Personal and Ubiquitous Computing (2022) 26:1281–1298
https://doi.org/10.1007/s00779-022-01688-8

ORIGINAL ARTICLE

Semi-supervised and personalized federated activity recognition based on active learning and label propagation

Riccardo Presotto[1] · Gabriele Civitarese[1] · Claudio Bettini[1]

Received: 15 September 2021 / Accepted: 30 May 2022 / Published online: 23 June 2022

Abstract

One of the major open problems in sensor-based Human Activity Recognition (HAR) is the scarcity of labeled data. Among the many solutions to address this challenge, semi-supervised learning approaches represent a promising direction. However, their centralized architecture incurs in the scalability and privacy problems that arise when the process involves a large number of users. Federated learning (FL) is a promising paradigm to address these problems. However, the FL methods that have been proposed for HAR assume that the participating users can always obtain labels to train their local models (i.e., they assume a fully supervised setting). In this work, we propose FedAR: a novel hybrid method for HAR that combines semi-supervised and federated learning to take advantage of the strengths of both approaches. FedAR combines active learning and label propagation to semi-automatically annotate the local streams of unlabeled sensor data, and it relies on FL to build a global activity model in a scalable and privacy-aware fashion. FedAR also includes a transfer learning strategy to fine-tune the global model on each user. We evaluated our method on two public datasets, showing that FedAR reaches recognition rates and personalization capabilities similar to state-of-the-art FL supervised approaches. As a major advantage, FedAR only requires a very limited number of annotated data to populate a pre-trained model and a small number of active learning questions that quickly decrease while using the system, leading to an effective and scalable solution for the data scarcity problem of HAR.

Keywords Activity recognition · Federated learning · Semi-supervised learning · Mobile computing

1 Introduction

The majority of approaches for Human Activity Recognition (HAR) based on data continuously acquired from mobile and wearable devices rely on supervised learning methods [1].

While supervised learning leads to high recognition rates, collecting a sufficiently representative amount of labeled data to train the recognition model is often a real challenge [2]. For instance, data annotation can be performed directly by the monitored subject while performing activities (self-annotation). However, this approach is very obtrusive and error-prone. Alternatively, external observers can annotate the activity execution of a subject (in real-time or by semi-automatic video annotation), but these methods are time-consuming and privacy-intrusive.

Among the solutions that have been proposed to tackle the labeled data scarcity problem, semi-supervised learning represents a promising research direction that has been explored in the last few years [3]. Semi-supervised methods only use a small amount of labeled data to initialize the recognition model, which is continuously updated taking advantage of unlabeled data. However, there are still several challenges that limit the deployment of these methods in realistic scenarios. Indeed, even though semi-supervised approaches mitigate the data scarcity problem, they do not consider the scalability and privacy issues that arise in training a real-world recognition model that includes data from a large number of different users. From the scalability point of view, the computational effort that is required to train a global model significantly grows as the number of users increases. Considering privacy aspects, activity data may

✉ Gabriele Civitarese
gabriele.civitarese@unimi.it

Riccardo Presotto
riccardo.presotto@unimi.it

Claudio Bettini
claudio.bettini@unimi.it

[1] Dept. of Computer Science, University of Milan, Milan, Italy

reveal sensitive information, like the daily behavior of a subject and her habits [4]. Accurate HAR also requires a certain amount of personalization on the end users [5].

In 2016, Google introduced the federated learning (FL) framework [6]. In the FL paradigm, the model training task is distributed over a multitude of nodes (e.g., mobile devices). Each node uses its own labeled data to train a local model. The resulting model parameters of each participating node are forwarded to a server that is in charge of aggregating them. Finally, the server shares the aggregated parameters to the participating nodes. FL is a promising direction to make activity recognition scalable for a large number of users. Moreover, FL mitigates the privacy problem since only model parameters, and not actual data, are shared with the server, and privacy-preserving mechanisms (e.g., Secure MultiParty Computation, Differential Privacy) are used when aggregating parameters [7].

FL has been recently applied to HAR showing that it can reach an accuracy very close to centralized methods [8]. However, all existing solutions assume that each node has complete availability of labeled sensor data. This is actually the general setting of existing works based on FL, which in the literature has been primarily considered for fully supervised learning tasks [9]. While this assumption may be valid for some applications (e.g., the Google approach for keyboard suggestions improvement relies on labeled data implicitly provided by users when typing or confirming suggestions [10]), it is not realistic for applications like HAR where labeled data availability is significantly limited. Extending FL to semi-supervised learning is one of the open challenges in this area [9].

In this work, we propose FedAR: a hybrid semi-supervised and FL framework that enables personalized privacy-aware and scalable HAR based on mobile and wearable devices. Different from the majority of the existing solutions, FedAR considers a limited availability of labeled data. In particular, FedAR combines active learning and label propagation to provide labels to a large amount of unlabeled data. Newly labeled data are periodically used by each node to perform local training, thus obtaining the model parameters that are then transmitted to the server that aggregates them using Secure Multiparty Computation. FedAR also relies on transfer learning to fine-tune the global model for each user, while generating a global model that generalizes over unseen users.

Considering the limitations of existing evaluation methodologies for FL applied to HAR [11], we designed a novel evaluation methodology to robustly assess both the generalization and the personalization capabilities of our approach. The results of our experimental evaluation on two publicly available datasets show that FedAR reaches recognition rates close to state-of-the-art solutions that assume the complete availability of labeled data. Moreover, both the generalization and the personalization capabilities of FedAR keep increasing over time. Last but not least, the amount of triggered active learning questions is small and acceptable for a real-world deployment.

To the best of our knowledge, FedAR is the first FL framework for HAR that tackles the data scarcity problem while considering personalization. Hence, we believe that FedAR is a significant step towards realistic deployments of HAR systems based on FL.

In summary, the contributions of this work are the following:

- We present FedAR, a novel hybrid approach that combines federated, semi-supervised, and transfer learning to tackle the data scarcity problem for real-world personalized HAR.
- We propose a novel strategy to reliably evaluate the evolution of the personalization and generalization capabilities of FedAR over time.
- An extensive evaluation on public datasets shows that FedAR reaches similar recognition rates with respect to well-known approaches that assume high availability of labeled data. At the same time, FedAR triggers a small number of active learning questions that quickly decreases while using the system.

2 Related work

2.1 Labeled data scarcity in HAR

Considering HAR based on data collected from mobile devices' inertial sensors, the majority of approaches rely on supervised machine learning [12–16]. However, these approaches need a significant amount of labeled data to train the classifier. Indeed, different users may perform the same activities in very different ways, but also distinct activities may be associated with similar motion patterns. The annotation task is costly, time-consuming, intrusive, and hence prohibitive on a large scale [2]. In the following, we summarize the main methodologies that have been proposed in the literature to mitigate this problem.

Unsupervised approaches have been proposed to derive activity clusters from unlabeled sensor data [17]. Those approaches still need annotations to reliably associate an activity label to each cluster. Since distinct human activities often share similar sensor patterns, purely unsupervised data-driven approaches for activity recognition are still a challenge considering real-world scenarios.

Some research efforts focused on knowledge-based approaches based on logical formalisms, especially targeting smart-home environments [18, 19]. These approaches usually rely on ontologies to represent the common-sense relationships between activities and sensed data. One of the main issues of knowledge-based approaches is their inadequacy to model the intrinsic uncertainty of sensor-based systems and the large variety of activity execution modalities.

Data augmentation is a more popular solution adopted in the literature to mitigate the data scarcity problem, especially considering imbalanced datasets [20, 21]. In these approaches, the available labeled data are slightly perturbed to generate new labeled samples. With respect to our method, data augmentation is an orthogonal approach that could be integrated to further increase the amount of labeled data. Recently, data augmentation in HAR has also been tackled taking advantage of GAN models to generate synthetic data more realistic than the ones obtained by the above-mentioned approaches [22, 23]. However, GANs require to be trained with a significant amount of data.

Many transfer learning approaches have been applied to HAR to fine-tune models learned from a source domain with available labeled data to a target domain with low-availability of labeled data [24–27]. FedAR relies on transfer learning to fine-tune the personal local model taking advantage of the global model trained by all the participating devices.

An effective method to tackle data scarcity for HAR when the feature space is homogeneous (like in FedAR) is semi-supervised learning [3, 28–30]. Semi-supervised methods only use a restricted labeled dataset to initialize the activity model. Then, a significant amount of unlabeled data is semi-automatically annotated. The most common semi-supervised approaches for HAR are self-learning [30], label propagation [31], co-learning [32], and active learning [33–35]. Active learning has also been adopted in HAR to handle the class imbalance problem [36]. Hybrid solutions based on semi-supervised learning and knowledge-based reasoning have been proposed in [37]. Existing semi-supervised solutions do not consider the scalability problems related to building a recognition model with a large number of users for real-world deployments. Moreover, the data required to build such collaborative models is sensitive, as it could reveal private information about the users (e.g., user health condition and habits) [4, 38, 39].

2.2 Federated learning for HAR

Recently, the FL paradigm has been proposed to distribute model training over a multitude of nodes [6, 7, 9, 40–42]. A recent survey divides FL methods in three categories: horizontal, vertical, and transfer FL [7]. FedAR is a horizontal FL method: the participating mobile devices share the same feature space, but they have a different space in samples (i.e., each device considers data for a specific user). Among the required characteristics of FL approaches, the personalization of the global model on each client plays a major role [43]

FL has been previously applied to mobile/wearable HAR to distribute the training of the activity recognition model among the participating devices [8, 11, 44–48]. In this area, recent works also proposed to learn the global model in a decentralized fashion [49]. Existing works show that FL solutions for HAR reach recognition accuracy similar to standard centralized models [45]. Moreover, since personalization is an important aspect for HAR [5], existing works also show that applying transfer learning strategies to fine-tune the global model on each client leads to a significantly improved recognition rate [8, 46]. One of the major drawbacks of these solutions is that they assume high availability of labeled data, hence considering a fully supervised setting.

