FedDL: Federated Learning via Dynamic Layer Sharing for Human Activity Recognition

Linlin Tu
Michigan State University
East Lansing, MI, USA
tulinlin@msu.edu

Xiaomin Ouyang
The Chinese University of Hong Kong
Hong Kong SAR, China
xmouyang@link.cuhk.edu.hk

Jiayu Zhou
Michigan State University
East Lansing, MI, USA
jiayuz@egr.msu.edu

Yuze He
The Chinese University of Hong Kong
Hong Kong SAR, China
yzhh@link.cuhk.edu.hk

Guoliang Xing*
The Chinese University of Hong Kong
Hong Kong SAR, China
glxing@cuhk.edu.hk

ABSTRACT

Deep learning has been increasingly applied to improve human activity recognition (HAR) accuracy and reduce the human efforts of handcrafted feature extractions. Federated Learning (FL) is an emerging learning paradigm that enables the collaborative learning of a global model without exposing users' raw data. However, existing FL approaches yield unsatisfactory HAR performance as they fail to dynamically aggregate models according to the statistical diversity of users' data. In this paper, we propose FedDL, a novel federated learning system for HAR that can capture the underlying user relationships and apply them to learn personalized models for different users dynamically. Specifically, we design a dynamic layer sharing scheme that learns the similarity among users' model weights to form the sharing structure and merges models accordingly in an iterative, bottom-up layer-wise manner. FedDL merges local models based on the dynamic sharing scheme, significantly speeding up the convergence while maintaining high accuracy. We have implemented FedDL and evaluated using a new data set we collected using LiDAR and four public real-world datasets involving 178 users in total. The results show that FedDL outperforms several state-of-the-art FL paradigms in terms of model accuracy (by more than 15%), converging rate (by more than 70%), and communication overhead (about 30% reduction). Moreover, the testing results on the datasets of different scales show that FedDL has high scalability and hence can be deployed for large-scale real-world applications.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Computing methodologies** → **Transfer learning**.

KEYWORDS

Human activity recognition, Federated learning, Multi-task learning, Federated Learning Personalization

ACM Reference Format:
Linlin Tu, Xiaomin Ouyang, Jiayu Zhou, Yuze He, and Guoliang Xing. 2021. FedDL: Federated Learning via Dynamic Layer Sharing for Human Activity Recognition. In *The 19th ACM Conference on Embedded Networked Sensor Systems (SenSys '21), November 15–17, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3485730.3485946

*Corresponding author

SenSys '21, November 15–17, 2021, Coimbra, Portugal

ACM ISBN 978-1-4503-9097-2/21/11…$15.00
https://doi.org/10.1145/3485730.3485946

1 INTRODUCTION

Human activity recognition (HAR) is a key enabling technology for a wide range of applications, including smart home, health surveillance, and medical assistance [30, 31, 67]. For instance, it has been shown that longitudinal monitoring of daily routine activities, such as indoor/outdoor time, meals with/without family, and sleeping, can help to detect early onsets of Alzheimer's Disease in aged population [38, 58]. Similarly, smart home systems can conserve home energy consumption and improve residents' comfort/safety by recognizing complex home activities (e.g., eating, taking a shower, washing dishes, etc.) [18, 29].

Deep learning has recently been applied to HAR thanks to its better generalization and the ability of automatic feature extraction with less human effort [24, 25, 57]. However, several major challenges have not been addressed. The data collected from each user is usually unbalanced and sparse. Activities such as taking a shower, shopping, and biking, usually take place in a relatively low frequency. Applying deep learning to sparse and unbalanced data is likely to result in severe under-sampling artifacts. Training a global model for HAR in the cloud in a centralized manner may reduce the effect of data sparsity. However, the sensing data for HAR is often privacy-sensitive and hence cannot be shared or uploaded [15, 55].

Federated Learning (FL) is an emerging technique used to collaboratively learn a global model, such as by computing an average aggregation of local models, without exposing users' raw data [11, 45, 46, 51, 62]. Existing FL paradigms learn a single global model that however fails to capture the statistical diversity of users' data. Such statistical diversity of users' data not only leads to significant convergence delay but also poor model accuracy [5, 12, 26]. Several FL approaches have been proposed to address this problem by

learning personalized models which capture both general and personal features of users [7, 14, 17, 44]. In [7], users share only lower layers of their models and leave upper layers user-specific to retain personal features. However, this approach assumes a pre-defined number of model layers shared among users, which is determined by empirical perception of user data distributions and their correlations. As a result, it suffers poor performance when the users' data distributions are highly dynamic and time-varying [59]. The post-personalized FL approach is proposed to further fine-tune the global federated model on the nodes' local data [20, 33]. However, the performance of such an approach is largely influenced by the accuracy of the global model.

In this paper, we propose FedDL - a novel federated learning system for HAR that can capture the underlying user relationships and apply them to learn personalized models for different users dynamically. Our design is motivated by two key observations. First, despite statistically diverse, users' data often exhibits significant similarity due to inherent behavior habits and biological features (e.g., gender, height, weight, etc.), or environments [37, 61, 69]. FedDL exploits such similarity among users and facilitates the collaboration among them to improve the model accuracy. Second, the degree of similarity among users' deep models reduces from the bottom up [41, 50, 68], motivating us to learn the layer-wise sharing structure of users' deep models dynamically to improve the model accuracy and efficiency. Specifically, we design a dynamic layer sharing scheme that captures the similarity among users' model weights to learn the sharing structure and merges models accordingly in an iterative, bottom-up layer-wise manner. FedDL not only improves the model accuracy but also reduces the overhead of communication, as each model merging at the server only involves the parameters of users' lower model layers. To evaluate the performance of FedDL, we collect a new LiDAR dataset in real-world settings, which contains high dimensional, sparse point clouds for recognizing various user activities. Our results based on our new dataset and four additional public real-world datasets involving a total of 178 users show that FedDL outperforms three state-of-art FL paradigms in terms of overall accuracy (e.g., by 16.67%, 19.51%, 30.67%, respectively). Besides, FedDL converges the fastest and has a relatively low communication overhead. For example, FedDL saves more than 50% communication overhead compared with the federated average approach when a large number of users are involved. Moreover, the results on the datasets of different sizes suggest that FedDL yields satisfactory scalability.

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 presents a motivation study that sheds key insights into the design of FedDL. Section 4 and 5 describe the system overview and design in detail. In section 6, we elaborate on the evaluation of FedDL and discuss the experimental results. Lastly, we conclude the paper in section 7.

2 RELATED WORK

Deep learning for HAR. Deep learning has been applied to improve the accuracy of human activity recognition and eliminate the human efforts of handcrafted feature extractions [9, 35, 54]. However, since many daily events, like taking a shower, shopping, and biking, only occur occasionally, one user usually has limited and unbalanced training samples, which can cause overfitting in training deep learning models [16, 22]. Data augmentation techniques may address the issue by expanding the local datasets. However, they will fail to discover the new activities in HAR when users' patterns of activities change largely. For instance, users without exercise habit start to do sports, which is not the situation that data augmentation works. As data augmentation cannot produce the data for a previous unknown activity, the model trained with data augmentation will fail to discover the new activity. Training a global model for HAR at the server is proposed to reduce the effect of data sparsity [71]. However, centralized methods require uploading users' sensing data to the cloud, leading to risk of privacy breach.

Federated learning (FL) [28, 66] is an emerging learning paradigm that only requires users to upload their model weights for collaborative learning, avoiding sharing user's raw data during the learning process. A typical FL approach named FedAvg [11, 28] averages all models from users to learn a single global model, which proves to suffer significant accuracy degradation under heterogeneous data distributions of users [39, 70]. Recently several personalized FL approaches are proposed to address this issue. Dinh et al. add a regularized term to the loss function of each user's local model during the FL process to reduce the distance between the local and global models (average of all models) [17, 20, 33]. However, the accuracy of models learned in this approach can be largely influenced by the diversity of users. Moreover, other studies [14, 44] tend to introduce a post-training procedure that personalizes the learned global model on each user's local data. However, careful fine-tuning is required in this approach to balance the local and global models, which varies among different applications and hence is hard to generalize. Compared with existing personalized FL approaches, FedDL is able to learn users' relationships during the FL process and utilize them to dynamically aggregate the local models in a layer-wise manner, which is applicable to different applications with highly diverse data distributions.

FL personalization via model sharing. In the FL approaches proposed in [7, 13], the lower layers between all users are shared, while several upper layers are user-specific. This design is motivated by the observation that the lower layers capture more general features, and hence can be shared across multiple tasks, whereas the top layers capture features at a higher level of abstraction and hence are more user-specific [68]. The above methods have been extended and applied to multi-task deep learning [42, 48], where the goal is to learn multiple different models. However, these multi-task methods rely on a pre-defined structure for model sharing. As network architectures become deep and the user relationship becomes more complex in large-scale HAR applications, finding the right level of feature sharing across local models through hand-crafted network branches is impractical. Moreover, most multi-task deep learning methods [43, 53] are centralized and do not address the communication efficiency of the learning process. To reduce the communication overhead of FL (especially for transferring deep learning models), previous solutions mainly focused on the techniques for model quantization [36, 60] or model compression [23]. FedDL reduces the communication overhead through the dynamic layer-wise sharing scheme, as each model merging at the server only involves the parameters of users' lower model layers, which is

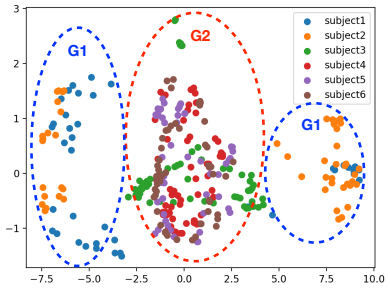

Figure 1: The data of "typing" from the HARBox dataset after reducing dimension to 2D using PCA. There exists a clear group relationship among different subjects' data.

