about **+10%** improvement across all metrics on two datasets on average.

REFERENCES

[1] Y. Guan and T. Plötz, "Ensembles of deep lstm learners for activity recognition using wearables," *IMWUT*, vol. 1, no. 2, p. 11, 2017.

[2] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[3] M. Kose, O. D. Incel, and C. Ersoy, "Online human activity recognition on smart phones," in *Workshop on mobile sensing: from smartphones and wearables to big data*, vol. 16, no. 2012, 2012, pp. 11–15.

[4] T. Kautz, B. H. Groh, J. Hannink, U. Jensen, H. Strubberg, and B. M. Eskofier, "Activity recognition in beach volleyball using a deep convolutional neural network," *Data Mining and Knowledge Discovery*, vol. 31, no. 6, pp. 1678–1705, 2017.

[5] M. Inoue, S. Inoue, and T. Nishida, "Deep recurrent neural network for mobile human activity recognition with high throughput," *Artificial Life and Robotics*, vol. 23, no. 2, pp. 173–185, 2018.

[6] M. S. Singh, V. Pondenkandath, B. Zhou, P. Lukowicz, and M. Liwickit, "Transforming sensor data to the image domain for deep learning¡a an application to footstep detection," in *IJCNN*. IEEE, 2017, pp. 2665–2672.

[7] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *WWW*. International World Wide Web Conferences Steering Committee, 2017, pp. 351–360.

[8] H. Ma, W. Li, X. Zhang, S. Gao, and S. Lu, "Attnsense: multi-level attention mechanism for multimodal human activity recognition," in *IJCAI*. AAAI Press, 2019, pp. 3109–3115.

[9] K. Sozinov, V. Vlassov, and S. Girdzijauskas, "Human activity recognition using federated learning," in *ISPA/IUCC/BDCloud/SocialCom/SustainCom*. IEEE, 2018, pp. 1103–1111.

[10] L. Yao, F. Nie, Q. Z. Sheng, T. Gu, X. Li, and S. Wang, "Learning from less for better: semi-supervised activity recognition via shared structure discovery," in *UbiComp*, 2016, pp. 13–24.

[11] M. Zeng, T. Yu, X. Wang, L. T. Nguyen, O. J. Mengshoel, and I. Lane, "Semi-supervised convolutional neural networks for human activity recognition," in *ICBD*. IEEE, 2017, pp. 522–529.

[12] Y. Zhao, H. Liu, H. Li, P. Barnaghi, and H. Haddadi, "Semi-supervised federated learning for activity recognition," *arXiv preprint arXiv:2011.00851*, 2020.

[13] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Applied Soft Computing*, vol. 62, pp. 915–922, 2018.

[14] G. Bhat, R. Deb, V. V. Chaurasia, H. Shill, and U. Y. Ogras, "Online human activity recognition using low-power wearable devices," in *ICCAD*. IEEE, 2018, pp. 1–8.

[15] T. Miu, P. Missier, and T. Plötz, "Bootstrapping personalised human activity recognition models using online active learning," in *CIT/IUCC/DASC/PICom*. IEEE, 2015, pp. 1138–1147.

[16] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[17] N. Hammerla, J. Fisher, P. Andras, L. Rochester, R. Walker, and T. Plötz, "Pd disease state assessment in naturalistic environments using deep learning," in *AAAI*, vol. 29, no. 1, 2015.