The combination of federated and active learning has been recently proposed for Intrusion Detection Systems [50]. However, semi-supervised federated learning solutions for HAR have been only partially explored. The existing works mainly focus on unsupervised methods to collaboratively learn (based on the FL setting) a robust feature representation from the unlabeled stream of sensor data. The global feature representation is then used to build activity classifiers using a limited amount of labeled data. For instance, the work in [47] proposes an approach based on autoencoders, while the work in [51] is based on self-supervised learning. However, those works do not consider model personalization and they do not propose approaches to continuously obtain new labeled data from each user. Nonetheless, we believe that those works focus on a very important orthogonal problem with respect to the one addressed by FedAR. Indeed, feature learning from unlabeled data could be integrated in FedAR to further reduce the amount of active learning questions and to improve the recognition rate. Recently, the work in [52] proposes a solution to build the global model by aggregating the local models' gradients from a small number of clients with labeled data and a large number of clients with unlabeled data. This method is based on a semi-supervised loss that relies on a novel unsupervised gradients aggregation strategy. Differently from this work, we do not assume the existence of clients with full availability of labeled data, and we also propose a practical solution to continuously improve the global model thanks to active learning and label propagation.

A common limitation in the literature is the methodology adopted to evaluate FL for HAR applications [8, 11, 48]. Indeed, none of the proposed methodologies truly assess the generality of the global model over users whose data have never been used for training. Moreover, only one iteration of the FL process is evaluated, while in a realistic deployment this process is repeated periodically with different data. In this work, we propose an evaluation methodology that overcomes the above-mentioned issues (see Section 5.2).

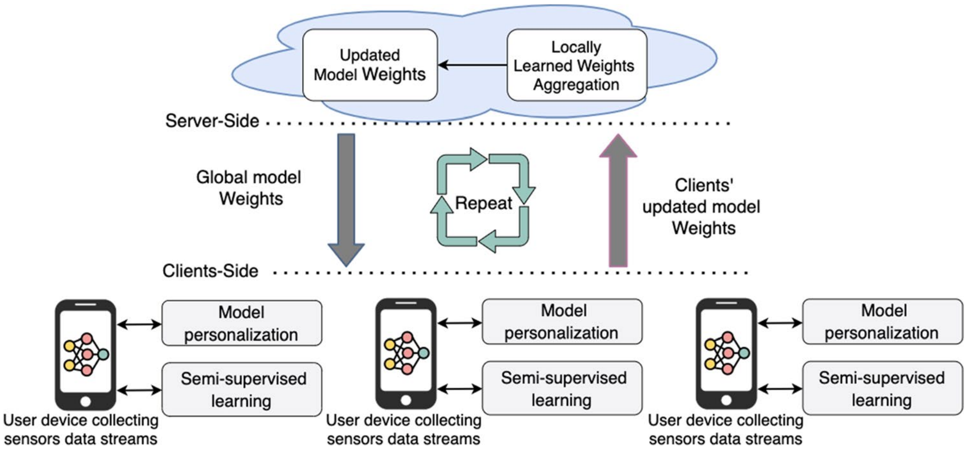


Fig. 1 Overall architecture of FedAR

3 Overview of FedAR

In this section, we describe the overall FedAR framework at a high-level.

3.1 Overall architecture

The overall architecture of FedAR is depicted in Fig. 1. For the sake of this work and without loss of generality, we illustrate FedAR applied to physical activity recognition based on inertial sensors data collected from personal mobile devices.

Following the FL paradigm, the actors of FedAR are a server and a set of clients that cooperate to periodically compute the weights of a global activity recognition model. In order to address the labeled data scarcity problem, FedAR initializes the global model in an offline phase with a limited amount of labeled data, while each client implements a semi-supervised learning strategy (i.e., a combination of active learning and label propagation) to semi-automatically label a portion of the unlabeled sensor data stream. An overview of our semi-supervised strategy is described in Section 3.3.

Periodically (e.g., every night), the server starts a process to update the weights of the global model. Each client uses its available labeled data to train its local model. The resulting local weights are transmitted to the server, which aggregates them with the ones from the other clients to obtain a new version of the global model. Finally, the new version of the global model weights is transmitted to each client. Since different users may perform activities in very different ways, a model personalization module on each client is in charge of fine-tuning the updated global model weights on the specific user. A more detailed overview on the global model update and personalization is described in Section 3.4.

3.2 Local models

One of the strengths of FedAR is that it is designed considering both personalization and generalization aspects. Personalization is crucial for the local models to recognize the activities of each user more accurately. On the other hand, generalization is a desirable property for the global model. Indeed, some participating users may not wish to collect labeled data (not even a small amount), or may have devices not adequate to perform local training. Those users are not able to actively contribute to the federated learning process, and their clients would directly use the last version of the global model for activity classification.

In order to guarantee both personalization and generalization, in FedAR, each client stores two distinct instances of the activity model. The former is called *Shareable Model*, and it is the one used for federated learning. In order to personalize the activity model on each user, a straightforward solution would be to fine-tune the *Shareable Model* with transfer learning approaches [53]. However, recent studies show that a global model built by aggregating the weights of fine-tuned models exhibits poor generalization capabilities on external users [11]. In order to overcome this problem, in FedAR, at the end of each global model update the clients that actively contribute to the federated learning process create a copy of the *Shareable Model* that is called *Personalized Model*. The *Personalized Model* is fine-tuned on the specific user and it is used for activity classification. Besides improving generalization, an advantage of keeping private the weights of the *Personalized Local Model* is a positive impact on privacy protection [54].

3.3 Semi-supervised data labeling and classification

Figure 2 depicts the semi-supervised data labeling and classification flow of FedAR.

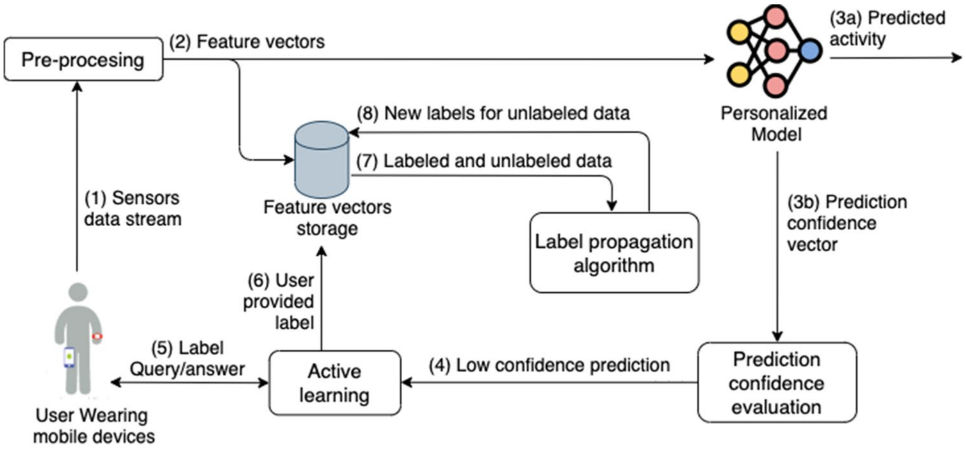

Pre-procesing
(2) Feature vectors
(3a) Predicted activity
Personalized Model
(8) New labels for unlabeled data
(7) Labeled and unlabeled data
(1) Sensors data stream
Feature vectors storage
Label propagation algorithm
(3b) Prediction confidence vector
(6) User provided label
(5) Label Query/answer
Active learning
(4) Low confidence prediction
Prediction confidence evaluation
User Wearing mobile devices

Fig. 2 Semi-supervised data labeling and classification data flow

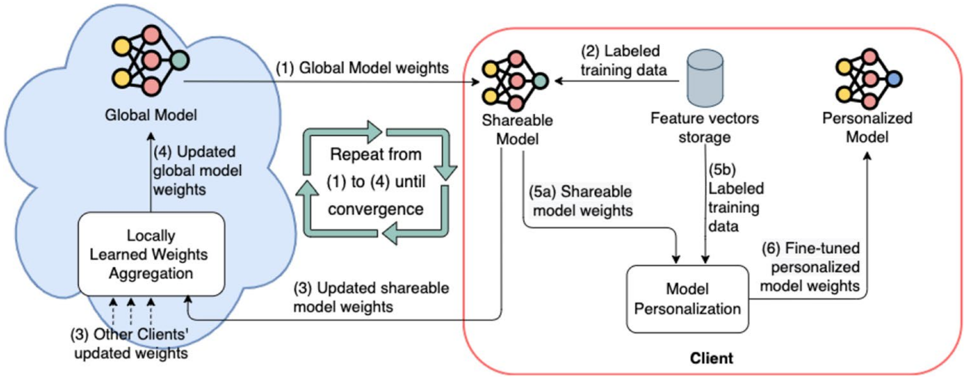

Global Model
(1) Global Model weights
(2) Labeled training data
Shareable Model
Feature vectors storage
Personalized Model
(4) Updated global model weights
Repeat from (1) to (4) until convergence
(5a) Shareable model weights
(5b) Labeled training data
Locally Learned Weights Aggregation
(6) Fine-tuned personalized model weights
(3) Updated shareable model weights
Model Personalization
(3) Other Clients' updated weights
Client

Fig. 3 Local models training and personalized model update

Each client in FedAR uses the *Personalized Model* to classify activities in real-time on the continuous stream of unlabeled pre-processed sensor data. Before classification, each unlabeled data sample is stored in the *Feature Vectors Storage*. This storage collects both unlabeled and labeled data samples. After classification, if the confidence on the current prediction is below a threshold, an active learning process is started, and the system asks the user about the activity that she was actually performing. The feedback from the user is then associated with the corresponding feature vector in the *Feature Vectors Storage*. Active learning makes it possible to assign a label to those informative data points that can effectively improve the local model. For the sake of usability, the number of active learning queries should be low, since they may bother the user during activity execution. For this reason, FedAR also periodically applies a Label Propagation algorithm to spread the labels acquired through active learning to a larger number of unlabeled data points. The advantage of label propagation is to further improve the recognition rate by training the classifier with a significant amount of labeled data samples and, at the same time, to reduce the number of needed active leaning queries over time.

