FedCLAR: Federated Clustering for Personalized Sensor-Based Human Activity Recognition

1st Riccardo Presotto
Dept. of Computer Science
University of Milan
Milan, Italy
riccardo.presotto@unimi.it

2nd Gabriele Civitarese
Dept. of Computer Science
University of Milan
Milan, Italy
gabriele.civitarese@unimi.it

3rd Claudio Bettini
Dept. of Computer Science
University of Milan
Milan, Italy
claudio.bettini@unimi.it

***Abstract*—Sensor-based Human Activity Recognition (HAR) has been a hot topic in pervasive computing for several years mainly due to its applications in healthcare and well-being. Centralized supervised approaches reach very high recognition rates, but they incur privacy and scalability issues. Federated Learning (FL) has been recently proposed to mitigate these issues. Each subject only shares the weights of a personal model trained locally, instead of sharing data. A cloud server is in charge of aggregating the weights to generate a global model. However, since activity data is non-independently and identically distributed (non-IID), a single model may not be sufficiently accurate for a large number of diverse users. In this work, we propose FedCLAR, a novel federated clustering method for HAR. Based on the similarity of the local model updates, the cloud server in FedCLAR derives groups of users that exhibit similar ways of performing activities. For each group, FedCLAR uses a specialized global model to mitigate the non-IID problem. We evaluated FedCLAR on two well-known public datasets, showing that it outperforms state-of-the-art FL solutions.**

***Index Terms*—human activity recognition, federated learning, clustering**

I. Introduction

Sensor-based Human Activity Recognition (HAR) is a well-established research area extensively studied by the pervasive computing community, mostly due to its impact on healthcare and well-being applications [1]. The most accurate HAR approaches are based on supervised machine learning, whose objective is to learn from labeled data the complex relationships between raw sensor data (e.g., accelerometer readings from a smartphone) and the activities performed by the user (e.g., running, taking the stairs). However, the deployment of real-world HAR systems is limited by several open research problems that still need to be addressed [2].

A major challenge is that building an activity recognition model using data from a large number of users poses several limitations regarding privacy and scalability. Indeed, locally collected labeled data is usually transmitted to a cloud server in charge of building an activity model by relying on a large amount of data. However, from the privacy point of view, activity data can be considered sensitive, since they may reveal personal habits or health conditions. On the other hand, this approach on a large scale may also pose issues related to communication latency and computational costs [3].

Federated Learning (FL) is a recently proposed learning paradigm, that shifts the training burden from the cloud to edge devices that are closer to the user and trusted [4]. Each edge device (e.g., a smartphone or home gateway) is in charge of training a local model with available labeled data. Then, only the weights of the resulting model are transmitted to the cloud server. By sharing model parameters instead of data, FL mitigates the above-mentioned privacy and scalability issues on large-scale scenarios.

The FL paradigm recently attracted attention from the pervasive computing community, including HAR [5]–[7]. Recent results show that FL solutions for HAR reach results that are similar to the ones of centralized approaches [8]. Despite the recognized potential of FL in real-world HAR scenarios, there are still some limitations. The major issue is that the FL cloud generates a global model that should generalize over a large number of users. However, different users may perform activities in very different ways depending on their physical characteristics, age, and habits. Indeed, data from different users are non-independently and identically distributed (non-IID). A trade-off between generalization and personalization should be considered by FL methods to build accurate HAR models [9].

The non-IID problem is well-known in FL [10]. One of the most promising approaches to mitigate this problem in HAR is to use transfer learning to fine-tune the local model on each user [8]. However, only relying on transfer learning and a single global model, it is challenging to balance personalization and generalization, especially considering large-scale scenarios [11]. Other approaches to mitigate the non-IID problem are based on multi-task learning [12]. However, those solutions are limited to models based on convex objective functions that limit both model complexity and scalability [13].

In the general literature on FL, *Federated Clustering* has been recently proposed to address the non-IID problem [13], [14]. In Federated Clustering, the cloud server builds specialized models for subsets of users to increase the recognition accuracy as well as to speed up the model convergence. In practice, the cloud server computes the pairwise similarity between the shared model weights to derive clusters of users. Intuitively, each cluster includes users that perform activities in a similar way. Federated Clustering mitigates the non-IID

problem while preserving the standard FL protocol between the clients and the server.

In this work, we propose FedCLAR: a federated clustering approach addressing the non-IID problem in FL-based HAR. With respect to existing federated clustering approaches, FedCLAR selects only a portion of the model weights shared by each client, with the objective of computing a similarity score and building groups of users using a hierarchical clustering algorithm. The selected weights intuitively characterize the subject-specific activity patterns. For instance, considering deep learning models, these would be the weights corresponding to layers that are closer to the output [15]. In FedCLAR, those users that can not be included in any cluster will use a generic global model that is trained by all the participating users, like in a standard federated learning setting. Moreover, FedCLAR also uses transfer learning methods to fine-tune activity recognition on each user to further improve personalization.

We evaluated FedCLAR on two well-known public datasets of sensor-based HAR, where labeled data is acquired from mobile devices. Our results show that FedCLAR outperforms FL-based state-of-the-art approaches that use transfer learning to tackle the non-IID problem. Moreover, our experimental evaluation also shows the advantage of combining federated clustering with transfer learning to improve personalization.

The contributions of this work are the following:

- We propose FedCLAR: a novel federated learning approach for personalized HAR based on hierarchical clustering and transfer learning.
- We design a new user clustering technique based on the server-side similarity computation, using only a portion of the model weights shared by each participating user.
- Our results on public datasets show that FedCLAR mitigates the non-IID problem, outperforming state-of-the-art FL approaches based on transfer learning.

II. Related work

A. Human Activity Recognition

Sensor-based HAR is a well-known research area, that has been deeply studied by several research groups in the last two decades [16]. The goal of HAR is to classify the current activity performed by the user analyzing the continuous stream of sensor data. Sensor-based HAR has been explored mainly considering two settings: using sensors of mobile/wearable devices [17], and using environmental sensors of smart-home environments [18]. Among the many proposed approaches, the most effective ones are based on Deep Learning (DL) [19]. However, the deployment of HAR systems in real-world scenarios is still limited by several open challenges. Among the many challenges, there are labeled data scarcity [20], the discovery of new activities [21], the transferability of activity models between different environments/users [22], and lifelong learning [23].