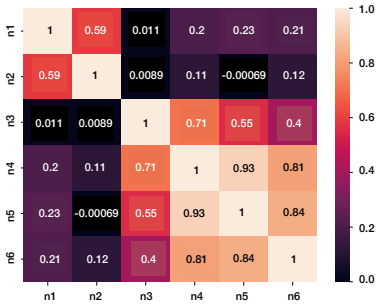

Figure 2: Correlation matrix of 6 users' HARBOX data. Each number is the Pearson correlation coefficient (PCC), measuring the linear correlation between two users' data. It is obvious there are two groups, (n_1, n_2) and (n_3, n_4, n_5, n_6). However, the users within each group are of different degrees of similarity.

orthogonal to the model quantization or compression techniques. In a recent work [52], the authors show significant similarity exists among users in a number of real-world datasets, which is similar to our finding in Section 3. However, in [52], the clustering structure is formulated as part of the learning objective, and the local models are required to share all the layers in their multi-task learning framework. On the contrary, FedDL dynamically captures the users' relationship while learning different models for users with a partial sharing structure, which leads to better model accuracy and lower communication overhead.

3 A MOTIVATION STUDY

In this section, we use an open real-world dataset, HARBox [1], to motivate the approach of FedDL in two aspects. First, there often exists underlying similarity amongst users' patterns of activities due to their habits of behavior or environments [37, 52, 61, 69], which can be utilized to improve the learned model accuracy by facilitating collaborations among similar users. Second, the degree of similarity among users' deep models reduces from the bottom up [41, 50, 68], which suggests that we may exploit such similarity of models and aggregate them in an iterative, layer-wise manner, rather than aggregating whole models. We show that such an approach improves the model accuracy and reduces communication overhead between users and the server since only partial models need to be transmitted.

The HARBox dataset is collected in real-world federated settings [52]. The 9-axis IMU data from 121 users' smartphones is recorded when the users conduct five activities of daily life (ADL), including walking, hopping, phone calls, waving, and typing. To visualize the data distribution, we plot the data of "typing" from 6 users in the HARBox dataset after reducing the dimension of features to 2D using Principal Component Analysis. As shown in Fig. 1, there exists a clear grouping relationship among the 6 subjects' data, with $G_1 = (n_1, n_2)$ and $G_2 = (n_3, n_4, n_5, n_6)$. We note that such similarity among users is also reported on other HAR datasets [6, 21, 32].

Our goal is to exploit the similarity among users' data to personalize their models. A natural idea is to share some model layers between similar users [7, 44]. We now explore different model sharing schemes for each user group and their impact on the shared model accuracy. Fig. 3 shows three sharing schemes of deep learning models for a specific user group. The "all-sharing" scheme shares all layers of the users' models within each group. The K-sharing scheme shares only the lowest K layers of the users' models, where the number of shared lower layers K is usually empirically pre-set and fixed during the learning process. This baseline is similar to several existing FL personalization methods [7, 42]. In the experiments, we set $K = 3$ for the two groups. However, we will show that the K-sharing scheme cannot accurately capture the complicated relationship among users' data distribution. Some users are closely related enough to share more than K layers, while others with a large difference in their data distributions may benefit from sharing fewer than K layers. We visualize the relationship among data of 6 users from the HARBOX datasets through a correlation matrix by computing the Pearson correlation coefficients (PCC) between each pair of users' data. As shown in Fig. 2, we see there are two groups, $G_1 = (n_1, n_2)$ and $G_2 = (n_3, n_4, n_5, n_6)$. However, the users within each group are of different degrees of similarity. For instance, n_3 is less related with the other users in G_2 (the statistically independent variables have correlation coefficients close to zero). This observation inspires a dynamic sharing structure, where only users with similar data distributions should collaborate in learning and users who are more closely related to each other will share more layers of their models. Based on this idea, we design a new scheme "layer-wise sharing" shown in Fig. 3, which is derived according to the correlation matrix of the six users (shown in Fig. 2) with closer-related users sharing more model layers. Specifically, n_4, n_5 and n_6 should share more layers than n_3, since they are more closely related to each other than n_3. Shown in Fig. 3, n_4, n_5 and n_6 share their lower 3 layers in this example, while n_3 only shares the lower two layers with them.

We also implement a baseline "global" method where all the six users share the same global model by averaging all their layers [45], and compare its performance on HAR with three sharing schemes (shown in Fig. 3): all-sharing, K-sharing, and the layer-wise sharing structure derived from the correlation matrix in Fig. 2. Fig. 4 presents the model accuracy performance of n_3 when trained under different sharing schemes. We see that the model based on the layer-wise sharing structure gives the highest testing accuracy.

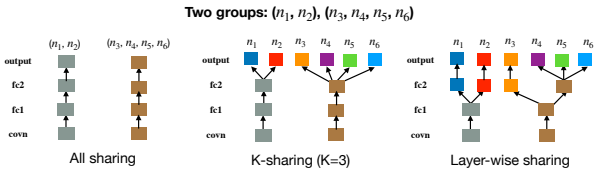

Figure 3: Illustration of three sharing schemes for a group.

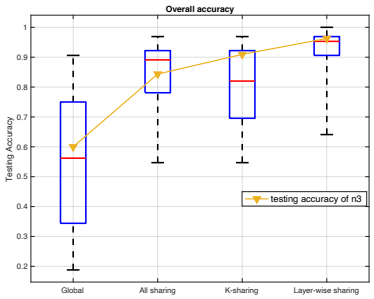

Figure 4: Illustration of the performance of federated learning under four sharing schemes. Layer-wise sharing scheme outperforms other sharing schemes in overall accuracy.

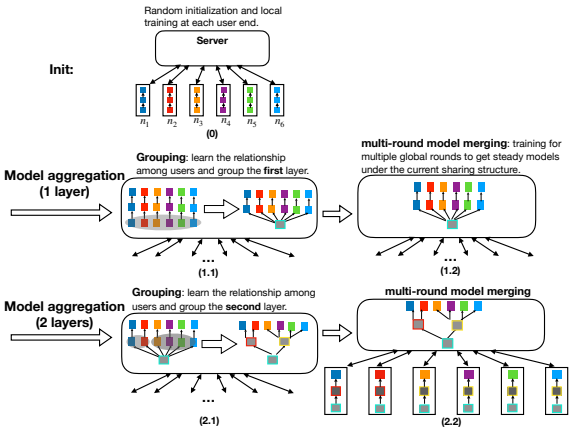

Figure 5: Illustration of the dynamic and hierarchical federated learning framework of FedDL when learning 3-layer models for 6 users.

Motivated by this result, we attempt to generate the layer-wise sharing structure from user relationships to improve the model accuracy. However, the correlation matrix of users' data in Fig. 2 is global information that cannot be obtained on the server without accessing the data of users. Thus, we design a dynamic sharing scheme to learn the similarity of users' model weights and generate the layer-wise model sharing structure accordingly during FL to improve the model accuracy. Specifically, FedDL learns the grouping relationship of the local models and then merges only the lower layers of models in a bottom-up layer-wise manner. In Section 5, we will elaborate on the proposed dynamic sharing scheme.

In addition to the possible improvement in the training accuracy and efficiency of FL, another key advantage of our dynamic sharing scheme is that it reduces communication overhead as it is unnecessary for users to upload their user-specific layers to the server for model merging during the distributed learning process. We discuss communication efficiency in Section 5.4.

4 SYSTEM OVERVIEW

This section presents an overview of the proposed Federated Learning via Dyanmic Layer Sharing (FedDL). FedDL aims to enable accurate daily activity recognition through communication-efficient deep FL, based on the underlying affinities among users' activity patterns. In this section, we first briefly introduce the application scenarios of FedDL, and then describe its system architecture.

FedDL is designed for monitoring a wide range of daily activities using sensors built in wearables or deployed in natural living environments. Representative applications include healthcare monitoring and smart home systems [18, 29]. These systems are usually designed to recognize a wide range of activities, like medicine taking, indoor/outdoor activities, and meal events, using ambient sensors and body-worn sensors [10, 56, 63]. However, since many events only occur occasionally, users tend to have limited and unbalanced training samples, which can cause overfitting in training deep learning models. Moreover, the sensing data for HAR is mostly privacy-sensitive and hence cannot be shared or uploaded. To address this issue, FedDL adopts the FL paradigm, utilizing a central server to collect local models and aggregate them, while avoiding the exposure of users' raw data during the learning process. However, models learned by FL may deliver unsatisfactory performance on recognition of each user's activities, due to the statistical diversity of users' data. To improve the model accuracy, FedDL learns the underlying relationship among users dynamically and merges the local models partially based on the degree of similarity among users in a layer-wise manner. Since the users' data distribution may change over time, FedDL will periodically update the layer-wise sharing structure and models.