3.4 Global model update and personalizazion

Periodically (e.g., every night) the server asks to the participating clients to update the global model. This process is depicted in Fig. 3.

First, each client replaces its *Shareable Model* with the current version of the *Global Model*. Then, the labeled data in the *Feature Vectors Storage* are used to perform local training of the *Shareable Model*. After training, the updated *Shareable Model* weights are then forwarded to the server, which is in charge of aggregating the weights from all the clients to generate a new version of the global model. These steps are repeated until convergence of the global model. At the end of this process, the *Shareable Model* of each client is replaced with the last stable version of the *Global Model*.

Then, the *Model Personalization* module generates a copy of the *Shareable Model* that is called *Personalized Model*, which is fine-tuned using the *Feature Vectors Storage*. The result of this process is a *Personalized Model* that takes advantage of the high-level features of the *Global Model* as well as the personalized aspects of the specific user.

4 FedAR under the hood

In this section, we describe in detail the algorithms of FedAR.

4.1 The activity model

Since we consider a setting with limited availability of labeled data, activity models that automatically learn features from raw data are not effective in FedAR. Indeed, based on our experiments that we describe in Section 5.3.6, CNN models reach significantly lower recognition rates in FedAR due to the high complexity of learning reliable features from limited labeled data. For this reason, in FedAR, the activity classification model is based on a fully connected deep learning model, and the input is a vector of handcrafted features. In particular, we choose features that proved to be effective for HAR [37]. Recent studies in the HAR domain demonstrate that a good choice of handcrafted features and fully connected models can lead to recognition rates comparable to the ones of state-of-the-art CNN models [55].

In particular, for each axis of each inertial sensor, we consider the following features: *average*, *variance*, *standard deviation*, *median*, *mean squared error*, *kurtosis*, *symmetry*, *zero-crossing rate*, *number of peaks*, *energy*, and *difference between maximum and minimum*. These features are extracted from fixed-length temporal windows of sensor data of *w* seconds. Before feature extraction, we apply a median filter on each temporal window to reduce noise in sensor measurements. After feature extraction, we apply standardization as a feature scaling technique.

4.2 Initialization of the global model

At the very beginning, the participating clients in FedAR need a pre-trained global model to infer labels on unlabeled data. However, in this work we assume limited availability of labeled data.

Hence, FedAR initializes the global model using a restricted annotated dataset (we will call it *pre-training dataset* in the following).[1] The *pre-training dataset* is also used to initialize label propagation algorithm. In realistic settings, the *pre-training dataset* can be, for example, a combination of publicly available datasets, or a small training set specifically collected by a restricted number of volunteers. Figure 4 summarizes the initialization mechanism of FedAR.

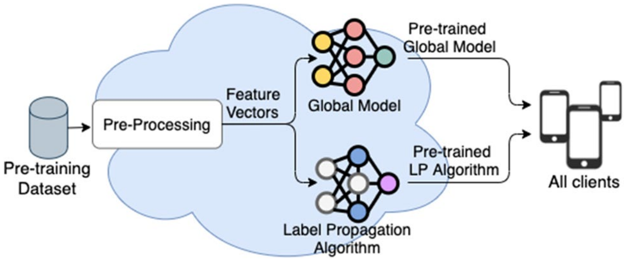


Fig. 4 Initialization of the global model in FedAR

4.3 The federated learning strategy of FedAR

In the following, we describe the FL process to update the global and local models. Periodically (e.g., each night) the server starts a global model update process. The devices that are available to perform computation (e.g., the ones idle and charging) inform the server that they are eligible to take part in the FL process. Afterwards, the server executes several communication rounds to update the weights of the global model.

A *communication round* consists of the following steps:

- The server sends the latest version of the global weights to a fraction of the eligible devices
- Each device uses the labeled data in the *Feature Vectors Storage* to train the *Shareable Model*
- When local training is completed, each device sends the new weights of the *Shareable Model* to the server
- The server aggregates the local weights to compute the new global weights

The communication rounds are repeated until the global model converges. Then, the new weights are transmitted to each participating device including the ones that did not actively contribute to the communication rounds.

The server updates the global model weights by executing a weighted average of the locally learned model weights provided from clients. Since the local weights may reveal private information, the aggregation is performed using the Secure Multiparty Computation approach presented in [41]. The pseudo-code of the server-side federated learning process is described in Algorithm 1, while the client-side in Algorithm 2.

[1] Note that, considering out target application, a labeled dataset is a collection of timestamped inertial sensors data acquired from mobile/wearable devices during activity execution. Examples of such sensors are accelerometer, gyroscope, and magnetometer. The labels are annotated time intervals that indicate the time-span of each performed activity.

Algorithm 1 Server side - Federated global model.

```
$PT \Leftarrow$ pre-training set
Initialize global model $w^G$ with $PT$
$d \Leftarrow$ participating devices
for each periodic update (e.g., every night) do
    for each communication round do
        ask for eligibility to each device in $d$
        $ed \Leftarrow$ eligible devices
        $ed' \Leftarrow k$ devices randomly sampled from $ed$
        send $w^G$ to each device in $ed'$
        aggregate updated models' weights received from devices in $ed'$ with SMC [41]
    end for
end for
```

Algorithm 2 FedAR - Client side - Model update.

```
$pm \Leftarrow$ Personalized Model
$sm \Leftarrow$ Shareable Model
Update the Feature Vectors Storage using the Label Propagation algorithm in Section 4.5.2.
for each communication round $i$ do
    train $sm$ using labeled data in the Feature Vector Storage
    send $sm$ to the server
    receive updated global model $w_i^G$
    $sm \Leftarrow w_i^G$
end for
$pm \Leftarrow sm$
fine-tune $pm$ using the transfer learning method described in Section 4.4
```

4.4 Model personalization

FedAR adopts a transfer learning strategy to fine-tune the *Personalized Model* on each user. The intuition behind the personalization mechanism is that the last layers of the neural network (i.e., the ones closer to the output) encode personal characteristics of activity execution, while the remaining layers encode more general features that are common between different users [56].

As depicted in Fig. 5, we refer to the last l layers of the neural network as the *User Personalized Layers*, while we refer to the remaining ones as *Shared Hidden Layers*. In FedAR, when the update of the global model is complete, each client creates the *Personalized Model* as a copy of the *Shareable Model*. In order to fine-tune the *Personalized Model* on each user, the *Shared Hidden Layers* are freezed, and the *Feature Vector Storage* is used to train the *User Personalized Layers*.

4.5 Semi-supervised learning

In the following, we describe how each client semi-automatically provides labels to the stream of unlabeled sensor data. FedAR relies on a combination of two semi-supervised learning techniques: *Active Learning* and *Label Propagation*.

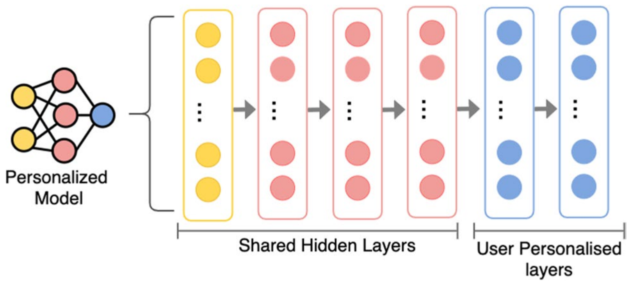


Fig. 5 Shared and personal layers

4.5.1 Active learning

An active learning process requires the user feedback about her currently performed activity when there is uncertainty in the classifier's prediction. The intuition is the following: unlabeled data samples for which the classification confidence is significantly low would have the most impact in improving the classifier if the label were available (i.e., they are the more informative ones).

FedAR relies on a state-of-the-art non-parametric active learning approach called *VAR-UNCERTAINTY* [57]. This approach compares the prediction confidence with a threshold $\theta \in [0, 1]$ that is dynamically adjusted over time. Initially, θ is initialised to $\theta = 1$. Let $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$ be the set of target activities. Given the probability distribution over the possible activities of the current prediction $\langle p_1, p_2, \dots, p_n \rangle$, we denote with $p^\star = \max_i p_i$ the probability value of the most likely activity $A^\star \in \mathbf{A}$ (i.e., the predicted activity). If $p^\star$ is below θ, we consider the system uncertain about the current activity performed by the user. In this case, an active learning process is started by asking the user the ground truth $A^f \in \mathbf{A}$ about the current activity. The feedback A^f is stored in the *Feature Vectors Storage*. When $A^f = A^\star$, it means that the most likely activity $A^\star$ is actually the one performed by the user, and hence the threshold θ is decreased to reduce the number of questions. On the other hand, when $A^f \neq A^\star$, θ is increased. More details about this active learning strategy can be found in [57]. The pseudo-code of classification and active learning is described in Algorithm 3

We assume that active learning queries are prompted to the user in real-time through a dedicated application, thanks to a user-friendly interface. Each query asks the user to choose the activity that she is currently performing among the possible ones. For the sake of usability, FedAR only presents a couple of alternatives taken from the most probable activities. Figure 6 shows a screenshot of an active learning application that we implemented for smart-watches in another research work.