[18] A. Sathyanarayana, S. Joty, L. Fernandez-Luque, F. Ofli, J. Srivastava, A. Elmagarmid, S. Taheri, and T. Arora, "Impact of

physical activity on sleep: A deep learning based exploration," *arXiv preprint arXiv:1607.07034*, 2016.
[19] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *arXiv preprint arXiv:1604.08880*, 2016.
[20] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan, "Deep activity recognition models with triaxial accelerometers," *arXiv preprint arXiv:1511.04664*, 2015.
[21] N. D. Lane, P. Georgiev, and L. Qendro, "Deepear: robust smartphone audio sensing in unconstrained acoustic environments using deep learning," in *UbiComp*, 2015, pp. 283–294.
[22] T. Plötz, N. Y. Hammerla, and P. L. Olivier, "Feature learning for activity recognition in ubiquitous computing," in *IJCAI*, 2011.
[23] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.
[24] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.
[25] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Transactions on Mobile Computing*, 2020.
[26] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.
[27] Y. Jiang, J. Konečnỳ, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," *arXiv preprint arXiv:1909.12488*, 2019.
[28] J. Feng, C. Rong, F. Sun, D. Guo, and Y. Li, "Pmf: A privacy-preserving human mobility prediction framework via federated learning," *IMWUT*, vol. 4, no. 1, pp. 1–21, 2020.
[29] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.
[30] X. Sun, H. Kashima, and N. Ueda, "Large-scale personalized human activity recognition using online multitask learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2551–2563, 2012.
[31] Z. Chen, Q. Zhu, Y. C. Soh, and L. Zhang, "Robust human activity recognition using smartphone sensors via ct-pca and online svm," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3070–3080, 2017.
[32] T. Miu, P. Missier, and T. Plötz, "Bootstrapping personalised human activity recognition models using online active learning," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. IEEE, 2015, pp. 1138–1147.
[33] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *ICML*, ser. ICPS, A. P. Danyluk, L. Bottou, and M. L. Littman, Eds., vol. 382. ACM, 2009, pp. 41–48. [Online]. Available: https://doi.org/10.1145/1553374.1553380
[34] A. Fallah, A. Mokhtari, and A. E. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *NeurIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/24389bfe4fe2eba8bf9aa9203a44cdad-Abstract.html
[35] T. Sztyler and H. Stuckenschmidt, "On-body localization of wearable devices: An investigation of position-aware activity recognition," in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE Computer Society, 2016, pp. 1–9, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7456521.
[36] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones." in *Esann*, vol. 3, 2013, p. 3.
[37] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2, 2013.
[38] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *arXiv preprint arXiv:1703.01780*, 2017.
[39] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," *arXiv preprint arXiv:1903.03825*, 2019.
[40] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, 2018.
[41] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2020.
[42] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 118–128.
[43] J. Feng, H. Xu, and S. Mannor, "Distributed robust learning," *arXiv preprint arXiv:1409.5937*, 2014.
[44] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5650–5659.
[45] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.
[46] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy." *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.
[47] ——, "The algorithmic foundations of differential privacy." *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

Hongzheng Yu Hongzheng Yu is currently a master student in Computer Science at the Shandong University, China. He received his B.S. degree from Shandong University, China.

Zekai Chen Zekai Chen is currently a Ph.D. student in Computer Science at the George Washington University. His research interests include general machine learning, multi-task learning, sequence modeling and efficient computation in deep learning.

Xiao Zhang Xiao Zhang received his B.S. and Ph.D degree from Central South University and Nanjing University, China, respectively. He is now an assistant professor in the School of Computer Science and Technology, Shandong University. Dr. Zhang's research interests include data mining, intelligent sensing, multi-task learning and federated learning.