FedDL features a dynamic and hierarchical FL framework that improves accuracy and communication efficiency by capturing the intrinsic relationship among users and applying it to learn layer-wise personalized models for different users. Fig. 5 depicts the hierarchical training procedure of FedDL. First of all, the local model of each user is optionally initialized randomly or from a pre-trained model. Then FedDL performs **model grouping** and **model merging** in a bottom-up layer-wise manner. Specifically, the server groups users based on the model affinities obtained from models' testing results on a common sample set using Kullback–Leibler divergence (KLD) (shown in Fig. 6(3.1)). It then performs model-merging to obtain stable models with the lower layers shared within each group. The merging process is implemented by calculating a weighted average of local models' parameters at the server over multiple rounds. Each model merging round involves 4 steps, as shown in Fig. 6. Users perform multiple epochs of local training and then upload local models to the server. The server computes the weighted average of local models based on grouping results. It then generates further personalized models through the weighted average of local models and their corresponding averaged models.

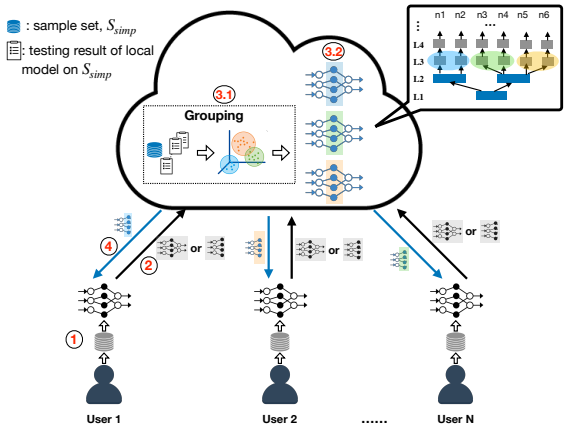

Figure 6: The system architecture of FedDL. Each grouping / model-merging round mainly consists of 4 steps.

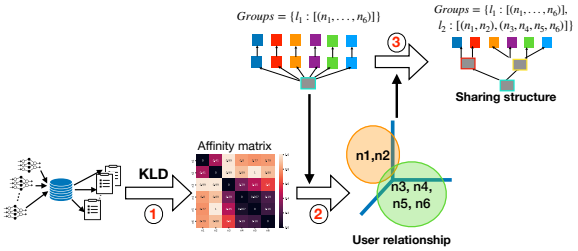

Figure 7: The procedure of model-affinity-based grouping. It consists of three steps: 1. Calculate the affinity matrix; 2. Group users based on the affinity matrix and previous grouping results; 3. Update the layer-wise sharing structure.

Finally, the server transmits personalized models back to users for local training. This model grouping and model merging process repeats till reaching the output layer (i.e., the top layer), as FedDL leaves the output layer user-specific without sharing between users.

It is challenging to learn the intrinsic relationship among users without accessing the users' data. FedDL learns the relationship among users based on their local models, and generates the sharing structure by grouping the lower model layers of closely related users, and keeps exploring the grouping relationship layer by layer within each group from the bottom up till reaching the top layer. Section 5.1 describes the model affinity-based grouping in detail. Moreover, based on the iteratively learned sharing structure, FedDL performs layer-wise model merging after each model grouping process to obtain stable models under the sharing structure. Section 5.2 presents the design of intra-group layer-wise model merging. FedDL generates shared models in a bottom-up layer-wise manner using a greedy algorithm. Section 5.3 describes the detail of bottom-up layer-wise model aggregation.

The layer-wise model aggregation of FedDL improves the model accuracy through dynamic sharing within groups and reduces communication overhead by only transmitting the merged layers rather than entire models. As shown in Fig. 6, except for the grouping iteration when whole local models are uploaded to the server, most of the global communication involves only their lower layers, which significantly reduces the communication overhead during the FL process. Section 5.4 discusses the communication efficiency of FedDL in detail.

5 DYNAMIC LAYER-WISE FEDERATED DEEP LEARNING FRAMEWORK

FedDL is a federated learning framework that learns personalized deep models for users with limited or unbalanced data in HAR applications. Specifically, FedDL learns the relationship among users, generates the dynamic sharing structure for models' lower layers based on the user relationship, and merges the models according to the sharing structure iteratively. In Section 5.1 and 5.2, we presents how to group users using their deep models and how to dynamically merge different layers of models for users in the same group, respectively. In Section 5.3, we describe the procedure of the bottom-up layer-wise model aggregation. Finally, we introduce the design on communication efficiency in Section 5.4.

5.1 Model Affinity-based User Grouping

FedDL learns the underlying relationship of users based on their model affinities. Specifically, FedDL measures model affinities using Kullback–Leibler divergence (KLD), which estimates how one probability distribution is different from the reference one and is recently used for knowledge distillation of deep learning models [2, 4]. As demonstrated in [19, 27], element-wise weight distances (e.g., L1/L2 norms) have severe limitations in modeling affinities of deep models since the neurons of each layer in hierarchical models are permutable. Besides, it is computational inefficient to measure the norm distance of high-dimensional weights for complex hierarchical models. Therefore, instead of directly analyzing the weight matrices, FedDL tests all local models on a reference distribution in the form of a common sample set, and then measures the model affinities using the KLD of the different model outputs, as shown in Fig. 7. Specifically, the KLD for a pair of models, $\boldsymbol{w}_p$ and $\boldsymbol{w}_q$, is calculated as follows:

$$D_{kl}(\boldsymbol{w}_p, \boldsymbol{w}_q) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(\delta_{p,i}\log\frac{\delta_{p,i}}{\delta_{ref,i}} + \delta_{q,i}\log\frac{\delta_{q,i}}{\delta_{ref,i}}) \quad (1)$$

$$\delta_{p,i} = \delta(\boldsymbol{w}_p, \boldsymbol{x}_i) \quad (2)$$

$$\delta_{ref,i} = \frac{1}{2}[\delta(\boldsymbol{w}_p, \boldsymbol{x}_i) + \delta(\boldsymbol{w}_q, \boldsymbol{x}_i)] \quad (3)$$

where $\delta_{q,i}$ denotes the softmax outputs of the model $\boldsymbol{w}_q$ on the ith record, $\boldsymbol{x}_i$, of the common sample set. $\delta_{ref,i}$ is the reference distribution. We take the average of the two models' outputs as the reference distribution and measure how these two models are different from the reference, where a lower D_{kl} value indicates a higher model affinity. Instead of directly using the KLD of P over Q, we adopt this symmetric metric for similarity measurement, which is more suitable for user grouping.

In the next, FedDL performs grouping at the l-th layer based on the model affinity and previous grouping results. Specifically, FedDL maintains an affinity matrix, M_a with $a_{(p,q)} = D_{kl}(\boldsymbol{w}_p, \boldsymbol{w}_q)$, and keeps the grouping results of lower l layers in the dictionary,

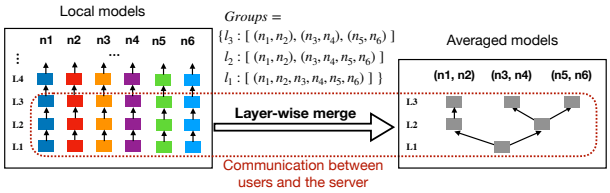

Figure 8: Illustration of the layer-wise model merging based on the grouping results, *Groups*. Only lower 3 layers of models are transferred between six users and the server for model merging.

Groups, to represent the dynamic sharing structure, as follows:

$$\begin{aligned} Groups = \{\ & 1 : [G_{1,1}, G_{1,2}, ..., G_{1,k_1}] \\ & 2 : [G_{2,1}, G_{2,2}, ..., G_{2,k_2}] \\ & ... \\ & l : [G_{l,1}, G_{l,2}, ..., G_{l,k_l}]\ \} \end{aligned}$$

where *Groups* keeps the layer index as the key and a list of groups at this layer as the value, respectively. $G_{l,i}$ denotes the i-th group for the aggregation of the l-th layer. k_l is the number of groups at the l-th layer.

With the affinity matrix, M_a, and previous grouping results *Groups*, FedDL groups the users at the server as shown in Fig. 7. Specifically, the l-th round of grouping operation only happens within groups that are obtained from the previous grouping round ($G_{l-1,k}$). To group users within $G_{l-1,k}$, FedDL checks the affinity between each pair of users, i and j ($i, j \in G_{l-1,k}$), and compares it with the threshold, θ_G, to decide if their models are related enough to be grouped together. We take the average of the affinities between users in $G_{l-1,k}$ as the adaptive threshold θ_G for the grouping within this group. It is noted that two less-related users may be grouped together as long as they are closely related to the same user. To differentiate the degree that users are related to their group, we consider not only the group members (m_h) but also their corresponding frequency ($freq_{m_h}$) as shown in Equation 4. $freq_{m_h}$ is the times the user being accessed during the procedure of grouping. A higher $freq_{m_h}$ indicates that the group member, m_h, is closely related to more users within the group. This information will be utilized to improve the accuracy of the model merging.

$$G_{l,i} = [(m_1, freq_{m_1}), (m_2, freq_{m_2}), ...(m_h, freq_{m_h})]. \quad (4)$$

Based on the group relationship among users, FedDL updates the layer-wise sharing structure by sharing one upper layer of users' models within each group, as shown in Fig. 7. FedDL performs the grouping operation periodically till the output layer. Moreover, we can stop the grouping operation in FedDL earlier, when the number of groups at a layer equals the number of users, i.e. $k_l = N$.