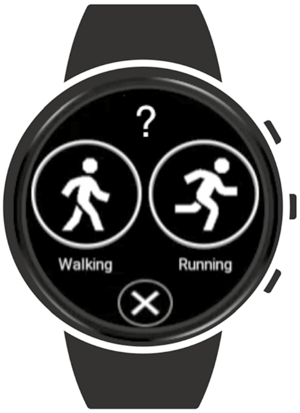


Fig. 6 Example of an active learning interface for smart-watches

4.5.2 Label propagation

The major drawback of active learning is that the queries may interrupt the user while performing an activity. In order to reduce the interaction with the user and, at the same time, to train the local models with a larger amount of labeled data, FedAR also relies on label propagation. The Label Propagation process is started when the server requires to update the global model (see Algorithm 1). Given a set of labeled and unlabeled data points, the goal of label propagation is to automatically spread labels to a portion of unlabeled data [58]. The intuition behind label propagation is that data points close in the feature space likely correspond to the same class label. The Label Propagation model of FedAR is a fully connected graph $g = (V, E)$ where the nodes V are all the data samples in the *Feature Vectors Storage* and the weight on each edge in E is the similarity between the connected data points. In the literature, this similarity is usually computed using K-Nearest Neighbors (KNN) or Radial Basis Function Kernel (RBF kernel). FedAR relies on the RBF kernel due to its trade-off between computational costs and accuracy [59]. Formally, the RBF kernel function is defined as $K(x, x') = e^{-\gamma||x-x'||^2}$ where $||x - x'||^2$ is the squared Euclidean distance between the feature vectors of two nodes x and x' (where x' is a labeled node), and $\gamma \in \mathbb{R}+$. Hence, the value of the RBF kernel function increases as the distance between data points decreases. The kernel is used to perform inductive inference to predict the labels on unlabeled data points, based on a threshold on the similarity between the nodes. This process is repeated until convergence (i.e., when there are no more unlabeled data points for which label propagation is reliable based on the threshold).

In FedAR, the Label Propagation model (i.e., the graph) is initialized with the labeled data points of the *pre-training dataset*. Moreover, this model is personal and never shared with other users nor with the server.

5 Experimental evaluation

In this section, we describe in detail the extensive experimental evaluation that we carried out to quantitatively assess the effectiveness of FedAR. First, we describe the public datasets that we considered in our experiments. Then, we present our novel evaluation methodology that aims at assessing both the generalization and personalization capabilities of FedAR. Finally, we discuss the results that we obtained on the target datasets.

Algorithm 3 Client side - Classification and data labeling.

```
$sm \Leftarrow$ Shareable Model
$pm \Leftarrow$ Personalized Model
receive pre-trained $w^G$ from the server
$sm \Leftarrow w^G$
$pm \Leftarrow w^G$
for each feature vector $fv$ computed in real-time from sensor data do
    $\vec{p} \Leftarrow$ probability distribution over the activities predicted by $pm$ on $fv$
    output the most likely activity according to $\vec{p}$
    if a feedback is needed according to VAR-UNCERTAINTY [57] then
        $l \Leftarrow$ activity label from the user
        add $(fv, l)$ to the Feature Vectors Storage
    else
        add $(fv, -)$ to the Feature Vectors Storage ▷ unlabeled data point
    end if
end for
```

5.1 Datasets

Since FL makes sense when many users participate in collaboratively training the global model, we considered publicly available datasets of physical activities (performed both in outdoor and indoor environments) that were collected involving a significant number of subjects. However, there are only a few public datasets with these characteristics. One of them is MobiAct [60], which includes labeled data from 60 different subjects with high variance in age and physical characteristics. The dataset contains data from inertial sensors (i.e., accelerometer, gyroscope, and magnetometer) of a smartphone positioned in a trousers' pocket freely chosen by the subject in any random orientation. 73% of the participants were male, while 27 are female. The subjects' age spanned between 20 and 47 (average: 26), the height ranged from 160cm to 189cm (average: 175), and the weight varied from 50kg to 120kg (average: 76). The adopted data acquisition frequency is the highest enabled by the sensors of the selected smartphone (i.e., at most 200Hz). Due to its characteristics, this dataset was also used in other works that proposed FL applied to HAR (e.g., the work presented in [48]). In our experiments, we considered the following physical activities:[2] *standing*, *walking*, *jogging*, *jumping*, and *sitting*. The distribution of activity labels in Mobiact is illustrated in Table 1.

Table 1 MobiAct: distribution of the considered activities

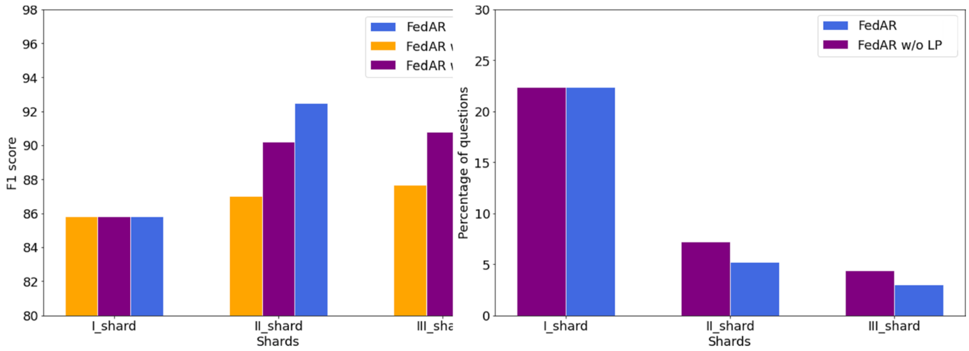
Activity	Percentage of samples
Standing	44%
Walking	42%
Jogging	6%
Jumping	6%
Sitting	2%
TOTAL	18.654 samples

Table 2 WISDM: distribution of the considered activities

Activity	Percentage of samples
Walking	38%
Jogging	30%
Sitting	6%
Standing	5%
Upstairs	11%
Downstairs	10%
TOTAL	13.726 samples

We also consider the well-known WISDM dataset [12]. This dataset has been widely adopted as benchmark for HAR. WISDM contains accelerometer data (sampling rate 20HZ) collected from a smartphone located in the front pants leg pocket of each subject during activity execution. WISDM includes data from 36 subjects. The data collection was supervised by one of the WISDM team members to ensure the quality of the collected data. The activities included in this dataset are the following: *walking*, *jogging*, *sitting*, *standing*, and *taking stairs*. The distribution of activity labels in WISDM is illustrated in Table 2. Unfortunately, further information about the participants like gender, age, and weight distribution is not publicly available.

[2] Note that we omitted from MobiAct those physical activities with a limited number of samples. Indeed, as we will explain later, our evaluation methodology requires to partition the data of each user. Activities with a small number of samples would be insufficiently represented in each partition and hence they are not suitable for our evaluation. We believe that this problem is only related to this specific dataset and that, in realistic settings, even short activities would be represented by a sufficient number of samples.



(a) Average F1 score

(b) Average percentage of active learning questions.

Fig. 7 MobiAct: The impact of label propagation and active learning on the subjects that participated in the FL process.

5.2 Evaluation methodology

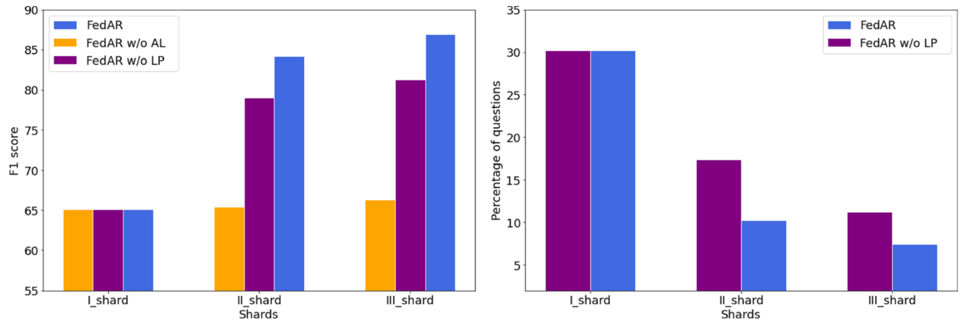
In the following, we describe the methodology that we designed to evaluate the effectiveness of FedAR, both in terms of personalization and generalization. We split each dataset into three partitions that we call *Pt*, *Tr*, and *Ts*. The partition *Pt* (i.e., pre-training data) contains data of users that we only use to initialize the global model. *Tr* (i.e., training data) is the dataset partition that includes data of users who participate in FL. Finally, *Ts* (i.e., test data) is a dataset partition that includes data of left-out users that we only consider to periodically evaluate the generalization capabilities of the global model. In our experiments, we randomly partition the users as follows: 15 % whose data will populate *Pt*, 65 % whose data will populate *Tr*, and 20 % whose data will populate *Ts*.

We partition the data for each user in *Tr* into *sh* shards of equal size. In realistic scenarios, each shard should contain data collected during a relatively long time period (e.g., a day) where a user executes many different activities. However, the considered datasets only have a limited amount of data for each user (usually less than 1 h of activities for each user). Hence, we generate shards as follows. Given a user $u \in Tr$, we randomly assign to each shard a fraction $\frac{1}{sh}$ of the available data samples associated with u in the dataset. Note that each data sample of a user is associated with exactly one shard. This mechanism allows us to simulate the realistic scenario described before, where users perform several types of activities in each shard.

Evaluation algorithm In the following, we describe our novel evaluation methodology step by step. First, the labeled data in *Pt* are used to initialize the global model, which is then distributed to the devices of all the users in *Tr* that will use it as the first version of the *Personalized Model*. We evaluate the recognition capabilities of the initial pre-trained global model on the partition *Ts* in terms of F1 score. This assessment allows us to measure how the initial global model generalizes on unseen users before any FL step.