In this work, we focus on personalization, privacy, and scalability issues that are related to training an activity model with a large number of users. Several studies indicate that personalization is a crucial aspect in HAR [9]. However, the main problem of a large-scale collaborative model is that balancing personalization and generalization at the same time is challenging [8]. Moreover, activity data reveal private information about the subject, like health conditions and habits. Hence, this private information should be protected when outsourced to untrusted third parties [24].

B. Federated Learning for HAR

In order to tackle the scalability and privacy issues mentioned above, Federated Learning (FL) is a promising direction [4]. In FL, the training of the global activity model is distributed among the participating clients. Indeed, each client is in charge of training a local model with its available labeled data and it only transmits local model parameters (weights) to the server, instead of sharing private activity data. The server aggregates the weights received by the participating clients and creates a global model. Moreover, privacy-preserving mechanisms like Differential Privacy (DP) and Secure MultiParty Computation (SMC) are usually adopted to further protect the shared model weights from attacks that can potentially reverse-engineer data or properties from the weights [25].

According to a classification proposed by a recent survey [25], FedCLAR is a *horizontal* FL method: the participating clients share the same feature space, but they have a different data distribution (i.e., each client uses data of a specific subject).

FL has been recently applied to sensor-based HAR to distribute the training of the activity recognition model among the participating devices [3], [6]. Some of these approaches also propose to collaboratively learn the feature representation to mitigate the data scarcity problem [26], [27].

However, a major problem in FL for HAR is that data from different users are non-independently and identically distributed (non-IID). Indeed, different users may have different physical characteristics and habits. Hence, a single global model would lack in capturing those differences on a large number of users [10].

C. Tackling the non-IID problem in FL-based HAR

Several recent works proposed approaches to mitigate the non-IID problem in FL-based HAR. A common solution is to adopt transfer learning techniques on the client-side to improve personalization [8], [28], [29]. In particular, the global model is fine-tuned on each client by training the last layers of the personal deep learning model (i.e., the ones closest to the output) with personal data. The intuition behind this approach is that the last layers capture the personal patterns of the users, while the first layers encode cross-subject features [15]. However, those approaches are still based on a single global model, and balancing personalization and generalization is still a challenge.

Multi-task federated learning is another approach proposed to mitigate the non-IID problem in FL-based HAR [12]. In particular, the clients contribute to collaboratively learning

only the common features, while the diversity is handled at the client side. However, these approaches are based on a convex objective function that is not suitable for complex HAR models based on deep learning.

A very recent and closely related work, that is called ClusterFL, proposes a multi-task federated clustering method for FL-based HAR [11]. This approach is based on a distributed optimization approach: the clients and the cloud server collaborate in optimizing both the local models as well as the clustering structure. A limitation of ClusterFL is that the information about the association between users and clusters, as well as the parameters of each local model, are distributed to all the participating clients. Hence, ClusterFL does not adhere to the standard FL protocol and it reveals sensitive information to each client. Moreover, ClusterFL requires mobile devices to compute an optimization task that is more computationally expensive with respect to the usual local training required by FL approaches.

The recent FL literature proposes Federated Clustering approaches that adhere to the standard FL protocol, without revealing clustering details to the participating clients [13], [14], [30]. Inspired by those approaches, FedCLAR offers an effective Federated Clustering solution for HAR with the following new and distinctive features: a) the similarity is computed only on the subject-specific parameters of the local models, and b) it relies on transfer learning to further personalize HAR.

III. Problem formulation

A. Non-IID problem in HAR

Let $U = \{U_1, \ldots, U_n\}$ be the set of users. Each user U_i is associated with a labeled dataset $D_i = \{(x, y)\}$, where x is a data point and y the corresponding activity label. Let $D = \{D_1, \ldots, D_n\}$ be the set of datasets, each one corresponding to a user in U. D is non-independently and identically distributed (non-IID) if at least a pair of datasets $D_i, D_j \in D$ satisfies one of the following conditions [14]:

- **Feature distribution skew**: $P_{D_i}(x) \neq P_{D_j}(x)$. This inequality between probability distributions is true when the data samples in D_i have a significantly different marginal distribution than the ones in D_j. In HAR, this often happens since each subject may perform activities in a peculiar way. Among many factors, users' physical characteristics have a strong impact on the activity patterns. For instance, a young subject would probably have a faster walking pattern than an elder subject.
- **Label distribution skew**: $P_{D_i}(y) \neq P_{D_i}(y)$. This inequality between probability distributions is true when the labels in D_i have a significantly different marginal distribution than the ones in D_j. In HAR, this usually happens since different users may have different daily routines. For example, a sporty subject would likely spend more time $running$ or $cycling$ than a sedentary subject.
- **Quantity distribution skew**: This condition is true when $|D_i|$ and $|D_j|$ are significantly different. In HAR, is not unusual to have significantly different sizes of labeled samples for different subjects.

B. Why non-IID is a problem in a FL setting

Given a non-IID set of datasets D, a standard *centralized* ML approach builds a recognition model M^C by using all the annotated data points in $D^* = D_1 \bigcup D_2 ... \bigcup D_n$. In this case, the training phase consists in finding the parameters $\mathbf{w} \in \mathbb{R}^{\mathbf{d}}$ that minimize a global objective function $f(\mathbf{w})$:

$$\min_{\mathbf{w} \in \mathbb{R}^{\mathbf{d}}} f(\mathbf{w}),\ f(\mathbf{w}) := \frac{1}{|D^*|} \sum_{k=1}^{|D^*|} \ell_k(\mathbf{w}) \tag{1}$$

where $\ell_k(\mathbf{w})$ is a loss function. Intuitively, the objective is to find the parameters $\mathbf{w}$ that minimize the average loss over all the annotated samples in D^*. By considering all the annotated samples at the same time, this *centralized* approach mitigates the non-IID problem.