5.2 Intra-group Layer-wise Model Merging

Based on the grouping results, *Groups*, FedDL merges the local models in a layer-wise manner. Fig. 8 illustrates the layer-wise model sharing at the server. Based on the grouping results of the lower 3 layers, the local models from users in the same group are merged layer by layer, as follows:

$$\boldsymbol{W}_{G_{l,k}} = \sum_{i \in G_{l,k}} \mu_i \boldsymbol{W}_{i,l} \quad (5)$$

$$\mu_i = \frac{freq_i}{\sum_{j \in G_{l,k}} freq_j} \quad (6)$$

where $\boldsymbol{W}_{G_{l,k}}$ is the weights shared by the users in $G_{l,k}$, the k-th group at l-th layer. $\boldsymbol{W}_{G_{l,k}}$ is a weighted average of all the group members' layer weights, $\boldsymbol{W}_{i,l}$. The weighted average coefficient, μ_i, of each group member is calculated based on the $freq_i$, which indicates how close the member is tied to the group. As a result, the models with higher $freq$ will contribute more to the group model.

After merging the layers of the models into shared models, the server further personalizes the shared models for each user by aligning each local model with its corresponding group model as follows,

$$\boldsymbol{W}_i' = (1 - \lambda_i)\boldsymbol{W}_i + \lambda_i \boldsymbol{W}_{G_{l,k}} \quad (7)$$

$$\lambda_i = min(1, \frac{\mu_i}{1/sizeof(G_{l,k})}) \quad (8)$$

where $i \in G_{l,k}$. λ_i indicates, from the user's stand, how closely local model $\boldsymbol{W}_i$ is related to the group model, $\boldsymbol{W}_{G_{l,k}}$. This alignment makes the models trained using FedDL robust to boundary cases, where the least related users are still included in a group (i.e., with the smallest μ_i). These users are likely to become a separate group in another training process. For instance, at a certain layer, three users are grouped as $\{1 : 0.5, 2 : 0.49, 3 : 0.01\}$, and the training process produces the grouping result $\{1 : 0.5, 2 : 0.5\}$ and $\{3 : 1\}$ at the same layer. Without alignment, the models of user 3 obtained from the two training processes are significantly different with $\boldsymbol{W}_3' = \boldsymbol{W}_G$ and $\boldsymbol{W}_3' = \boldsymbol{W}_3$ respectively. However, after the alignment between the shared models and users' local models, the models of user 3 under these two grouping results become similar with $\boldsymbol{W}_3' = 0.03\boldsymbol{W}_G + 0.97\boldsymbol{W}_3$ and $\boldsymbol{W}_3' = \boldsymbol{W}_3$ respectively.

As illustrated in Fig. 8, in this model-merging round, we get three shared models: $[\ \boldsymbol{W}_{G_{1,1}}, \boldsymbol{W}_{G_{2,1}}, \boldsymbol{W}_{G_{3,1}}\ ]$, $[\ \boldsymbol{W}_{G_{1,1}}, \boldsymbol{W}_{G_{2,2}}, \boldsymbol{W}_{G_{3,2}}\ ]$ and $[\ \boldsymbol{W}_{G_{1,1}}, \boldsymbol{W}_{G_{2,2}}, \boldsymbol{W}_{G_{3,3}}\ ]$ shared within the three groups, $(n1, n2)$, $(n3, n4)$ and $(n5, n6)$ respectively. Finally, the three shared models are aligned with their corresponding users' local models. For example, the second shared model will be aligned with the local models of n_3 and n_4 and sent to them, respectively. Moreover, only the lower layers of models are necessarily transferred between server and users during the model-merging iterations, which will significantly reduce the overall communication overhead during the FL process.

5.3 Bottom-up Layer-wise Model Aggregation

At the core of FedDL is the multi-round greedy model aggregation in a bottom-up layer-wise fashion. The learning procedures for the i-th user and the server in FedDL are presented in Algorithm 1 and Algorithm 2, respectively.

Consider a situation where there are N users. Initially, all the users start with the same neural network model and initialize it randomly or from a pre-trained model. After users perform multiple epochs (denoted as R) of local updates, the server will receive the latest N local models from all the users. The model aggregation operation of the server starts from the bottom layer, l_1. It will first

group the N branches into k_1 groups where $k_1 \leq N$. After that, FedDL will greedily perform the bottom-up model aggregation within their groups. We note that finding the optimal sharing structure is combinatorial prohibitive. A brute-force method would need to train and test all the $((C_N^N)^{L-1}$ possible structures for finding the optimal aggregation scheme for N users with L-layer models. Our approach is more efficient since it only takes $O(N \log N * L)$ time. For each round of model grouping, it takes $O(N \log N)$ time, and takes a total of $O(N \log N) * (L-1)$ time at the worse case to form the sharing structure.

For L-layer deep models, FedDL will perform L rounds of user grouping with each grouping round followed by *intvl* rounds of model merging, as shown in Fig.5. At each grouping round, FedDL learns the affinity relationship of local models and groups one upper layer of the users' models into groups. After that, FedDL performs multiple rounds of model merging within each group according to the current sharing structure. It is noted that the interval between grouping rounds, *intvl*, decreases with a decay rate, λ. When more layers of local models are merged according to their layer-wise similarity, the divergence among local models reduces gradually. Therefore it takes fewer global training rounds for the shared models to converge [41, 50]. Specifically, the procedure of model aggregation for the lower l layers is as follow:

(1) **Grouping round** (line 5-9 in Algorithm 2): the users send their complete models to the server. The server groups the users based on current model affinities and the grouping results from $l-1$th iteration, $Groups[l-1]$. It was noted that the grouping operation at lth iteration only happens within each group obtained from $(l-1)$th iteration, i.e., the users being separated into different groups in the first $l-1$ rounds no longer share their upper layers. Moreover, the grouping result will be added to *Groups*, where the grouping results of all the lower layers are kept for the model merging process.

(2) **Model-merging round** (line 10-12 in Algorithm 2): After the model grouping, FedDL performs *intvl* rounds of layer-wise model merging within each group. For each model-merging round, the clients perform R epochs of local training and then upload the lower l layers of their local models to the server. Upon receiving all local models, the server weighted averages the lower l layers of local models' within each group based on the grouping structure, *Groups*, and then aligns each local model with its corresponding group model to generate the shared model for each user. At the end, the server sends the shared models to their corresponding clients. It is noted that, for global communication, only the lower l layers of models are transferred between users and the server, which makes the FedDL very communication-efficient.

The grouping operation stops at the layer before the output layer. As a result, the higher layers of each model will be user-specific, while the lower shared layers will ensure generality across similar users. Moreover, the grouping structure and the models of FedDL will be updated periodically with continuously collected data.

Fig. 5 shows the procedure of learning a 3-layer model for 6 users. As shown in Fig. 5(1.1), after grouping based on the initial local training models, all the users are grouped together for model aggregation. As a result, the first layers of their models are merged as the group model. After that, FedDL performs the model aggregation operation for the lower two layers within the groups obtained from the previous round, i.e., $\{n_1{}_6\}$, as shown in Fig. 5(2.1) and (2.2). The server groups the users into two groups, $\{n_1, n_2\}$ and $\{n_3, n_4, n_5, n_6\}$ and updates the sharing structure with parameters of the second layer shared within each group. The dynamic sharing structure is finalized after the 2-round model aggregation, and FedDL keeps the output layer user-specific.

Algorithm 1 FedDL-CLient(i)

Input: $\mathcal{D}_i$, T, R, *intvl*, λ.

1: Initialize $\boldsymbol{W}_{i,0}^{(0)}$ at random.
2: **for** $t = 0$ to $T-1$ **do** ▷ Global communication rounds
3: **for** $r = 1$ to R **do** ▷ Local training rounds
4: $\boldsymbol{W}_{i,r}^{(t)} \leftarrow \mathbf{SGD}(\boldsymbol{W}_{i,r-1}^{(t)}, \eta)$
5: **end for**
6: Check if it is a new round of grouping at server, and send
7: $\boldsymbol{W}_{i,R}^{(t)}$ or $\boldsymbol{W}_{i,R}^{(t)}[0 : len(\boldsymbol{W}_s)]$ to the server.
8: Receive $\boldsymbol{W}_s$ from servers.
9: $\boldsymbol{W}_{i,0}^{(t+1)} = [\boldsymbol{W}_s,\ \boldsymbol{W}_{i,R}^{(t)}[len(\boldsymbol{W}_s) : end]\ ]$
10: **end for**

Algorithm 2 FedDL-Server

Input: L, N, T, *intvl*, λ, $\mathcal{D}_s$

1: $Groups = \{0 : [G_{0,1}]\}$, $(G_{0,1} = \{1, 2, ...N\})$
2: $step = 0$
3: **for** $t = 0$ to $T-1$ **do** ▷ Global update rounds
4: Receive models, $\boldsymbol{W}_i$, from all the clients.
5: **if** $step = intvl$ **then** ▷ Grouping
6: $M_a = Affinity(\mathcal{D}_s,\ [\boldsymbol{W}_1, ...\boldsymbol{W}_N])$
7: $Grouping(\ M_a, Groups\ )$
8: $step = 0$, $intvl = (1-\lambda)intvl$
9: **end if**
10: $[\boldsymbol{W}_{G_{l,1}}^{(t)}, ...\boldsymbol{W}_{G_{l,k}}^{(t)}] = Merge(Groups, [\boldsymbol{W}_1, ...\boldsymbol{W}_N])$ ▷ Merge
11: $[\boldsymbol{W}_1^{(t)}, ...\boldsymbol{W}_N^{(t)}] = Align([\boldsymbol{W}_{G_{l,1}}^{(t)}, ...\boldsymbol{W}_{G_{l,k}}^{(t)}], [\boldsymbol{W}_1, ...\boldsymbol{W}_N])$
12: Send $\boldsymbol{W}_1^{(t)}, ...\boldsymbol{W}_N^{(t)}$ to the corresponding clients.
13: $step = step + 1$
14: **end for**

5.4 Reducing Communication Overhead

In typical FL systems [11, 45, 51, 62], a large number of global communication rounds between users and the server is required, which can be the bottleneck of the learning process. FedDL takes advantage of the dynamic layer-wise sharing scheme to improve communication performance. Specifically, FedDL reduces the number of parameters that each user needs to upload to the server as well as the number of global training rounds.