As we previously mentioned, for each user, we partition its data samples in *Tr* into exactly *sh* shards. For the sake of evaluation, we assume a synchronous system in which the shards of the different users in *Tr* are actually temporally aligned and occur simultaneously (i.e, the first shards of every user occur at the same time interval, the second shards of every user occur at the same time interval, and so on). Note that, in the considered datasets, each user has a different data distribution and a different number of samples. Hence, within a specific shard, each client contributes with data collected considering its personal distribution. The evaluation process is composed by *sh* iterations, one for each shard. Considering the *i*-th shard we proceed as follows:

1. The devices of the users in *Tr* exploits the *Personalized Model* to classify the continuous stream of inertial sensor data in its shard. We use the classification output to evaluate the recognition rate in terms of F1 score providing an assessment of personalization. Note that, during this phase, we also apply our active learning strategy and we keep track of the number of triggered questions.
2. When all data in the shard have been processed (by all devices), the server starts a number *r* of communication rounds with a subset of the devices in order to update the global weights. Each round is implemented as follows:
   (a) The server randomly selects a certain percentage *p*% of users in *Tr* and sends to their devices the last update of the global weights.
   (b) Each user's device, by receiving the global weights, applies Label Propagation (See Section 4.5.2) and uses the newly labeled data to train its *Shareable Model*. After training, the resulting weights are transmitted to the server.
   (c) The server merges the received weights obtaining a new version the global model weights.



(a) Average F1 score

(b) Average percentage of active learning questions.

Fig. 8 WISDM: The impact of label propagation (LP) and active learning (AL) on the subjects that participated in the FL process.

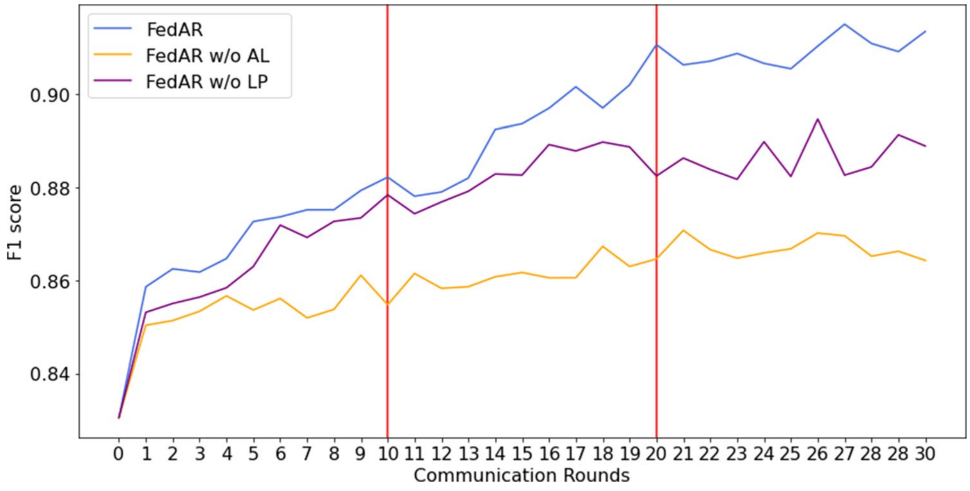


Fig. 9 MobiAct: the trend of F1 score on the left-out users after each communication round. This Figure also shows the impact of active learning and label propagation. Each red line marks the end of a shard

(d) We evaluate in terms of F1 score the recognition rate of the resulting global model on the left-out users in *Ts* (providing an assessment of generalization).

3. After the execution of all the communication rounds, each users' device:

    (a) replaces the weights of the *Shareable Model* and *Personalized Model* with the ones of the latest global model

    (b) fine-tunes the *Personalized Model* with labeled data from active learning and label propagation

    (c) starts the personalization process described in Section 4.4.

Note that our evaluation methodology introduces several levels of randomness: assigning users to *Ts*, *Tr* and *Pt*; assigning data samples to shards; selecting devices at each communication round. We iterate experiments 10 times and average the results in order to make our estimates more robust.

5.3 Results

In the following, we report the results of the evaluation of FedAR.

5.3.1 Classification model and hyper-parameters

As explained and motivated in Section 4.1, our classification model is a fully connected deep neural network. The network consists of four fully connected layers having respectively 128, 64, 32, and 16 neurons, and a softmax layer for classification. We use Adam [61] as optimizer. The choice of this specific network architecture is due to the good performance reported in the federated HAR literature [48]. As hyper-parameters, we empirically chose $w = 4s$, $p = 30\%$, $r = 10$, $l = 2$, $sh = 3$, and 10 local training epochs with a batch size of 30 samples. These hyper-parameters have been empirically determined based on data in *Ts*. The low number of epochs and communication rounds is due to the small size of the public datasets. This also limits the data in each shard. In a large-scale deployment, these parameters should be accurately calibrated.

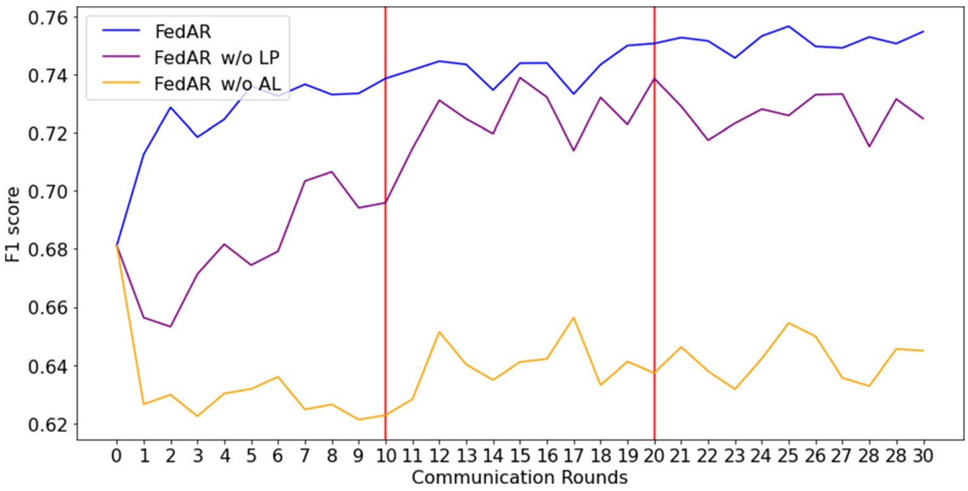


Fig. 10 WISDM: the trend of F1 score on the left-out users after each communication round. This Figure also shows the impact of active learning and label propagation. Each red line marks the end of a shard

5.3.2 Impact of semi-supervised learning

Figure 7 and Fig. 8 show how the F1 score and the percentage of active learning questions change at each shard for the users in *Tr*.

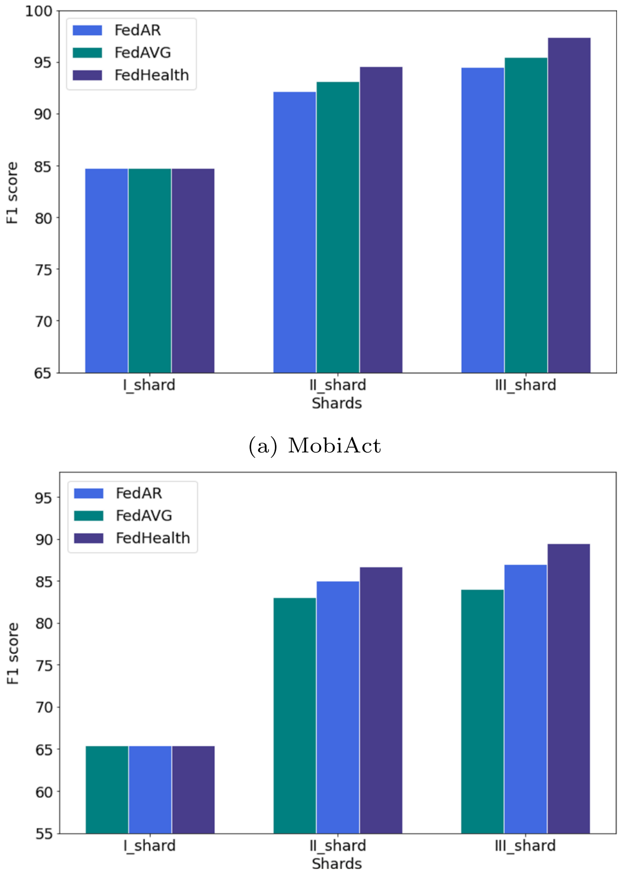
We observe that the F1 score significantly improves shard after shard, while the number of active learning questions decreases. Averaging the results of both datasets, the number of active learning questions at the first shard is around 25%, while at the last shard is only around 5%. This result indicates that our method continuously improves the recognition rate with a limited amount of labels provided by the users. Moreover, the continuous decrease of the number of questions militates for the usability of our method, which will prompt fewer and fewer questions with time. These figures also show the impact of combining active learning with label propagation. Without label propagation, active learning alone leads to a lower recognition rate and a higher number of questions. This means that the labeled data points derived by label propagation positively improve the activity model. On the other hand, we observe that label propagation leads to unsatisfying results without active learning. Indeed, the labeled samples obtained by active learning represent informative data that are crucial for label propagation. Hence, the evaluation on both datasets confirms that the combination of active learning and label propagation leads to the best results.

In Fig. 9 and Fig. 10 we show the generalization capability of the global model on left-out users (i.e., users in partition *Ts*) after each communication round performed during the FL process with the users in *Tr*.

The red lines mark the end of each shard. The results indicate that the federated model constantly improves also for those users that did not contribute with training data, even if the active learning questions continuously decrease. These plots also confirm that the combination of label propagation and active learning leads to the best results on both datasets.