Chen.png Chen.png

Xu Chen (Senior Member, IEEE) received the Ph.D. degree in information engineering from the Chinese University of Hong Kong in 2012. He is a Full Professor with Sun Yat-sen University, Guangzhou, China, and the Vice Director of the National and Local Joint Engineering Laboratory of Digital Home Interactive Applications. He was a Post-Doctoral Research Associate with Arizona State University, Tempe, USA, from 2012 to 2014, and a Humboldt Scholar Fellow with the Institute of Computer Science, University of Goettingen, Germany, from 2014 to 2016. He was a recipient of the Prestigious Humboldt Research Fellowship awarded by Alexander von Humboldt Foundation of Germany, the 2014 Hong Kong Young Scientist Runner-Up Award, the 2017 IEEE Communication Society Asia–Pacific Outstanding Young Researcher Award, the 2017 IEEE ComSoc Young Professional Best Paper Award, the Honorable Mention Award of 2010 IEEE international conference on Intelligence and Security Informatics, the Best Paper Runner-Up Award of 2014 IEEE International Conference on Computer Communications (INFOCOM), and the Best Paper Award of 2017 IEEE International Conference on Communications. He is currently an Area Editor of IEEE Open Journal of the Communications Society, an Associate Editor of the IEEE Transactions Wireless Communications, IEEE Internet of Things Journal and IEEE Journal on Selected Areas in Communications (JSAC) Series on Network Softwarization and Enablers.

Fuzhen Zhuang Fuzhen Zhuang received the Ph.D. degrees in the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2011. He is currently a full Professor with the Institute of Artificial Intelligence, Beihang University. He has published more than 100 papers in some prestigious refereed journals and conference proceed-ings such as the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the ACM Transactions on Knowledge Discovery from Data, the ACM Transactions on Intelligent Systems and Technology, Information Sciences, Neural Networks, SIGKDD, IJCAI, AAAI, TheWebConf, SIGIR, ICDE, ACM CIKM, ACM WSDM, SIAM SDM, and IEEE ICDM. His research interests include transfer learning, machine learning, data mining, multitask learning, knowledge graph and recommendation systems. He is a Senior Member of CCF. He was a recipient of the Distinguished Dissertation Award of CAAI in 2013.

Hui Xiong Hui Xiong is currently a Full Professor at the Rutgers, the State University of New Jersey, where he received the 2018 Ram Charan Management Practice Award as the Grand Prix winner from the Harvard Business Review, RBS Dean's Research Professorship (2016), the Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence (2009), the ICDM Best Research Paper Award (2011), and the IEEE ICDM Outstanding Service Award (2017). He received the Ph.D. degree from the University of Minnesota (UMN), USA. He is a co-Editor-in-Chief of Encyclopedia of GIS, an Associate Editor of IEEE Transactions on Big Data (TBD), ACM Transactions on Knowledge Discovery from Data (TKDD), and ACM Transactions on Management Information Systems (TMIS). He has served regularly on the organization and program committees of numerous conferences, including as a Program Co-Chair of the Industrial and Government Track for the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), a Program Co-Chair for the IEEE 2013 International Conference on Data Mining (ICDM), a General Co-Chair for the IEEE 2015 International Conference on Data Mining (ICDM), and a Program Co-Chair of the Research Track for the 2018 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. He is an IEEE Fellow and an ACM Distinguished Scientist.

Xiuzhen Cheng Xiuzhen Cheng received her MS and PhD degrees in computer science from the University of Minnesota – Twin Cities, in 2000 and 2002, respectively. She is a professor at the Department of Computer Science, The George Washington University, Washington DC, and a professor in School of Computer Science and Technology, Shandong University. Her current research focuses on Blockchain computing, intelligent Internet of Things, wireless and mobile security, and algorithm design and analysis. She is the founder and steering committee chair of the International Conference on Wireless Algorithms, Systems, and Applications (WASA, launched in 2006). She served/is serving on the editorial boards of several technical journals (e.g. IEEE Transactions on Computers, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Wireless Communications, IEEE Wireless Communications Magazine) and the technical program committees of many professional conferences/workshops (e.g. ACM Mobihoc, ACM Mobisys, IEEE INFOCOM, IEEE ICDCS, IEEE ICC, IEEE/ACM IWQoS). She also chaired several international conferences (e.g. ACM Mobihoc'14, IEEE PAC'18). Xiuzhen worked as a program director for the US National Science Foundation (NSF) from April to October in 2006 (full time), and from April 2008 to May 2010 (part time). She published more than 200 peer-reviewed papers. She is a Fellow of IEEE.