As shown in Fig. 5, after learning the grouping results for the l-th layer, only the lower l layers of local models need to be merged at the server for each global training. Thus, FedDL uploads the

lower l layers of local models for the global model merging where l increases from 1 to $L-1$ during the training process, largely reducing the amount of data transferred. Besides, FedDL further reduces the communication overhead by stopping the upload of each model's user-specific layers whenever possible. As shown in the left of Fig. 3, at the third layer, n_1, n_2 and n_3 no longer belong to any group, i.e. their layers are user-specific. After obtaining the fourth layer's grouping results, unlike $n_4 - n_6$, n_1, n_2 and n_3 need to upload only the lower 2 layers to the server during all the following model merging rounds.

Moreover, the models trained using FedDL converge fast even with a small number of local training rounds, which is detailed in Section 6. Thus, FedDL can use a small number of global training rounds to reduce communication costs. As shown in Section 6, FedDL-based models can always converge within 10 global rounds with different settings of local rounds R, while other FL methods may take more than 30 rounds to converge. Therefore FedDL largely reduces the communication overhead.

6 EVALUATION

In this section, we evaluate the performance of FedDL from three aspects, including the performance on different datasets, the scalability of the system, and its performance with different local computation rounds. For each evaluation, we compare the performance of FedDL with four baselines as follows:

(1) **FedAvg** [45]: the standard FL method, where all users share one global model.
(2) **FedPer** [7]: a federated deep learning approach, where all the users share their lower K layers and leave their upper layers user-specific. This approach adopts a K-sharing scheme and pre-sets the value of K empirically, as mentioned in Section 3. In our experiments, we set K to be 3.
(3) **pFedMe** [17]: an algorithm for personalized FL using the distance between the global model and the user's local model as the user's regularized loss functions. The global model is an average aggregation of all the local models at the server.
(4) **Local training**: the model learned from local data at each user.

6.1 Datasets

In our evaluation, we use one self-collected dataset and four public real-world datasets (Table 1) for deep learning. We use the self-collected LiDAR data for two main reasons. First, most of the existing HAR datasets lack detailed information about subjects, such as gender, height, and weight. Such information is critical to understand the underlying similarity of users' data and hence is important to validate the design of FedDL. Second, LiDAR has a long detection distance, which facilitates recognizing whole-body movements, like bending and falling. At the same time, compared with RGB images, Lidar data is more spare presents a major challenge in achieving high model accuracy, which motivates the adoption of FL to enable collaborative learning from multiple users.

Moreover, we choose additional four public datasets for evaluation as they are collected in real-world settings with significant dynamics. Besides, these datasets are collected from various HAR tasks based on different sensors, like depth sensor and IMU (inertial measurement units). Moreover, some datasets are of large scale, which can be utilized to evaluate systems' scalability.

(1) **Human Activity Recognition using LiDAR**[1] : We record the point cloud data of 6 types of human activities (walking/sitting/standing/bending/checking watch/phone calls) conducted by 10 subjects using a Livox Horizon LiDAR [40] in an indoor environment. The LiDAR collects point clouds at 10Hz, and each activity of a subject lasts for 2 minutes. Fig. 9 shows the preprocessing steps for the collected point clouds, which are first proposed in [8, 49]. First, we conduct the cylinder projection to project the 3D point cloud to a range image of 120×30 pixels, where each pixel's grayscale represents the range value (the whiter, the farther). Then we average every 25 consecutive range images in sliding windows of 2.5 seconds and 50% overlap to form each data record. After that, the ROIs (region of interest) of each image are retrieved, and then we down-sample the original image to 60×30 pixels and normalize the depth value to 0-1. This dataset has a large number of data records (6560 records in total), and each data record's dimension is relatively high, thus increasing the difficulty of activity recognition.
(2) **Human Movement Detection using UWB** [1]: To detect if there were human movements in a specific area, two UWB (Ultra Wide Band) nodes are deployed 3m away from each other in 3 different environments (i.e., parking lot, corridor, room) with or without a person walking between them. This dataset is collected using 8 subjects, with each one walking randomly in the area for 10 minutes. The two-way ranging at 5Hz is captured and labeled manually. Then the data is sampled in sliding windows of 10 seconds and 50% overlap (50 readings/window) to form each data record (50×1 dimensions).
(3) **Hand Gesture Recognition using Depth Camera** [1]: Five types of gestures (good/ok/victory/stop/fist) are conducted by 8 subjects using a depth camera. The region of interest of the depth image is retrieved, and then we down-sample the original image to 40×40 pixels and normalize the depth value to 0-1.
(4) **Activity of Daily Life (ADL) Recognition using Smartphones** [1]: The "HARBOX" App is developed to collect 9-axis IMU (inertial measurement units) data from users' smartphones when the user conducts five types of ADL, including walking, hopping, phone calls, waving and typing. Labeled IMU data from 121 users is collected in total. The data from each user is filtered and then sliced into multiple frames (100×9 dimensions) using a window of 2 seconds and 50% overlap.
(5) **Human Activity Recognition using Smartphones** [2]: this online dataset is collected from 30 subjects performing six activities (walking, walking_upstairs, walking_downstairs, sitting, standing, lying) while carrying a waist-mounted smartphone (Samsung Galaxy S II) with embedded IMU. Specifically, the 3-axial linear acceleration and 3-axial angular velocity are captured at a constant rate of 50Hz and are labeled

[1]The data collection was approved by IRB of the authors' institution.
[2]https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

Table 1: Five HAR datasets (UWB, Depth Images, HARBOX-IMU, IMU and LiDAR).

Application	Tasks	Sensor	Data Dimension	Number of subjects	Number of records per subject
Human movement detection	with/without human movement	UWB	50x1	8	~ 80
Hand Gesture Recognition	good/ok/victory/stop/fist	Depth camera	36x36x1	9	~ 400
Activity of Daily Life (ADL) recognition using IMU	walking/hopping/phone calls/waving/typing	IMU	100x9x1	121	~ 300
Human Activity Recognition using IMU	walking-upstair/ walking-downstair/ walking/sitting /standing/laying.	IMU	128x3x2	30	~ 300
Human Activity Recognition using LiDAR	walking/bending/phone calls/sitting/standing/ checking watch.	Livox Horizon LiDAR	60x30x1	10	~ 600

manually through video records. The 6-dimension sensor signals were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 seconds and 50% overlap (128 readings/window) to form each data frame with a size of 128×6.

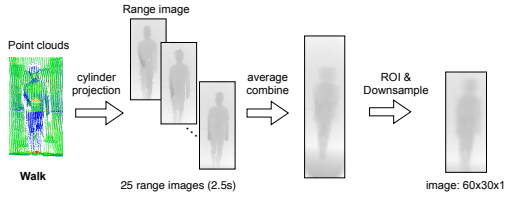

Figure 9: The preprocessing of LiDAR data for the recognition of activities, including walking, sitting, standing, bending, checking the watch and phone calls.

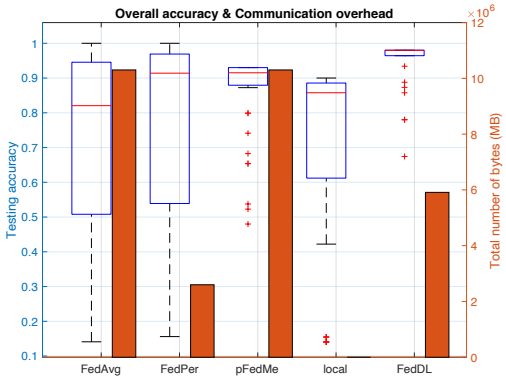

Figure 10: Comparison of different approaches' performance on the LiDAR dataset. FedDL outperforms other approaches in accuracy performance by more than 15%, and save about 42.6% communication overhead compared with approaches that share the whole models (Fedavg and pFedMe).

6.2 Implementation

We design and implement a FedDL phototype on Amazon Elastic Compute Cloud (Amazon EC2). This EC2 instance is built on the Ubuntu platform and has 96 virtual CPUs (3.1 GHz) and 768 GB memory. We build a server on the instance and run each user end on one CPU to simulate the FL. The communication between the server and users is implemented locally using sockets. The system is implemented in Python3.