5.3.3 FedAR versus approaches based on fully labeled data

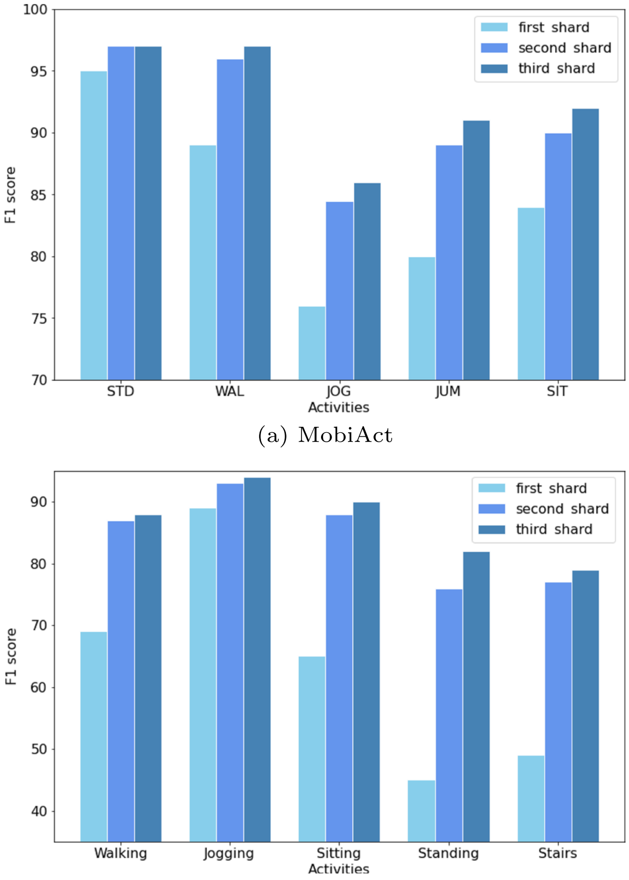
We compared our approach with two existing FL methods based on fully labeled data. The first one is the well-known FedAVG [6], which is the most common FL method in the



(a) MobiAct

(b) WISDM

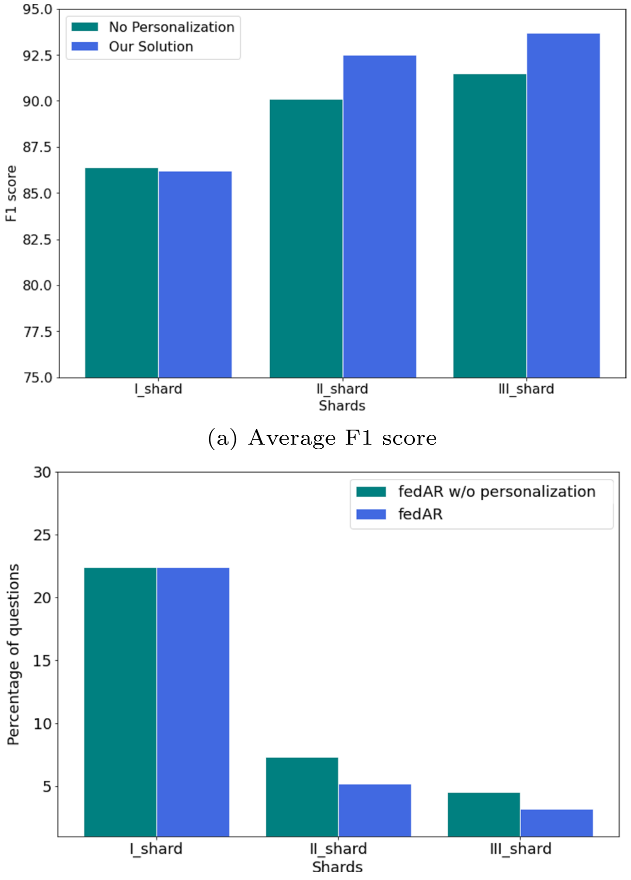
Fig. 11 Comparison of FedAR with methods based on fully labeled data.



(a) MobiAct

(b) WISDM

Fig. 12 F1 score at each shard for each activity on the users that participated in the FL process.



(a) Average F1 score

(b) Average percentage of active learning questions

Fig. 13 MobiAct: results on the users that participated to the FL process for each shard, with and without personalization.

literature. FedAVG simply averages the model parameters derived by the local training on the participating nodes (without any personalization).

The second method that we use for comparison is called FedHealth [8]. This is one of the first FL approaches proposed for activity recognition on wearable sensors data. Similarly to our approach, FedHealth applies personalization using transfer learning.

Since FedAR considers a limited amount of available labeled data, our goal is to achieve a recognition rate that is as close as possible to the one obtained by solutions that assume full availability of annotations.

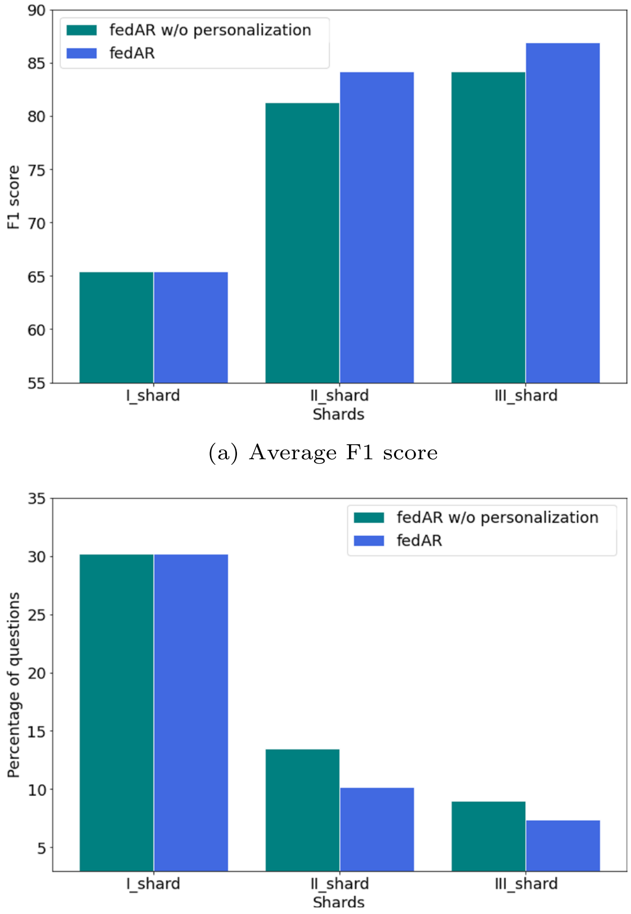
For the sake of fairness, in our experiments we adapted FedAVG and FedHealth to use the same neural network that we use in FedAR. Hence, we performed our experiments using our evaluation methodology by simulating that, for FedAVG and FedHealth, each node has the ground truth for each data sample on each shard. Hence, the evaluation of those methods does not include active learning and label propagation. Moreover, differently from FedAR, FedAVG and FedHealth only use a single local model.

The results of this comparison for the users in *Tr* (i.e., the ones that actively participated in the FL process) are reported in Fig. 11a and b.

From these plots, we observe that FedAR converges to recognition rates that are similar to solutions based on fully labeled data at each shard. The advantage of FedAR is that it can be used for realistic HAR deployments where the availability of labeled data is scarce. Despite a reduced number of required annotations, FedAR performs even better than FedAVG on the WISDM dataset, while on MobiAct it performs slightly worse. Moreover, FedAR is only ≈ 3% behind FedHealth on both datasets.

5.3.4 FedAR performance on each activity

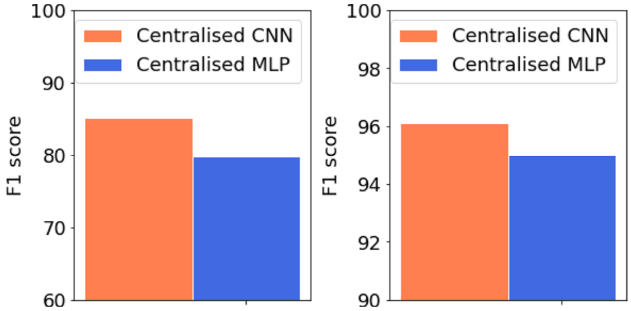
Figure 12 shows how the recognition rate improves between shards for each activity for the users in *Tr* on both datasets.



(a) Average F1 score

(b) Average percentage of active learning questions

Fig. 14 WISDM: results on the users that participated to the FL process for each shard, with and without personalization.



(a) WISDM

(b) MobiAct

Fig. 15 Centralized setting: MLP vs CNN based on leave-one-subject-out cross-validation.

We observed an improvement of the recognition rate shard after shard for each considered activity. The only exception is the *standing* activity on the MobiAct dataset in the third shard, which maintains the same F1 score.

In general, the greatest improvement occurs between the first and the second shards. This is due to the fact that, in the first shard, activities are recognized using the initial global model only trained with the *pre-training dataset*. Starting from the second shard, classification is performed with the *Personalized Model* updated thanks to FL and personalized using our transfer learning approach.

5.3.5 Impact of personalization

Figure 13 and Fig. 14 show the impact of the FedAR personalization strategy based on transfer learning. This evaluation is performed on the users in the *Tr* partition.

As expected, fine-tuning the personal models leads to an improvement both on the recognition rate and on the number of questions in active learning. Note that, during the first shard, classification is performed using the weights derived from the *pre-trained dataset* and personalization is applied starting from the second shard.

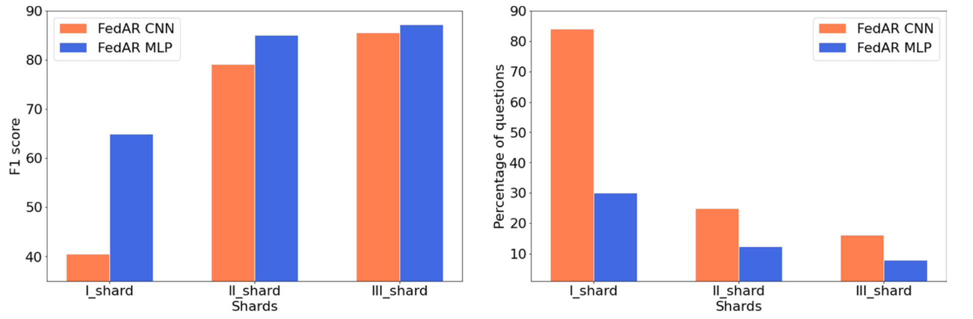
5.3.6 Fully connected vs convolutional models

The classification model in FedAR is a fully connected network (we will refer it as MLP[3] for the sake of brevity) that receives as input handcrafted feature vectors. Nonetheless, Convolutional Neural Networks (CNNs) proved to be very effective in fully supervised HAR approaches, since they can automatically learn features from raw data [55].