However, there are significant differences in an FL setting. Indeed, each user U_i locally trains a model M_i, and it transmits to the server only the M_i parameters $\mathbf{w_i}$. The server is in charge of building a global model $\overline{M}$ from the local parameters $\mathbf{W} = \langle \mathbf{w_1}, \ldots, \mathbf{w_n} \rangle$, and it is not possible to directly access D^*. The objective function $\overline{f}(\overline{w})$ of the federated model to derive the global parameters $\overline{w}$ is the following:

$$\min_{\overline{\mathbf{w}} \in \mathbb{R}^d} \overline{f}(\overline{\mathbf{w}}),\ \overline{f}(\overline{\mathbf{w}}) = \sum_{i=1}^{n} \frac{|D_i|}{|D^*|} f_i(\mathbf{w_i}) \tag{2}$$

where $f_i(\mathbf{w_i})$ is the local objective function that each user U_i minimizes by using D_i to obtain $\mathbf{w_i}$:

$$\min_{\mathbf{w_i} \in \mathbb{R}^d} f_i(\mathbf{w_i}),\ f_i(\mathbf{w_i}) := \frac{1}{|D_i|} \sum_{k=1}^{|D_i|} \ell_k(\mathbf{w_i}) \tag{3}$$

Ideally, the parameters of the *federated* model should approximate the ones of the *centralized* model. However, in a non-IID setting, the overall data distribution of D^* (that is captured by the *centralized* approach) may be considerably different from the distribution of each $D_i \in D$ that is captured by the *federated* approach. For this reason, minimizing $\overline{f}(\mathbf{w})$ may lead to a global model that would significantly underperform the one derived by minimizing $f(\mathbf{w})$.

C. The federated clustering problem

A possible solution to tackle the non-IID problem in the FL setting issue is to partition U into s clusters $C = C_1, ..., C_s$ so that each cluster minimizes the non-IID properties among the datasets of the users assigned to the same cluster. Hence, it is possible to derive a federated model $\overline{M}^{C_j}$ for each cluster. The objective function $\overline{f}^{C_j}(\overline{\mathbf{w}}^{C_j})$ of each model M^{C_j} can be optimized by using data from the cluster:

$$\min_{\mathbf{w}^{C_j} \in \mathbb{R}^d} \overline{f}^{C_j}(\overline{\mathbf{w}}^{C_j}),\ \overline{f}^{C_j}(\overline{\mathbf{w}}^{C_j}) = \sum_{i=1}^{|C^j|} \frac{|D_i|}{|D^{C_j}|} f_i(\mathbf{w_i}) \tag{4}$$

where D^{C_j} is the set of datasets of the users belonging to the cluster C_j. If the clusters actually capture the similarity between the distributions of the datasets, the resulting model would better approximate the one generated by a *centralized* approach on the users of the same cluster.

However, in the FL setting it is not possible to access each D_i to compute the clusters, since only the model parameters $\mathbf{w_i}$ are available. Hence, a major problem that we tackle in this work is how to compute user clustering in the FL setting.

IV. Methodology

In the following, we describe how FedCLAR mitigates the non-IID problem by combining federated clustering and transfer learning [1]. First, we explain how to compute the similarity between users only based on local model weights. Then, we describe our federated hierarchical clustering approach. Since different users in the same group may still have some peculiarities, we also describe how each client further personalizes its local model thanks to transfer learning.

A. Computing similarity between users

In general, clustering approaches are based on a similarity metric that is computed on each pair of items that may be clustered. In sensor-based HAR, similar users are those that share similar sensor data patterns (i.e., similar activity patterns). However, in an FL learning process, only the weights of the local models are available, and not sensor data. Nonetheless, if two local models share similar weights, they were likely trained with similar patterns of data. Hence, given the parameter vectors $\mathbf{w_i}$ and $\mathbf{w_j}$ of the models corresponding to the users U_i and U_j, it is possible to compute their similarity. FedCLAR relies on the cosine similarity since it proved to be effective for federated clustering [13]. The cosine similarity between the model weights of two users U_i and U_j can be computed as follows:

$$sim(\mathbf{w_i}, \mathbf{w_j}) = \frac{\mathbf{w_i} \cdot \mathbf{w_j}}{\|\mathbf{w_i}\|\|\mathbf{w_j}\|} \tag{5}$$

However, considering HAR models and the recent results on transfer learning [8], we realized that computing the similarity taking into account the whole parameter vector would not be the optimal choice. Considering local models based on deep learning, the closest layers to the input reflect high-level features that are common between all the subjects [15]. On the contrary, the layers that are closest to the output are the ones that encode user-specific activity patterns.

Let $pers(\mathbf{w})$ be a function that extracts from parameter vector $\mathbf{w}$ the user-specific parameters. Hence, FedCLAR computes the pairwise similarity between model weights as follows:

$$sim(\mathbf{w_i}, \mathbf{w_j}) = \frac{pers(\mathbf{w_i}) \cdot pers(\mathbf{w_j})}{\|pers(\mathbf{w_i})\|\|pers(\mathbf{w_j})\|} \tag{6}$$

[1]For the sake of this work and without loss of generality, we describe FedCLAR considering HAR based on sensor data acquired by mobile devices.

Since FedCLAR is based on deep learning, the function $pers(\mathbf{w})$ returns the weights corresponding to the last l layers of $\mathbf{w}$.

B. Hierarchical Clustering

Using the similarity function described above, the cloud server in FedCLAR can apply a clustering algorithm to derive groups of users that perform activities in a similar way. In this work, we use a hierarchical approach, since in the literature it proved to be effective for federated clustering [14].

The pseudo-code for the hierarchical clustering method of FedCLAR is described in Algorithm 1. The intuition behind this process is the following. Initially, there is one cluster for each user. Clusters are grouped based on the pairwise similarity of the participating users and a clustering threshold ct. When two clusters are merged into a single one, a new specialized model for that cluster is generated by merging the models of the clusters that originated it. The process is repeated until no more clusters can be grouped. In the end, our method only considers those clusters that contain more than one user. The users in the singleton clusters are considered as *non-clustered users*.