We adopt randomly initialized convolutional neural networks (CNN) for the human activity recognition tasks of the five datasets. The CNN network is composed of 2 convolutional layers, 2 full-connect layers, and one softmax output layer. It uses mini-batch Stochastic Gradient Descent (SGD) for optimization. For the data samples of each subject, we use 75% of the local data for model training, while the rest 25% is for model testing. We set the initial learning rate to be 0.01 with periodical decay and the batch size to be 32. Although with the same depth, the CNN models for different datasets will have various network structures (e.g., input dimension, kernel size, stride, and padding) depending on the data characteristics and the tasks.

6.3 Validation on LiDAR Dataset

In this section, we validate the design of FedDL on the LiDAR dataset. Specifically, we compare the performance of FedDL with four baselines, FedAvg, FedPer, pFedMe, and local training. We set the local communication rounds (R) to be 30 and the global computation rounds (T) to be 40. We involve totally 10 users for the FL on the LiDAR dataset.

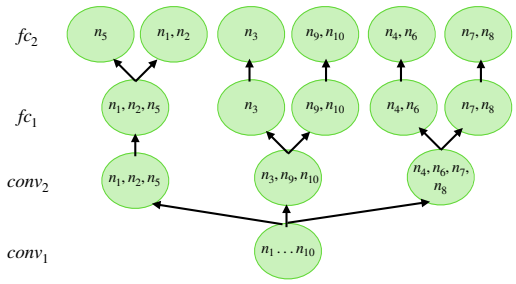

Figure 11: The sharing structure for 10 users, which is dynamically learned by FedDL. n_2, and n_1 share more layers as they have similar behavior habits and biological features.

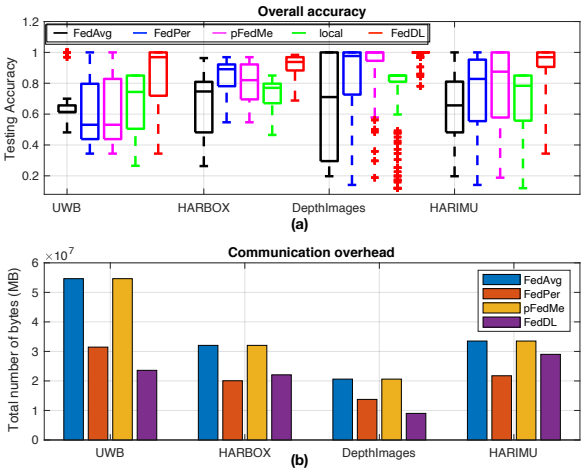

Figure 12: Comparison of different approaches' performance on four datasets, UWB, HARBOX, Depth Images and IMU. FedDL outperforms other approaches in accuracy performance and has a lower communication overhead than approaches that share the whole models (Fedavg and pFedMe).

Fig. 10 shows the overall accuracy and the communication overhead of different approaches on the LiDAR dataset. We evaluate the overall accuracy by observing the distribution of testing accuracy after 30 rounds of global training for all the users. From Fig. 10, we can see that FedDL achieves the best accuracy performance with $mean_{acc} = 0.98$ and the interquartile range $IQR = 0.025$. Compared with other methods, FedDL improves the mean testing accuracy by more than 15% and reduces the variation significantly by more than 94%, suggesting that FedDL can converge fast to a steady and accurate model for most users. In contrast, FedAvg and FedPer yield larger testing accuracy variations as models of some users barely converge even after 30 rounds of global training. FedDL achieves a significantly lower variation as it facilitates the collaboration among users with similar data distributions, which mitigates the noise/outliers from other users, improving the convergence rate and accuracy. Fig. 10 also compares the communication overhead of FedDL with the other three FL methods. We measure the communication overhead (Q_{comm}) by calculating the total amount of data transferred between the server and users during the training procedure. It is shown that FedDL saves about 42.6% communication overhead compared with FedAvg and pFedMe, which share the entire models during FL.

To better understand the above results, we take a closer look at the sharing structure dynamically learned by FedDL (shown in Fig. 11). From the figure, we can see, n_3, n_9, and n_{10} share the lower two layers, which is consistent with the fact that they are the only three subjects using the left hand to make phone calls. Among these three subjects, n_9 and n_{10} are females (n_9: heights 1.66m, weights 50kg; n_{10}: heights 1.63m, weights 48kg), while n_3 is a male with the height 1.78m and weight 66kg. It is shown that n_9 shares more layers with n_{10} than with n_3, which can be attributed to the distinct effects of the body shapes on the collected LiDAR data. The effect of biological features on the LiDAR data is also reflected on n_5. Users n_5, n_2, and n_1 use both the left and right hands to answer phone calls, and they are all males. However, n_5 (height 1.93m, weights 95kg) is much taller and heavier than the other two subjects. In the sharing structure, n_5 shares the lower 3 layers with n_2 and n_1, while n_2 and n_1 keep sharing more upper layers.

The above results confirm that FedDL can capture the different degrees of similarity among users' data due to behavior habits or biological features, and can effectively apply them to layer-wise model merging to improve model accuracy and communication efficiency.

6.4 Performance on Different Datasets

In this section, we evaluate the performance of FedDL on different datasets, UWB, HARBOX-IMU, depth images, and IMU (Table .1). Specifically, for each dataset, we compare the overall accuracy and communication overhead of FedDL with four baselines, FedAvg, FedPer, pFedMe, and local training. We fix the local communication rounds (R) to be 30 and the global computation rounds (T) to be 40 for all the approaches. Also, we involve 8 users for the FL on each dataset, where the number of data samples varies for different users to simulate an unbalance data setting in FL. It is noted that we evaluate the scalability of FedDL on HARBOX dataset involving up to 90 users in Section 6.4.

Overall accuracy. Fig. 12(a) compares the testing accuracy of different approaches for the four datasets. It is shown that compared with four baselines, FedDL achieves the best and stable accuracy performance on the four datasets with a high mean value ($mean_{acc} > 90\%$) and $IQR < 0.2$. Specifically, compared with local training ($0.05 < IQR < 0.4$, $75\% < mean_{acc} < 85\%$), FedDL, FedPer, and pFedMe improve the accuracy of the model while FedAvg fails, as the data distributions of users are too heterogeneous to learn a good global model. Specifically for the UWB dataset, FedAvg barely converges within 40 global rounds. FedPer and pFedMe also fail to improve the accuracy as their model aggregation schemes are oblivious to the underlying relationship among users. Moreover, FedDL outperforms them, as FedDL can capture the intrinsic relationship among users dynamically and aggregate users' models within each group in a layer-wise manner.

Communication overhead. Fig. 12(b) compares the communication overhead of different methods for the four datasets. In our

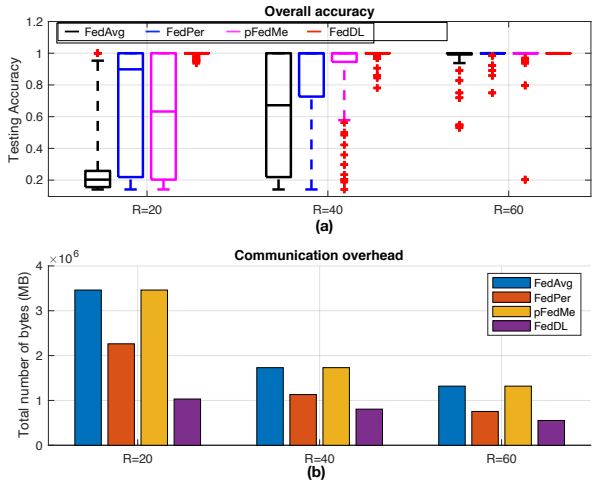

Figure 13: Comparison of different approaches' performance on Depth images datasets with different number of local computation rounds ($R = 20, 40, 60$). All the methods benefits from a larger R, and FedDL maintains the best accuracy and communication performance with different numbers of R.

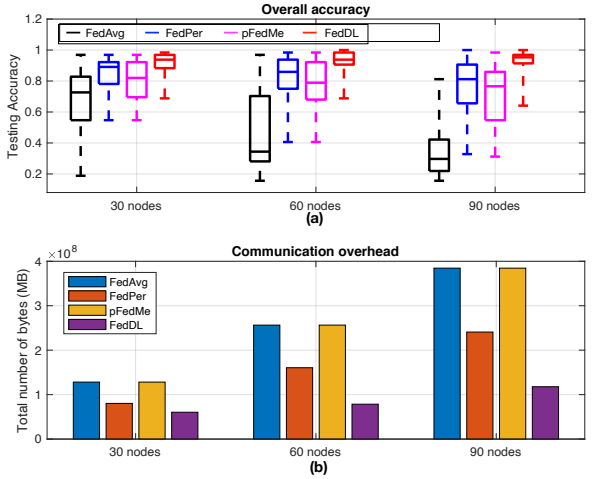

Figure 14: Comparison of different approaches' performance on 30-, 60- and 90-user HARBOX datasets. FedDL outperforms FedAvg, FedPer and pFedMe in both overall accuracy and communication overhead.

experiments, we set the number of global rounds $T = 40$. The communication overhead measures the total amount of the parameters transferred between users and the server during the whole FL process, which is determined by the sharing scheme and the size of the CNN model. From the figure, we can see FedDL is able to maintain a relatively low communication overhead, which suggests our dynamic bottom-up layer-wise model aggregation strategy improves the communication efficiency. Specifically, FedDL and FedPer have a relatively low communication cost for all the datasets, as they only share part of model layers among users. FedPer combines the lower 3 layes of local models and FedDL merges models according to layer-wise grouping results. In particular, FedDL outperforms FedPer for UWB and depth images datasets. The reason is that the data distributions of users are so heterogeneous in these two datasets that most of the users' upper layers are user-specific in FedDL's grouping results, i.e., they share and upload less than 3 lower layers.