We performed a preliminary experiment to compare MLP and CNN in a fully supervised centralized approach using a leave-one-subject-out cross-validation. As CNN architecture, we consider the one recently proposed in [62] since it proved to be one of the most effective for HAR. Figure 15 shows the outcome of this comparison. We observe that, considering a fully supervised centralized setting, CNN is more effective on both datasets.

However, we observed that CNN struggles in learning reliable features considering our federated and semi-supervised setting, since the amount of labeled data to train the classifier is limited (cold start issue). Figures 16 and 17 show the comparison of FedAR using our MLP model with handcrafted features and the CNN model. On both datasets, MLP quickly reaches a higher F1 score with respect to CNN with a significantly lower number of active learning queries. Since features are computed a priori, the MLP model

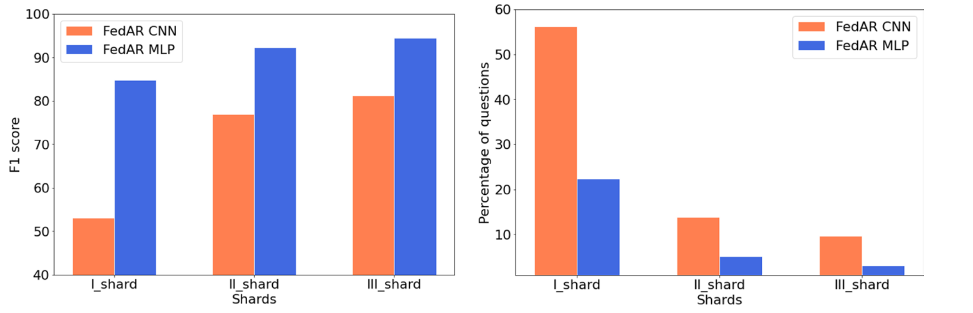
[3] MultiLayer Perceptron



(a) Average F1 score

(b) Average percentage of active learning questions

Fig. 16 WISDM: results on the users that participated to the FL process for each shard using both CNN and MLP networks



(a) Average F1 score

(b) Average percentage of active learning questions

Fig. 17 MobiAct: results on the users that participated to the FL process for each shard using both CNN and MLP networks

can immediately focus on training the classification layers rather than learning features. Hence, these results motivate our choice of adopting a MLP model with handcrafted features in FedAR.

6 Discussion

6.1 Generality of the approach

While we designed FedAR with wearable-based activity recognition as target application, we believe that this combination of semi-supervised and FL can be applied also to many other applications. Our method is suitable for human-centered classification tasks that include the following characteristics:

- There is a large number of clients that participate in the FL process.
- Classification needs to be performed on a continuous data stream, where labels are not naturally available.
- Each node generates a significant amount of unlabeled data.
- It is possible to periodically obtain the ground truth by delivering active learning questions to users that are available to provide a small number of labels. Note that, in real-time applications like HAR, for the sake of usability active learning questions should be prompted temporally close to the prediction.
- It is possible to obtain a limited training set to initialize the global model. Hence, a small group of volunteers should be available (in an initial phase) for annotated data acquisition.
- The nodes should be capable of computing training operations. Clearly, nodes can also rely on trusted edge gateways/servers (like proposed in [48]).

6.2 Privacy concerns

Despite FL is a significant step towards protecting user privacy in distributed machine learning, the shared model weights may still reveal some sensitive information about the participating users [63, 64]. Similarly to other works, FedAR uses Secure Multiparty Computation (SMC) [65, 66] to mitigate this problem. However, other approaches have been proposed, including differential privacy (DP) [67, 68], and hybrid approaches that combine SMC and DP [69].

The advantage of DP is the reduced communication overhead, with the cost of affecting the accuracy of the model. For the sake of this work, we opted for SMC in order to more realistically compare the effectiveness of our semi-supervised approach with other approaches, considering privacy as an orthogonal problem. However, we also plan to investigate how to integrate differential privacy in our framework and its impact on the recognition rate.

6.3 Need of larger datasets for evaluation

We evaluated FedAR choosing those well-known public HAR datasets that involved the highest number of users, simulating the periodicity of FL iterations by partitioning the dataset. However, the effectiveness of FedAR on large-scale scenarios should be evaluated on significantly larger datasets. By larger, we mean in terms of the number of users involved, the amount of available data for each user, and the number of target activities. Indeed, FedAR makes sense when thousands of users are involved, continuously performing activities day after day. However, observing the encouraging results on our limited datasets, we are confident that FedAR would perform even better on such large-scale evaluations.

Another limitation of the considered datasets is that both of them include data from only one position of the mobile device (trousers pocket). Since mobile devices can actually have different positions (e.g., wristbands), a larger evaluation should also consider different positions of the mobile device.

6.4 Resource efficiency

It is important to mention that FedAR is not optimized in terms of computational efficiency. Indeed, training two deep learning models on mobile devices may be computationally demanding and it may be problematic especially for those devices with low computational capabilities. This problem could be mitigated by relying on trusted edge gateways, as proposed in [46].

We want to point out that several research groups are proposing effective ways to dramatically reduce computational efforts for deep learning processes on mobile devices [70, 71]. Moreover, the GPU modules embedded in recent smartphones exhibit performances similar to the ones of entry-level desktop GPUs and this trend is expected to improve in the next few years [72].

Another limitation of our work is that the label propagation model requires storing the collected feature vectors as a graph. This is clearly not sustainable for a long time on a mobile device. This problem could be solved by imposing a limit on the size of the label propagation graph and periodically deleting old or poorly informative nodes.

7 Conclusion and future work

In this work we presented FedAR, a novel semi-supervised federated learning framework for activity recognition on mobile devices. FedAR takes into account the data scarcity problem, combining active learning and label propagation to semi-automatically annotate sensor data for each user. To the best of our knowledge, FedAR is the first application of federated learning to personalized activity recognition that is not based on the assumption that labeled data exists for all participating clients. Our results show that the combination of active learning and label propagation leads to recognition rates that are close to the ones reached by solutions that rely on fully supervised learning to train the local models.

In future work, we plan to investigate how federated clustering can further help improving the non-IID problem for HAR [73]. Indeed, HAR is more effective when the collaborative model only involves users that are similar between them [74]. We will study solutions based on federated clustering to automatically group users considering model similarity, creating a dedicated federated model for each cluster. Also, we plan to extend FedAR to automatically learn features from the unlabeled data stream, following the research direction proposed in [51].

Funding Open access funding provided by Università degli Studi di Milano within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Lara OD, Labrador MA et al (2013) A survey on human activity recognition using wearable sensors. IEEE Communications Surveys and Tutorials 15(3):1192–1209
2. Cook DJ, Feuz KD, Krishnan NC (2013) Transfer learning for activity recognition: A survey. Knowledge and Information Systems 36(3):537–556