Algorithm 1 HierarchicalClustering

Input: $\mathbf{W} = \{\mathbf{w_1}, \ldots, \mathbf{w_n}\}$
Output: A set of clusters C, a set of specialized models W

$C \leftarrow \{\{U_1\}, \ldots, \{U_n\}\}$
$cmap \leftarrow$ empty map from model weights to clusters
$cmap[\mathbf{w_1}] \leftarrow \{U_1\}$
...
$cmap[\mathbf{w_n}] \leftarrow \{U_n\}$
do
  $P \leftarrow$ pairwise similarity matrix on $\mathbf{W}$ based on sim
  $\mathbf{w_a}, \mathbf{w_b} \leftarrow \underset{\mathbf{w_a}, \mathbf{w_b} | a \neq b}{\arg\min}\ P_{ab}$
  if $sim(\mathbf{w_a}, \mathbf{w_b}) \geq ct$ **then**
    $\mathbf{w_{ab}} \leftarrow$ merge $\mathbf{w_a}$ and $\mathbf{w_b}$ using FedAvg
    $c_a \leftarrow cmap[\mathbf{w_a}]$
    $c_b \leftarrow cmap[\mathbf{w_b}]$
    $c_{ab} \leftarrow c_a \cup c_b$
    $C \leftarrow C \setminus \{c_a, c_b\}$
    $C \leftarrow C \cup \{c_{ab}\}$
    $\mathbf{W} \leftarrow \mathbf{W} \setminus \{\mathbf{w_a}, \mathbf{w_b}\}$
    $\mathbf{W} \leftarrow \mathbf{W} \cup \{\mathbf{w_{ab}}\}$
    $cmap[\mathbf{w_{ab}}] = c_{ab}$
  else
    $\mathbf{W} \leftarrow \{\mathbf{w} \in \mathbf{W}$ *such that* $|cmap(\mathbf{w})| > 1\}$
    $C \leftarrow \{c \in C$ *such that* $|c| > 1\}$
    return C and $\mathbf{W}$
  end if
while True

C. The server side of FedCLAR

The sever-side mechanism of FedCLAR is described by Algorithm 2. Periodically (e.g., every night), the server requires an update of the global models. Hence, a sequence of

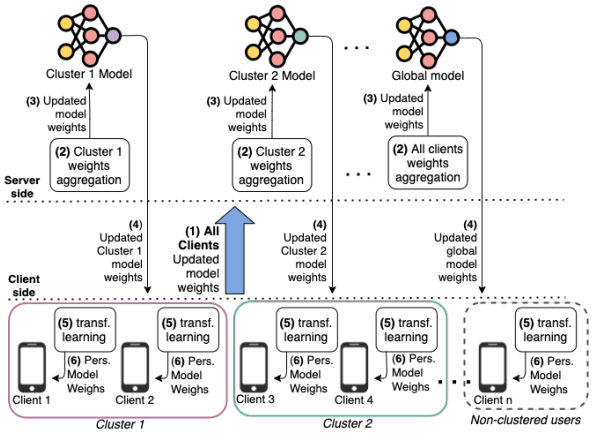


Fig. 1: Overall data flow of FedCLAR after the clustering communication round

communication rounds is started. Each client locally trains its model and transmits the resulting weights to the server. Upon receiving the weights from the clients, the first task of the server is generating an overall global model using FedAvg.

Then, the server continues with the federated clustering algorithm. An important observation is that computing the similarity between users is effective only if performed after a certain number of communication rounds. Otherwise, we experimentally observed the risk of considering models parameters that are not sufficiently trained, thus generating unreliable clusters. For this reason, in FedCLAR, our hierarchical clustering method explained in Section IV-B is performed after a predefined number r of communication rounds. From the communication round r, the clustered clients will update and use the specialized models, while the *non-clustered* users will update and use the overall global model. Note that, in order to provide to *non-clustered* users a global model with sufficient generalization capabilities, the clustered users also contribute to updating the overall global model.

D. The client side of FedCLAR

A valuable property of FedCLAR is that, considering standard FL solutions like FedAVG, the FL protocol is not altered from the client point of view. Indeed, each client collaborates to the federated clustering without knowing if it is collaborating in training a specialized model or to the overall global model. Algorithm 3 describes what happens at the client-side when the server requires a periodic update of the global model. Note that, even though federated clustering has the advantage of mitigating the non-IID problem, it is still possible that different users in the same cluster have some personal way of performing some activities. For instance, some users may be grouped in the same cluster because they perform the majority of the activities in a similar way, while they exhibit slight differences in the execution of a restricted number of activities. In order to further personalize the recognition model, FedCLAR also relies on transfer learning on each client. In particular, our solution is inspired by a strategy that proved to be effective in FL-based HAR [8]. Once a client concludes

Algorithm 2 FedCLAR - Server side

$C \leftarrow$ nil
$\mathbf{W}^C \leftarrow$ nil
for each periodic update (e.g., every night) **do**
  for each communication round i **do**
    receive $\mathbf{W}$=$\{\mathbf{w_1}, .., \mathbf{w_n}\}$ from clients
    $\mathbf{w^G} \leftarrow$ aggregate $\mathbf{W}$ using FedAvg (global model)
    if $i < r$ **then**
      send $\mathbf{w^G}$ to each client
    else
      if $i == r$ **then**
        $C, \mathbf{W}^C \leftarrow$ *HierarchicalClustering*($\mathbf{W}$)
      else
        use $\mathbf{W}$ to update $\mathbf{W}^C$ based on C
      end if
      for $c_i \in C$ **do**
        send $\mathbf{w}_i^C$ to each client in c_i
      end for
      send $\mathbf{w^G}$ to *non-clustered* clients
    end if
  end for
end for

the communication rounds, its local model is updated with the weights received from the cloud server. Then, a fine-tuning process starts, with the goal of better capturing the specific activity patterns of each user. The client's available labeled data are used one more time to update the new local model. However, this fine-tuning process only updates the last p layers (i.e., the ones closest to the output), while the remaining ones are left as received by the server.

E. The overall FedCLAR data flow

In order to summarize the general mechanism behind FedCLAR, Figure 1 depicts the federated learning data flow after the clustering communication round. Clients are grouped in clusters based on the similarity of their users. The clients in the same cluster collaborate, through their sharing of local

Algorithm 3 FedCLAR - Client side

```
lm ← local model
for each communication round do
    train lm using available labeled data
    send the weights w of lm to the server
    receive updated model w^S from the server
    replace the weights of lm with w^S
end for
freeze the layers of lm except for the last p layers
train lm using available labeled data
unfreeze lm layers
```

weights, to generate and refine their cluster model. All clients, including the ones of *non-clustered* users, collaborate to generate and refine a general global model. This model will be used by *non-clustered* users as well as users not participating in the federated training. Finally, each client uses transfer learning to further personalize the local recognition model.