6.5 Scalability

To evaluate the scalability of FedDL, we compare the performance of different approaches (FedDL, FedAvg, FedPer, pFedMe) when training on the data of 30, 60, 90 users from the HARBOX dataset.

Overall accuracy. Fig. 14 shows the experiment results with different number of users. From Fig. 14(a), It is obvious that the overall accuracy of FedAvg decreases with the increase of the number of users, as the heterogeneity of users' data becomes higher. In this case, FedAvg performs the worst among all approaches and can not even converge within 40 global rounds when 90 users are involved. Besides, FedDL outperforms FedAvg, FedPer and pFedMe under different settings as FedDL can capture the relationship among users and dynamically merge user' models within each group in a layer-wise manner. On the contrary, FedPer adopts a static sharing scheme that shares the lower 3 layers of models for all the users, which fails to capture the complicated user relationship, resulting in worse performance when the number of involved users is large. pFedMe aligns each user's local model with the averaged global models, which makes the overall accuracy partially dependent on the global model's performance, which is hence largely influenced by users' data heterogeneity. Moreover, the accuracy of FedDL is more stable (with small IQRs) as the number of users increases, which shows the advantage of its group-based dynamic model aggregation scheme.

Communication overhead. Fig. 14(b) compares the communication overhead of different approaches with the data of 30, 60, 90 users from the HARBOX datasets. We can see that the communication overhead of FedAvg, pFedMe and FedPer increases dramatically in proportion to the number of users involved in the training procedure. However, FedDL always maintains a relatively low communication overhead, as FedDL can stop uploading the parameters of models' upper layers earlier when the users' data is significantly heterogeneous.

The above results suggest that FedDL exhibits satisfactory scalability by maintaining relatively high accuracy and low communication overhead and performs better on large-scale datasets.

6.6 Impact of Local Computation Rounds

The number of local computation rounds, R, is a critical hyperparameter in FL. The setting of R shows a trade-off between the computation and communication: a larger R requires more computations at local devices of users, while a smaller R means more global communication rounds to converge. To understand how R affects the convergence of different FL methods, we conduct the experiments on an 8-user Depth Images dataset with ($R = 20, T = 30$), ($R = 40, T = 15$) and ($R = 60, T = 10$, respectively. It is noted that, for all the baselines, we only change the value of R with the model

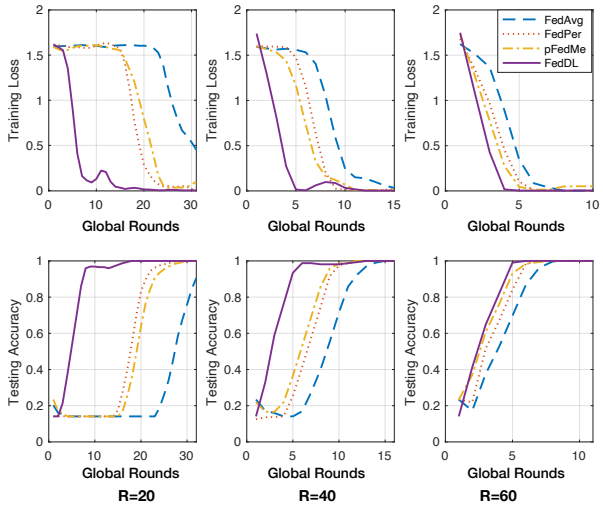

Figure 15: The training loss and testing accuracy of a specific user's model changing over global rounds with different settings of R. Larger R improves convergence, especially for FedAvg. However, FedDL will always converge fastest with different local computation rounds R.

structure and all the other settings of the models stay the same. Specifically, the initial learning rate is set to be 0.01 with periodic decay and the batch size is set to be 32.

Fig. 13 illustrates the performance of different methods with different settings of local computation rounds R. It shows that a larger value of R will improve the performance on the accuracy and communication overhead of both the personalized and the global models. Fig. 15 visualizes the change of training loss and testing accuracy over global rounds with different settings of R for a specific user. We can see that all the methods have improvements in convergence when R is larger. For example, FedAvg takes a much smaller number of global communication rounds to converge (reduce from more than 30 to 10 rounds) when R increases from 20 to 40. However, FedDL will always converge fastest (with the smallest number of global rounds), especially when the local computation round R is set small (e.g., R=20).

7 DISCUSSION AND FUTURE WORK

Convergence of FedDL. In our experiments (discussed in Section 6.3-6.6), FedDL is demonstrated to converge on the five real-world HAR datasets. In particular, it converges fast even when training on 90 users with a limited number of local rounds. We now provide some insights into the convergence guarantee of FedDL. Firstly, FedDL groups users with similar data distributions, which mitigates the impact of noise/outliers from other users, thus improving the convergence performance. Second, the intra-group model merging entails a weighted average of the local models (see section 5.2), where the weights quantify how closely each local model is to the group model. In FedDL, the weights of users whose models lie at the border or intersection of multiple groups are relatively small, and hence the models will contribute less to the intra-group model merging. Thus, such a design mitigates the impact of dynamic grouping on model convergence.

Scalability of FedDL. FedDL is generally more scalable as a clustering-based approach since the number of user groups (who share some degree of similarity among their data) may not increase drastically with the number of users. For the scenarios where users arrive dynamically, FedDL merges the new users in the sharing structure instead of retraining the sharing structure for all the users for scratch, which substantially reduces the compute and communication overhead. Specifically, FedDL considers each group as one user and learns the new users' relationship with existing groups to update the sharing structure by merging the new users into different groups.

Future work. Firstly, the local models transmitted in FedDL may reveal certain information about user activities [34, 65]. In the future, we will integrate additional mechanisms, like differential privacy [64], in FedDL to provide stronger privacy protection. However, such privacy-preserving mechanisms can have a complicated impact on the overall performance. We will conduct a comprehensive study of privacy-preserving techniques and the trade-off between the privacy and performance of FedDL. Besides, we will extend FedDL to other applications where the users' data has a high level of dynamics while exhibiting significant similarity. For example, FedDL can be applied to applications like health monitoring [3] and road traffic prediction [47], where the data of nodes (e.g., users or cars) share spatial-temporal similarity due to spatial proximity, models of devices/cars, user routines, etc. Finally, as the real-world HAR applications may involve high-dimension data (e.g., images or videos), deeper or wider neural network models are required to avoid underfitting. We will evaluate how the model complexity, including the depth and width of the model, affects the convergence and accuracy of FedDL.

8 CONCLUSION

This paper proposes FedDL, a novel federated deep learning system for HAR that captures the similarity of users' models and generates personalized user models through dynamic layer sharing in an iterative layer-wise manner. We evaluate the performance of FedDL for the recognition of various activities on five datasets collected from 178 users in total. The experimental results show that FedDL outperforms the other methods in terms of overall accuracy (e.g., by 24.05%, 16.67%, 19.51%, and more than 30.67%, to local training, pFedMe, FedPer, and FedAvg respectively). Moreover, FedDL saves more than 50% communication overhead when there is a large number of users and achieves a high convergence rate even with a small number of local computation rounds. As future work, we will deploy FedDL on edge devices, like smartphones, to evaluate the system overhead of FedDL. Moreover, we will also explore the application scenarios with intrinsic statistical heterogeneity beyond HAR by leveraging domain adaptation techniques.

9 ACKNOWLEDGMENTS

This work is supported in part by Research Grants Council (RGC) of Hong Kong under General Research Fund grants 14209619 and 14203420, and National Natural Science Foundation of China under Grant 62032021.

REFERENCES

[1] 2021. Federated learning datasets for human activity recognition. https://github.com/xmouyang/FL-Datasets-for-HAR/tree/main/datasets

[2] Karim T Abou-Moustafa and Frank P Ferrie. 2012. A note on metric properties for some divergence measures: The Gaussian case. In *Asian Conference on Machine Learning*. PMLR, 1–15.

[3] Khan Alam, Salman Qureshi, and Thomas Blaschke. 2011. Monitoring spatio-temporal aerosol patterns over Pakistan based on MODIS, TOMS and MISR satellite data and a HYSPLIT model. *Atmospheric environment* 45, 27 (2011), 4641–4651.

[4] Elie Aljalbout, Vladimir Golkov, Yawar Siddiqui, Maximilian Strobel, and Daniel Cremers. 2018. Clustering with deep learning: Taxonomy and new methods. *arXiv preprint arXiv:1801.07648* (2018).

[5] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. 2012. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *International workshop on ambient assisted living*. Springer, 216–223.

[6] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. 2013. A public domain dataset for human activity recognition using smartphones.. In *Esann*, Vol. 3. 3.

[7] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. 2019. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818* (2019).

[8] Csaba Benedek, Bence Gálai, Balázs Nagy, and Zsolt Jankó. 2016. Lidar-based gait analysis and activity recognition in a 4d surveillance system. *IEEE Transactions on Circuits and Systems for Video Technology* 28, 1 (2016), 101–113.