3. Abdallah ZS, Gaber MM, Srinivasan B, Krishnaswamy S (2018) Activity recognition with evolving data streams: A review. ACM Computing Surveys (CSUR) 51(4):71
4. Bettini C, Riboni D (2015) Privacy protection in pervasive systems: State of the art and technical challenges. Pervasive and Mobile Computing 17:159–174
5. Weiss GM, Lockhart J (2012) The impact of personalization on smartphone-based activity recognition. In: Workshops at the twenty-sixth AAAI conference on artificial intelligence. Citeseer
6. McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp 1273–1282. PMLR
7. Yang Q, Liu Y, Chen T, Tong Y (2019) Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST) 10(2):1–19
8. Chen Y, Qin X, Wang J, Yu C, Gao W (2020) Fedhealth: A federated transfer learning framework for wearable healthcare. IEEE Intell Syst
9. Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, Bonawitz K, Charles Z, Cormode G, Cummings R et al (2019) Advances and open problems in federated learning. arXiv:1912.04977
10. Hard A, Rao K, Mathews R, Ramaswamy S, Beaufays F, Augenstein S, Eichner H, Kiddon C, Ramage D (2018) Federated learning for mobile keyboard prediction. arXiv:1811.03604
11. Ek S, Portet F, Lalanda P, Vega G (2020) Evaluation of federated learning aggregation algorithms: application to human activity recognition. In: Adjunct proceedings of the 2020 ACM international joint conference on pervasive and ubiquitous computing and proceedings of the 2020 ACM international symposium on wearable computers. pp 638–643
12. Kwapisz JR, Weiss GM, Moore SA (2011) Activity recognition using cell phone accelerometers. ACM SigKDD Explorations Newsletter 12(2):74–82
13. Györbíró N, Fábián Á, Hományi G (2009) An activity recognition system for mobile phones. Mobile Networks and Applications 14(1):82–91
14. Sun L, Zhang D, Li B, Guo B, Li S (2010) Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations. In: International conference on ubiquitous intelligence and computing. Springer, pp 548–562
15. Bao L, Intille SS (2004) Activity recognition from user-annotated acceleration data. In: Pervasive computing: Second international conference, PERVASIVE 2004, Linz/Vienna, Austria, April 21-23, 2004. Proceedings. Springer, pp 1–17
16. Bulling A, Blanke U, Schiele B (2014) A tutorial on human activity recognition using body-worn inertial sensors. ACM Computing Surveys 46(3):33–13333
17. Kwon Y, Kang K, Bae C (2014) Unsupervised learning for human activity recognition using smartphone sensors. Expert Systems with Applications 41(14):6067–6074
18. Chen L, Nugent C (2009) Ontology-based activity recognition in intelligent pervasive environments. Int J Web Inf Syst
19. Civitarese G, Sztyler T, Riboni D, Bettini C, Stuckenschmidt H (2019) Polaris: Probabilistic and ontological activity recognition in smart-homes. IEEE Transactions on Knowledge and Data Engineering 33(1):209–223
20. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16:321–357
21. Rashid KM, Louis J (2019) Times-series data augmentation and deep learning for construction equipment activity recognition. Advanced Engineering Informatics 42:100944
22. Wang J, Chen Y, Gu Y, Xiao Y, Pan H (2018) Sensorygans: An effective generative adversarial framework for sensor-based human activity recognition. In: 2018 International joint conference on neural networks (IJCNN). IEEE, pp 1–8
23. Chan MH, Noor MHM (2020) A unified generative model using generative adversarial network for activity recognition. J Ambient Intell Humanized Comput, 1–10
24. Cook D, Feuz KD, Krishnan NC (2013) Transfer learning for activity recognition: A survey. Knowledge and Information Systems 36(3):537–556
25. Wang J, Zheng VW, Chen Y, Huang M (2018) Deep transfer learning for cross-domain activity recognition. In: Proceedings of the 3rd international conference on crowd science and engineering. pp 1–8
26. Sanabria AR, Zambonelli F, Ye J (2021) Unsupervised domain adaptation in activity recognition: A gan-based approach. IEEE Access 9:19421–19438
27. Soleimani E, Nazerfard E (2021) Cross-subject transfer learning in human activity recognition systems using generative adversarial networks. Neurocomputing 426:26–34
28. Stikic M, Van Laerhoven K, Schiele B (2008) Exploring semi-supervised and active learning for activity recognition. In: 2008 12th IEEE international symposium on wearable computers. IEEE, pp 81–88
29. Guan D, Yuan W, Lee Y-K., Gavrilov A, Lee S (2007) Activity recognition based on semi-supervised learning. In: 13th IEEE international conference on embedded and real-time computing systems and applications, 2007. RTCSA 2007. IEEE, pp 469–475
30. Longstaff B, Reddy S, Estrin D (2010) Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In: Pervasive computing technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on Pervasive Computing Technologies for Healthcare. IEEE, pp 1–7
31. Stikic M, Larlus D, Schiele B (2009) Multi-graph based semi-supervised learning for activity recognition. In: 2009 International symposium on wearable computers. IEEE, pp 85–92
32. Lee Y-S, Cho S-B (2014) Activity recognition with android phone using mixture-of-experts co-trained with labeled and unlabeled data. Neurocomputing 126:106–115
33. Miu T, Missier P, Plötz T (2015) 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, pp 1138–1147
34. Abdallah ZS, Gaber MM, Srinivasan B, Krishnaswamy S (2015) Adaptive mobile activity recognition system with evolving data streams. Neurocomputing 150:304–317
35. Hossain HS, Khan MAAH, Roy N (2017) Active learning enabled activity recognition. Pervasive and Mobile Computing 38:312–330
36. Nguyen KT, Portet F, Garbay C (2018) Dealing with imbalanced data sets for human activity recognition using mobile phone sensors. In: 3rd international workshop on smart sensing systems
37. Bettini C, Civitarese G, Presotto R (2020) Caviar: Context-driven active and incremental activity recognition. Knowledge-Based Systems 196:105816
38. Voigt P, Von dem Bussche A (2017) The eu general data protection regulation (gdpr). A Practical Guide, 1st edn. Springer International Publishing, Cham
39. Samarati P (2014) Data security and privacy in the cloud. In: International conference on information security practice and experience. Springer, pp 28–41
40. Konečnỳ J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D (2016) Federated learning: Strategies for improving communication efficiency. arXiv:1610.05492

41. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Ramage D, Segal A, Seth K (2017) Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. pp 1175–1191
42. Damaskinos G, Guerraoui R, Kermarrec A-M, Nitu V, Patra R, Taiani F (2020) Fleet: Online federated learning via staleness awareness and performance prediction. In: Proceedings of the 21st international middleware conference. pp 163–177
43. Fallah A, Mokhtari A, Ozdaglar A (2020) Personalized federated learning: A meta-learning approach. arXiv:2002.07948
44. Ek S, Portet F, Lalanda P, Vega G (2021) A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison. In: 19th IEEE international conference on pervasive computing and communications PerCom 2021
45. Sozinov K, Vlassov V, Girdzijauskas S (2018) Human activity recognition using federated learning. In: 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom). IEEE pp 1103–1111
46. Wu Q, He K, Chen X (2020) Personalized federated learning for intelligent iot applications: A cloud-edge based framework. IEEE Computer Graphics Appl
47. Zhao Y, Liu H, Li H, Barnaghi P, Haddadi H (2020) Semi-supervised federated learning for activity recognition. arXiv:2011.00851
48. Wu Q, Chen X, Zhou Z, Zhang J (2020) Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring. IEEE Trans Mob Comput
49. Lee S, Zheng X, Hua J, Vikalo H, Julien C (2021) Opportunistic federated learning: An exploration of egocentric collaboration for pervasive computing applications. In: 2021 IEEE international conference on pervasive computing and communications (PerCom). IEEE, pp 1–8
50. Kelli V, Argyriou V, Lagkas T, Fragulis G, Grigoriou E, Sarigiannidis P (2021) Ids for industrial applications: a federated learning approach with active personalization. Sensors 21(20):6743
51. Saeed A, Salim FD, Ozcelebi T, Lukkien J (2020) Federated self-supervised learning of multisensor representations for embedded intelligence. IEEE Internet of Things Journal 8(2):1030–1040
52. Yu H, Chen Z, Zhang X, Chen X, Zhuang F, Xiong H, Cheng X (2021) Fedhar: Semi-supervised online learning for personalized federated human activity recognition. IEEE Trans Mob Comput
53. Arivazhagan MG, Aggarwal V, Singh AK, Choudhary S (2019) Federated learning with personalization layers. arXiv:1912.00818
54. Nasr M, Shokri R, Houmansadr A (2019) Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In: 2019 IEEE symposium on security and privacy (SP). IEEE, pp 739–753
55. Cruciani F, Vafeiadis A, Nugent C, Cleland I, McCullagh P, Votis K, Giakoumis D, Tzovaras D, Chen L, Hamzaoui R (2019) Comparing cnn and human crafted features for human activity recognition. In: 2019 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computing, scalable computing & communications, cloud & big data computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). IEEE, pp 960–967
56. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: Advances in neural information processing systems. pp 3320–3328
57. Žliobaitė I, Bifet A, Pfahringer B, Holmes G (2013) Active learning with drifting streaming data. IEEE Transactions on Neural Networks and Learning Systems 25(1):27–39
58. Zhou D, Bousquet O, Lal TN, Weston J, Schölkopf B (2004) Learning with local and global consistency. Advances in Neural Information Processing Systems 16(16):321–328
59. Widmann N, Verberne S (2017) Graph-based semi-supervised learning for text classification. In: Proceedings of the ACM SIGIR international conference on theory of information retrieval. pp 59–66
60. Vavoulas G, Chatzaki C, Malliotakis T, Pediaditis M, Tsiknakis M (2016) The mobiact dataset: Recognition of activities of daily living using smartphones. In: ICT4AgeingWell. pp 143–151
61. Kingma DP, Ba J (2017) Adam: A method for stochastic optimization
62. Wan S, Qi L, Xu X, Tong C, Gu Z (2020) Deep learning models for real-time human activity recognition with smartphones. Mobile Networks and Applications 25(2):743–755
63. Nasr M, Shokri R, Houmansadr A (2018) Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks. arXiv:1812.00910
64. Shokri R, Stronati M, Song C, Shmatikov V (2017) Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy (SP). IEEE, pp 3–18
65. Bhowmick A, Duchi J, Freudiger J, Kapoor G, Rogers R (2018) Protection against reconstruction and its applications in private federated learning. arXiv:1812.00984
66. Cramer R, Damgård I, Maurer U (2000) General secure multiparty computation from any linear secret-sharing scheme. In: International conference on the theory and applications of cryptographic techniques. Springer, pp 316–334
67. Agarwal N, Suresh AT, Yu FXX, Kumar S, McMahan B (2018) cpsgd: Communication-efficient and differentially-private distributed sgd. In: Advances in neural information processing systems. pp, 7564–7575
68. Dwork C (2008) Differential privacy: A survey of results. In: International conference on theory and applications of models of computation. Springer, pp 1–19
69. Truex S, Baracaldo N, Anwar A, Steinke T, Ludwig H, Zhang R, Zhou Y (2019) A hybrid approach to privacy-preserving federated learning. In: Proceedings of the 12th ACM workshop on artificial intelligence and security. pp 1–11
70. Lane ND, Bhattacharya S, Georgiev P, Forlivesi C, Jiao L, Qendro L, Kawsar F (2016) Deepx: A software accelerator for low-power deep learning inference on mobile devices. In: 2016 15th ACM/IEEE international conference on information processing in sensor networks (IPSN). IEEE, pp 1–12
71. Zhang C, Patras P, Haddadi H (2019) Deep learning in mobile and wireless networking: A survey. IEEE Communications Surveys & Tutorials 21(3):2224–2287
72. Ignatov A, Timofte R, Kulik A, Yang S, Wang K, Baum F, Wu M, Xu L, Van Gool L (2019) Ai benchmark: All about deep learning on smartphones in 2019. In: 2019 IEEE/CVF international conference on computer vision workshop (ICCVW). IEEE, pp 3617–3635
73. Briggs C, Fan Z, Andras P (2020) Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In: 2020 international joint conference on neural networks (IJCNN). IEEE, pp 1–9.
74. Sztyler T, Stuckenschmidt H (2017) Online personalization of cross-subjects based activity recognition models on wearable devices. In: 2017 IEEE international conference on pervasive computing and communications (PerCom). IEEE, pp 180–189