V. Experimental evaluation

A. Datasets

In order to evaluate the effectiveness of FedCLAR, we considered two well-known HAR datasets: WISDM [31] and MobiAct [32]. Those datasets were selected since they involve a relatively large number of subjects with respect to other sensor-based HAR datasets. Despite a real deployment would involve a much larger number of participants, this aspect is crucial to evaluate our FL-based approach, considering that data (and participant) augmentation techniques may not lead to realistic results. Moreover, the subjects that participated to data collection in these datasets exhibit both data and labels distribution skew, that are necessary to evaluate the clustering capabilities of FedCLAR. WISDM includes labeled activity data from 36 different subjects obtained from the accelerometer of a smartphone placed in the pants pocket during the activity execution. The activities considered in this dataset are: *walking*, *jogging*, *sitting*, *standing*, and *taking stairs*. The MobiAct dataset includes labeled activity data from 60 different subjects. Those data were collected from the inertial sensors (i.e., accelerometer, gyroscope, and magnetometer) of a smartphone placed in the pants pocket. In our experiments, we considered the following physical activities *standing*, *walking*, *jogging*, *jumping*, *going upstairs*, *going downstairs*, and *sitting*.

B. Experimental setup

In order to evaluate the clustering capabilities of FedCLAR, the activity model that we used in our experiments is a simple feed-forward deep neural network. In particular, it is composed of three fully connected layers having respectively 32, 16, and 16 neurons, and a softmax layer for classification. The inputs of the network are hand-crafted feature vectors extracted in real-time from the stream of sensor data. We consider features that proved to be effective for HAR in the literature [1]. We used Adam [33] as optimizer. Even though existing FL approaches proposed more sophisticated deep learning classifiers (even to collaboratively learn features representation), a simpler model with hand-crafted features allowed us to focus only on the specific clustering problem. Moreover, we believe that an advantage of our simple model is a reduced computational effort, that is more suitable for mobile devices. The hyper-parameters were selected using a grid search, with the objective of optimizing the overall F1-score. In particular we chose $l = 1$, $p = 2$, $r = 5$, and 10 local training epochs with a batch size of 30 samples. Considering the clustering threshold ct, we chose $ct = 0.005$ for the WISDM dataset, and $ct = 0.010$ for the MobiAct dataset. We will describe the impact of those hyper-parameters on the recognition rate and on clustering quality in Section V-D5.

C. Evaluation methodology

For each user in the dataset, we simulate a corresponding client of FedCLAR. Each client uses the 70% of its data to participate in the collaborative model update (i.e., to create the clusters and the corresponding specialized models). When the communication rounds are concluded, the remaining 30% of data is used to evaluate the recognition rate of the resulting local model in terms of F1-score.

D. Results

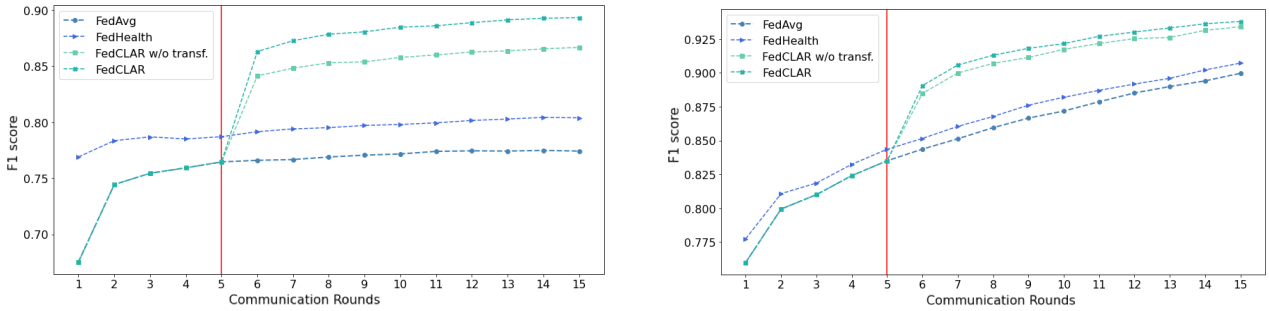
1) Considered baselines: We compared FedCLAR with three baselines. The first is *FedAvg*, which is the first FL method proposed in the literature [4]. *FedAvg* does not consider the non-IID problem. The second baseline is *FedHealth* [8], a recent FL-based HAR approach that tackles the non-IID problem using transfer learning. The third baseline is called *FedCLAR w/o transfer*, which is our approach without fine-tuning the local models with transfer learning.

2) Overall recognition rate: Table I compares FedCLAR with those baselines. Our results show that FedCLAR significantly outperforms the other approaches on both datasets. In particular, the comparison between *FedHealth* and *FedCLAR w/o transfer* shows that federated clustering leads to a higher F1-score with respect to only relying on transfer learning. Nonetheless, when federated clustering and fine-tuning are combined (i.e., FedCLAR), we observed a further slight improvement in the recognition rate.

TABLE I: Comparison of the F1-score of FedCLAR with respect to alternative approaches.

Dataset	FedAvg	FedHealth	FedCLAR w/o transfer	FedCLAR
WISDM	0.76	0.80	0.86	**0.89**
MobiAct	0.89	0.90	0.93	**0.94**

3) Recognition rate at each communication round: Figure 2 shows how the recognition rate evolves at each communication round. We observed that, from the communication round where clustering is performed, the recognition rates of FedCLAR and FedCLAR w/o transfer learning increase significantly with respect to the baselines. Moreover, these plots confirm the advantage of including transfer learning in the federated clustering process.



(a) WISDM (b) MobiAct

Fig. 2: Trend of F1 score at each communication round. The red line specifies the clustering communication round.