[9] Sourav Bhattacharya and Nicholas D Lane. 2016. From smart to deep: Robust activity recognition on smartwatches using deep learning. In *2016 IEEE International conference on pervasive computing and communication workshops (PerCom Workshops)*. IEEE, 1–6.

[10] Chongguang Bi, Guoliang Xing, Tian Hao, Jina Huh, Wei Peng, and Mengyan Ma. 2017. Familylog: A mobile system for monitoring family mealtime activities. In *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 21–30.

[11] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, et al. 2019. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046* (2019).

[12] Liang Cao, Yufeng Wang, Bo Zhang, Qun Jin, and Athanasios V Vasilakos. 2018. GCHAR: An efficient Group-based Context—Aware human activity recognition on smartphone. *J. Parallel and Distrib. Comput.* 118 (2018), 67–80.

[13] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.

[14] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. 2020. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461* (2020).

[15] Mariella Dimiccoli, Juan Marín, and Edison Thomaz. 2018. Mitigating bystander privacy concerns in egocentric activity recognition with deep learning and intentional image degradation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 1–18.

[16] Shuya Ding, Zhe Chen, Tianyue Zheng, and Jun Luo. 2020. RF-net: a unified meta-learning framework for RF-enabled one-shot human activity recognition. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 517–530.

[17] Canh T Dinh, Nguyen H Tran, and Tuan Dung Nguyen. 2020. Personalized federated learning with Moreau envelopes. *arXiv preprint arXiv:2006.08848* (2020).

[18] Yegang Du, Yuto Lim, and Yasuo Tan. 2019. A novel human activity recognition and prediction in smart home based on interaction. *Sensors* 19, 20 (2019), 4474.

[19] Theodoros Evgeniou, Charles A Micchelli, Massimiliano Pontil, and John Shawe-Taylor. 2005. Learning multiple tasks with kernel methods. *Journal of machine learning research* 6, 4 (2005).

[20] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948* (2020).

[21] Gene Glass and Kenneth Hopkins. 1996. Statistical methods in education and psychology. *Psyccritiques* 41, 12 (1996).

[22] Yu Guan and Thomas Plötz. 2017. Ensembles of deep lstm learners for activity recognition using wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), 1–28.

[23] Farzin Haddadpour, Mohammad Mahdi Kamani, Aryan Mokhtari, and Mehrdad Mahdavi. 2021. Federated learning with compression: Unified analysis and sharp guarantees. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2350–2358.

[24] Nils Y Hammerla, Shane Halloran, and Thomas Plötz. 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880* (2016).

[25] Mohammed Mehedi Hassan, Md Zia Uddin, Amr Mohamed, and Ahmad Almogren. 2018. A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems* 81 (2018), 307–313.

[26] Andrey Ignatov. 2018. Real-time human activity recognition from accelerometer data using Convolutional Neural Networks. *Applied Soft Computing* 62 (2018), 915–922.

[27] Laurent Jacob, Francis Bach, and Jean-Philippe Vert. 2008. Clustered multi-task learning: A convex formulation. *arXiv preprint arXiv:0809.2085* (2008).

[28] Martin Jaggi, Virginia Smith, Martin Takáč, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I Jordan. 2014. Communication-efficient distributed dual coordinate ascent. *arXiv preprint arXiv:1409.1458* (2014).

[29] Ahmad Jalal and Shaharyar Kamal. 2014. Real-time life logging via a depth silhouette-based human activity recognition system for smart home services. In *2014 11th IEEE International conference on advanced video and signal based surveillance (AVSS)*. IEEE, 74–80.

[30] Ahmad Jalal, Md Zia Uddin, and T-S Kim. 2012. Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home. *IEEE Transactions on Consumer Electronics* 58, 3 (2012), 863–871.

[31] Yanjun Jia. 2009. Diatetic and exercise therapy against diabetes mellitus. In *2009 Second International Conference on Intelligent Networks and Intelligent Systems*. IEEE, 693–696.

[32] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, et al. 2018. Towards environment independent device free human activity recognition. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. 289–304.

[33] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. 2019. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488* (2019).

[34] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2019. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977* (2019).

[35] Md Abdullah Al Hafiz Khan, Nirmalya Roy, and Archan Misra. 2018. Scaling human activity recognition via deep learning-based domain adaptation. In *2018 IEEE international conference on pervasive computing and communications (PerCom)*. IEEE, 1–9.

[36] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* (2016).

[37] Nicholas D Lane, Ye Xu, Hong Lu, Shaohan Hu, Tanzeem Choudhury, Andrew T Campbell, and Feng Zhao. 2011. Enabling large-scale human activity inference on smartphones using community similarity networks (csn). In *Proceedings of the 13th international conference on Ubiquitous computing*. 355–364.

[38] Jia Li, Yu Rong, Helen Meng, Zhihui Lu, Timothy Kwok, and Hong Cheng. 2018. TATC: predicting Alzheimer's disease with actigraphy data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 509–518.

[39] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2019. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189* (2019).

[40] Zheng Liu, Fu Zhang, and Xiaoping Hong. 2021. Low-cost retina-like robotic lidars based on incommensurable scanning. *IEEE/ASME Transactions on Mechatronics* (2021).

[41] Mingsheng Long, Yue Cao, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2018. Transferable representation learning with deep adaptation networks. *IEEE transactions on pattern analysis and machine intelligence* 41, 12 (2018), 3071–3085.

[42] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Philip S Yu. 2015. Learning multiple tasks with multilinear relationship networks. *arXiv preprint arXiv:1506.02117* (2015).

[43] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. 2017. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5334–5343.

[44] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. 2020. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619* (2020).

[45] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*. PMLR, 1273–1282.

[46] H. B. McMahan and D. Ramage. 2017. (2017). http://dx.doi.org/10.1007/s00779-014-0773-4

[47] Wanli Min and Laura Wynter. 2011. Real-time road traffic prediction with spatio-temporal correlations. *Transportation Research Part C: Emerging Technologies* 19, 4 (2011), 606–616.

[48] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3994–4003.

[49] Mirco Moencks, Varuna De Silva, Jamie Roche, and Ahmet Kondoz. 2019. Adaptive feature processing for robust human activity recognition on a novel multi-modal dataset. *arXiv preprint arXiv:1901.02858* (2019).

[50] Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? *arXiv preprint arXiv:1603.06111* (2016).

[51] Adrian Nilsson, Simon Smith, Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. 2018. A performance evaluation of federated learning algorithms. In *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning*. 1–8.

[52] Xiaomin Ouyang, Zhiyuan Xie, Jiayu Zhou, Jianwei Huang, and Guoliang Xing. 2021. 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*. 54–66.

[53] Liangying Peng, Ling Chen, Zhenan Ye, and Yi Zhang. 2018. Aroma: A deep multi-task learning based simple and complex human activity recognition method using wearable sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2 (2018), 1–16.

[54] Valentin Radu, Nicholas D Lane, Sourav Bhattacharya, Cecilia Mascolo, Mahesh K Marina, and Fahim Kawsar. 2016. Towards multimodal deep learning for activity recognition on mobile devices. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. 185–188.

[55] Daniele Riboni and Claudio Bettini. 2011. COSAR: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing* 15, 3 (2011), 271–289.

[56] Daniel Jorge Loureiro Fidalgo do Vale Rodrigues. 2019. *Risk Assessment for Alzheimer Patients, using GPS and Accelerometers with a Machine Learning Approach*. Ph.D. Dissertation.

[57] Charissa Ann Ronao and Sung-Bae Cho. 2016. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert systems with applications* 59 (2016), 235–244.

[58] Patrice C Roy, Sylvain Giroux, Bruno Bouchard, Abdenour Bouzouane, Clifton Phua, Andrei Tolstikov, and Jit Biswas. 2011. A possibilistic approach for activity recognition in smart homes for cognitive assistance to Alzheimer's patients. In *Activity Recognition in Pervasive Intelligent Environments*. Springer, 33–58.

[59] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).

[60] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. 2019. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems* (2019).

[61] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. 2017. Federated multi-task learning. In *Advances in Neural Information Processing Systems*. 4424–4434.

[62] Konstantin Sozinov, Vladimir Vlassov, and Sarunas Girdzijauskas. 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, 1103–1111.

[63] Arijit Ukil, Soma Bandyoapdhyay, Chetanya Puri, and Arpan Pal. 2016. IoT healthcare analytics: The importance of anomaly detection. In *2016 IEEE 30th international conference on advanced information networking and applications (AINA)*. IEEE, 994–997.

[64] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. 2020. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security* 15 (2020), 3454–3469.

[65] Liyang Xie, Inci M Baytas, Kaixiang Lin, and Jiayu Zhou. 2017. Privacy-preserving distributed multi-task learning with asynchronous updates. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1195–1204.

[66] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

[67] Jie Yin, Qiang Yang, and Jeffrey Junfeng Pan. 2008. Sensor-based abnormal human-activity detection. *IEEE Transactions on Knowledge and Data Engineering* 20, 8 (2008), 1082–1090.

[68] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in neural information processing systems*. 3320–3328.

[69] Shizhen Zhao, Wenfeng Li, and Jingjing Cao. 2018. A user-adaptive algorithm for activity recognition based on k-means clustering, local outlier factor, and multivariate gaussian distribution. *Sensors* 18, 6 (2018), 1850.

[70] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582* (2018).

[71] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. 2014. Learning deep features for scene recognition using places database. (2014).