4) Cluster-based results: In the following, we show the results of FedCLAR at the cluster level. In particular, we consider the clusters generated by FedCLAR. For each cluster, we compare the F1 of FedCLAR with the ones reached by the considered baselines. Note that these results are just a detailed version of the ones proposed in Table I: each alternative FL approach is actually computed considering all the users, while we show the F1 score for subsets of the users based on the output of FedCLAR. This visualization strategy makes it possible to focus on the improvement of FedCLAR on the users of the generated clusters. These results are depicted in Figure 3. We observed that only a small percentage of users was not clustered. Since the general global model is the one used to classify for those users, the recognition rate of FedCLAR is similar to the ones of standard baselines like FedAvg. Also, we can observe that transfer learning has a limited impact on those users. On the other hand, we can observe that federated clustering has a significant impact on each cluster, especially considering the WISDM dataset. This trend is also confirmed on the MobiAct dataset, except for the first two clusters. This is due to the fact those users reflect the average subject in MobiAct. Hence, the global model in FedAvg is heavily influenced by those users, and it is already accurate in discriminating their activities. Moreover, we observed that the users of those two clusters are very similar (and probably the clusters should have been merged), thus confirming our hypothesis.

5) Impact of hyper-parameters:
Finally, we show the impact of the FedCLAR's hyper-parameters both on recognition rate and clustering. These results are reported in Table II and III. First, we observed that the higher the number l of layers that we consider to compute the similarity, the smaller the distance between the models of different users. Hence, this result confirms that only using the closest layers to the output leads to the highest recognition rates. For instance, by using 2 or 3 layers, FedCLAR generates a few clusters that have small or no impact on the F1-score. On the other hand, only using 1 layer generally leads to a higher number of clusters and an improvement of the recognition rate. We also observe that the clustering threshold ct also has an impact on the considered metrics. If the threshold is too low, FedCLAR excludes a too large portion of the users from clustering, negatively impacting the recognition rate. On the other hand, if ct is too high, FedCLAR would generate a lower number of clusters with users not actually similar to each other, with a negative impact on the recognition rate.

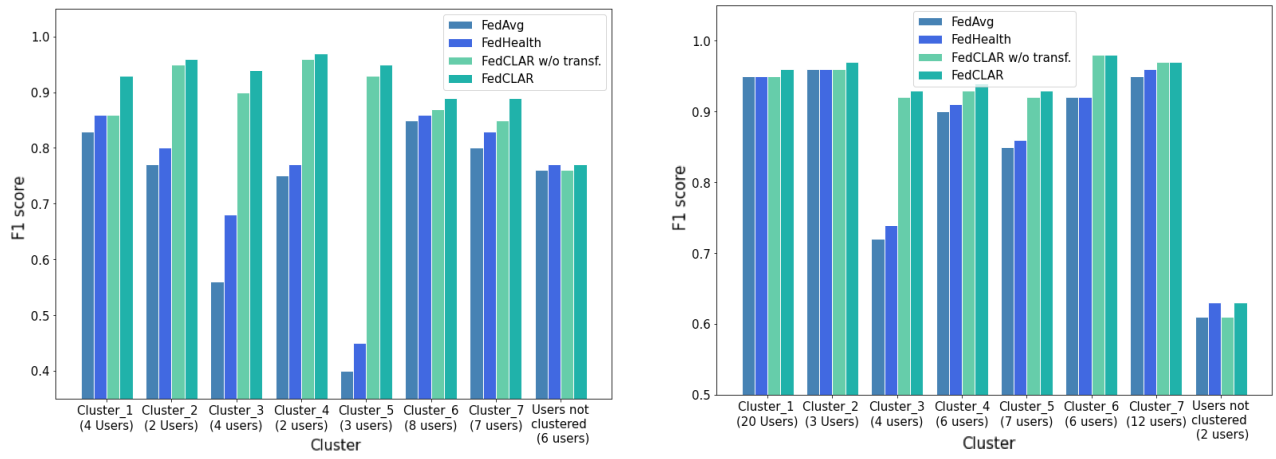
TABLE II: WISDM: Impact of hyper-parameters

l	*ct*	F1	# clusters	Users not clustered
	0.001	0.83	2	86,11%
	0,003	0.87	7	50.00%
1	**0,005**	**0.89**	**7**	**16,67%**
	0,007	0.87	5	11,11%
	0,009	0.84	4	5,56%
	0,002	0.81	3	83,33%
	0,0025	0.82	4	72,22%
2	0,003	0.81	1	55,56%
	0,0035	0.80	1	27,78%
	0,004	0.80	1	8,33%
	0,003	0.82	4	61,11%
	0,0035	0.81	2	47,22%
3	0,004	0.80	2	22.22%
	0,0045	0.80	1	16,67%
	0,005	0.80	1	5,56%

TABLE III: MobiAct: Impact of hyper-parameters

l	*ct*	F1	# clusters	Users not clustered
	0.0025	0.90	10	38.98%
	0,003	0.91	11	30.51%
1	0,005	0.92	7	15.25%
	0,010	**0.94**	**7**	**3.39%**
	0,020	0.92	3	0.0%
	0,001	0.89	4	52.54%
	0,0015	0.89	4	32.20%
2	0,002	0.90	1	11.86%
	0,0025	0.90	1	10,17%
	0,003	0.90	1	3,39%
	0,001	0.89	2	64,41%
	0,0015	0.89	4	32,20%
3	0,002	0.89	1	20.34%
	0,0025	0.89	1	13.56%
	0,003	0.90	2	5,08%

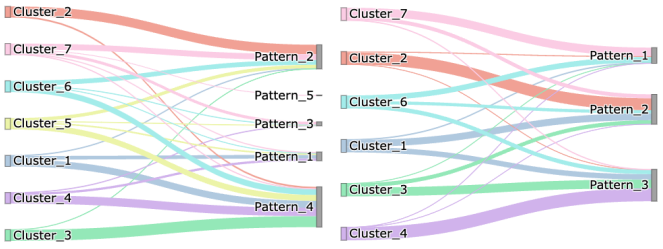
6) Evaluating the impact on non-IID data:
In the following, we show how the non-IID problem (formulated in Section III) is actually mitigated by FedCLAR on the considered datasets. First, we investigate the feature



(a) WISDM (b) MobiAct

Fig. 3: Comparison of FedCLAR with the alternative approaches cluster by cluster.

distribution skew. We expect that users grouped in the same cluster perform activities in a similar way, while users in different clusters execute activities in different ways. In order to evaluate if the clusters generated by FedCLAR have this property, from the raw sensor data of all users in each dataset we extract, for each activity, a set of patterns, each one characterising a way of performing that activity [2]. Then, we correlate the patterns with the clusters of users generated by FedCLAR. For the sake of brevity, we report a couple of examples related to the WISDM dataset in Figure 4.

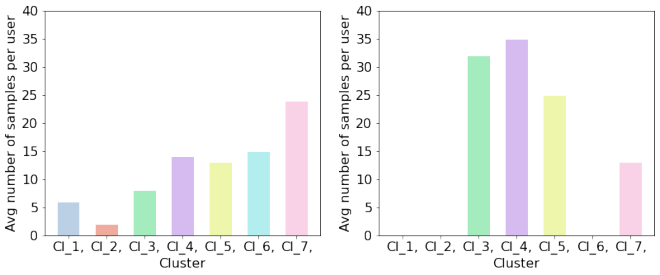


(a) Going Downstairs (b) Jogging

Fig. 4: WISDM: examples of feature distribution skew. The plot shows the correlation between clusters and activity patterns.

From this analysis, it emerges that many clusters generated by FedCLAR in WISDM exhibit a peculiar correlation with activity patterns. Hence, the non-IID problem is reduced with a positive impact on the recognition rate. For instance, considering the activities in Figure 4, the improvement in overall F1-score of FedCLAR with respect to FedAVG is $+24\%$ for *going downstairs* (from 0.48 to 0.72), while $+5\%$ for *jogging* (from 0.93 to 0.98). We observed an improvement in the F1 score for each activity in WISDM whenever there is a clear correlation between clusters and patterns.

We also noticed that the feature distribution skew does not clearly emerge in MobiAct, since the users in this dataset tend to perform activities with similar patterns. This is reflected by the results presented above: FedCLAR has a minor improvement on this dataset with respect to WISDM. The improvement of FedCLAR on MobiAct is still appreciable since, differently from WISDM, this dataset suffers from a significant label distribution skew. Hence, FedCLAR is still able to improve the recognition rate by grouping users that have similar labels distributions. Figure 5 shows this property for a couple of activities. Considering the examples in this figure, the improvement in F1-score of FedCLAR with respect to FedAVG is $+28\%$ (from 0.30 to 0.58) for *going upstairs*, while $+12\%$ for *sitting* (from 0.81 to 0.92). We observed an improvement in F1 score for each activity in MobiAct whenever there is a clear correlation between clusters and skewed label distributions.



(a) Going Upstairs (b) Sitting

Fig. 5: MobiAct: examples of labels distribution skew. The plot shows the average number of activity samples for each user in the clusters generated by FedCLAR.

[2]We normalize raw sensor data, we apply PCA for dimensionality reduction and we apply the K-Means algorithm. In order to find the optimal number of clusters for each activity, we maximize the Silhouette score.

VI. Discussion

A. Personal data protection

Among the limitations of the current version of FedCLAR, there is the potential leak of private information to a honest-but-curious service provider running the server infrastructure. It is well known that, despite only model parameters are shared with the server, some personally identifiable data could be still inferred from them. In order to mitigate this issue, FL approaches usually rely on Secure MultiParty Computation (SMC) to aggregate the local weights in a privacy-preserving fashion [34]. SMC makes it possible to hide from the service provider the mapping between each local model and the corresponding subject. Even when this type of protection is applied, FL models are exposed to several types of attacks that extract private information from the global model parameters [35]. Examples of such attacks are the reconstruction attack [36], the membership inference attack [37], and the property inference attack [38].

Among the three categories of attacks mentioned above, property inference is particularly problematic for the HAR domain and in particular for FedCLAR. First, since clusters are derived server-side, the SMC based technique cannot be applied, and the service provider could observe the relationships between local models, clusters and subjects. Moreover, a cluster may actually group users with similar sensitive conditions. For instance, suppose that FedCLAR derives a cluster composed only of subjects with Parkinson disease and a similar gait impairment. An honest-but-curious service provider may use external data of a subject with the same health condition to understand in which cluster is included. The cluster indirectly identifies all the associated subjects revealing that they suffer from the same health condition. There are several possible ways to extend FedCLAR in order to address this and other privacy attacks as it would be required for a real-world deployment. In the following, we mention some of them but a complete investigation is out of the scope of this contribution.

A promising solution, that has been recently proposed in the literature, is a federated learning architecture based on Trusted Execution Environment (TEE) [39]. In this scenario, the server-side algorithms of FedCLAR are executed within a protected environment. Indeed, code and data inside a TEE are confidential. The participating clients would transmit encrypted model weights to the service provider, which are then decrypted inside the TEE to update the global models and computing hierarchical clustering. The global model leaves the TEE encrypted. Another possible approach is to use distance-preserving homomorphic encryption to compute clusters and specialized models in a privacy-preserving fashion [40]. The drawback of this approach is that homomorphic encryption may introduce significant computational efforts. Alternatively, it would be possible to use Local Differential Privacy in order to introduce noise during the training of the local models, based on privacy preferences [41]. The major challenge in this case is finding an acceptable trade-off between privacy protection and recognition accuracy.

B. Clustering with a dynamic number of clients

In this work, we show the effectiveness of clustering within a single FL process, with a fixed number of clients. However, in FL, the global model is updated periodically. For instance, considering a mobile computing scenario, this process may happen every night, when the personal devices of the users are idle and charging. Hence, two events can occur between two updates: a) new clients join the system, b) clients that previously contributed to training the global model abandon the system. When those events occur, the clustering structure may change. In order to tackle this challenge, a possibility is that the cloud server stores every intermediate model computed during clustering (i.e., the dendograms associated with intermediate steps of hierarchical clustering). Hence, when clients join or leave the system, the server can recompute an optimal set of clusters by reversing some of the clustering steps. A similar approach was proposed in [13].

C. Evaluation on a large scale

In this work, we took advantage of the public datasets suited for this task and with the highest number of subjects. However, real-world FL scenarios may involve thousands of users, or even more. Hence, while FedCLAR exhibits promising results, they need to be confirmed on more realistic experiments on a larger scale.

A significant limitation of FedCLAR in large scale scenarios is that, at each communication round, every participating client is involved in the global model update. However, for the sake of scalability, FL methods randomly sample a limited number of clients at each communication round [4]. Hence, we will investigate a scalable solution to distribute the clustering process in multiple communication rounds.

Another significant problem related to deploying FedCLAR on a large scale is the correct choice of the hyper-parameters. In this work, we split the data of each user in 70% for training and 30% for testing and performed an exhaustive grid search. However, the hyper-parameters that proved to be effective in our experiments may not reflect the ones that are effective on a large scale. Hence, we will study the challenging problem of choosing the correct hyper-parameters in large-scale scenarios, where only a limited amount of labeled data is actually available.

VII. Conclusion and future work

In this paper, we presented FedCLAR, a novel federated clustering approach for HAR. FedCLAR combines the recent research on federated clustering with transfer learning approaches for FL-based HAR to mitigate the non-IID problem. Our results indicate that FedCLAR outperforms state-of-the-art FL solutions based on a single global model. Besides the limitations described in Section VI, a significant problem of FedCLAR is that it assumes that each client has high availability of labeled data. However, this is not realistic in

real-world HAR scenarios. Hence, we will investigate if our clustering approaches can be enhanced with self-supervised and semi-supervised learning techniques to derive reliable clusters with a limited amount of labeled data and, at the same time, to maintain a high recognition rate.

REFERENCES

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

[2] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–40, 2021.

[3] S. Ek, F. Portet, P. Lalanda, and G. Vega, "A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison," in *19th IEEE International Conference on Pervasive Computing and Communications PerCom 2021*, 2021.

[4] 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.

[5] S. Ek, F. Portet, P. Lalanda, and G. Vega, "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*, 2020, pp. 638–643.

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

[7] Z. Xiao, X. Xu, H. Xing, F. Song, X. Wang, and B. Zhao, "A federated learning system with enhanced feature extraction for human activity recognition," *Knowledge-Based Systems*, vol. 229, p. 107338, 2021.

[8] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, 2020.

[9] G. M. Weiss and J. Lockhart, "The impact of personalization on smartphone-based activity recognition," in *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*. Citeseer, 2012.

[10] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

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

[12] T. Yu, T. Li, Y. Sun, S. Nanda, V. Smith, V. Sekar, and S. Seshan, "Learning context-aware policies from multiple smart homes via federated multi-task learning," in *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 2020, pp. 104–115.

[13] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, 2020.

[14] C. Briggs, Z. Fan, and P. Andras, "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, 2020, pp. 1–9.

[15] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[16] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Trans. Syst. Man Cybern. C:App. Rev.*, vol. 42, no. 6, pp. 790–808, 2012.

[17] T. Plötz and Y. Guan, "Deep learning for human activity recognition in mobile computing," *Computer*, vol. 51, no. 5, pp. 50–59, 2018.

[18] U. Bakar, H. Ghayvat, S. Hasanm, and S. C. Mukhopadhyay, "Activity and anomaly detection in smart home: A survey," *Next Generation Sensors and Systems*, pp. 191–220, 2016.

[19] J. Wang, V. W. Zheng, Y. Chen, and M. Huang, "Deep transfer learning for cross-domain activity recognition," in *proceedings of the 3rd International Conference on Crowd Science and Engineering*, 2018, pp. 1–8.

[20] M. Z. Zadeh, A. Ramesh Babu, A. Jaiswal, M. Kyrarini, and F. Makedon, "Self-supervised human activity recognition by augmenting generative adversarial networks," in *The 14th PErvasive Technologies Related to Assistive Environments Conference*, 2021, pp. 171–176.

[21] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *IEEE pervasive computing*, vol. 9, no. 1, pp. 48–53, 2009.

[22] D. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: A survey," *Knowl. Inf. Sys.*, vol. 36, no. 3, pp. 537–556, 2013.

[23] J. Ye, S. Dobson, and F. Zambonelli, "Lifelong learning in sensor-based human activity recognition," *IEEE Pervasive Computing*, vol. 18, no. 3, pp. 49–58, 2019.

[24] L. Lyu, X. He, Y. W. Law, and M. Palaniswami, "Privacy-preserving collaborative deep learning with application to human activity recognition," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1219–1228.

[25] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[26] A. Saeed, F. D. Salim, T. Ozcelebi, and J. Lukkien, "Federated self-supervised learning of multisensor representations for embedded intelligence," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1030–1040, 2020.

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

[28] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent iot applications: A cloud-edge based framework," *IEEE Computer Graphics and Applications*, 2020.

[29] 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.

[30] Z. Chen, P. Tian, W. Liao, and W. Yu, "Zero knowledge clustering based adversarial mitigation in heterogeneous federated learning," *IEEE Trans. Netw. Sci. Eng.*, 2020.

[31] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[32] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis, "The mobiact dataset: Recognition of activities of daily living using smartphones." in *ICT4AgeingWell*, 2016, pp. 143–151.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] W. Mou, C. Fu, Y. Lei, and C. Hu, "A verifiable federated learning scheme based on secure multi-party computation," in *International Conference on Wireless Algorithms, Systems, and Applications*. Springer, 2021, pp. 198–209.

[35] D. Zhang, X. Chen, D. Wang, and J. Shi, "A survey on collaborative deep learning and privacy-preserving," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2018, pp. 652–658.

[36] L. Zhu and S. Han, "Deep leakage from gradients," in *Federated learning*. Springer, 2020, pp. 17–31.

[37] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[38] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 691–706.

[39] A. Huang, Y. Liu, T. Chen, Y. Zhou, Q. Sun, H. Chai, and Q. Yang, "Starfl: Hybrid federated learning architecture for smart urban computing," *ACM Trans. Intell. Syst. Technol.*, vol. 12, no. 4, pp. 1–23, 2021.

[40] H. Wang, A. Li, B. Shen, Y. Sun, and H. Wang, "Federated multi-view spectral clustering," *IEEE Access*, vol. 8, pp. 202 249–202 259, 2020.

[41] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K.-Y. Lam, "Local differential privacy-based federated learning for internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8836–8853, 2020.