Multimodal Federated Learning on IoT Data

Yuchen Zhao
UK Dementia Research Institute
Imperial College London
yuchen.zhao19@imperial.ac.uk

Payam Barnaghi
UK Dementia Research Institute
Imperial College London
p.barnaghi@imperial.ac.uk

Hamed Haddadi
UK Dementia Research Institute
Imperial College London
h.haddadi@imperial.ac.uk

***Abstract*—Federated learning is proposed as an alternative to centralized machine learning since its client-server structure provides better privacy protection and scalability in real-world applications. In many applications, such as smart homes with Internet-of-Things (IoT) devices, local data on clients are generated from different modalities such as sensory, visual, and audio data. Existing federated learning systems only work on local data from a single modality, which limits the scalability of the systems.**

In this paper, we propose a multimodal and semi-supervised federated learning framework that trains autoencoders to extract shared or correlated representations from different local data modalities on clients. In addition, we propose a multimodal FedAvg algorithm to aggregate local autoencoders trained on different data modalities. We use the learned global autoencoder for a downstream classification task with the help of auxiliary labelled data on the server. We empirically evaluate our framework on different modalities including sensory data, depth camera videos, and RGB camera videos. Our experimental results demonstrate that introducing data from multiple modalities into federated learning can improve its classification performance. In addition, we can use labelled data from only one modality for supervised learning on the server and apply the learned model to testing data from other modalities to achieve decent F_1 scores (*e.g.*, with the best performance being higher than 60%), especially when combining contributions from both unimodal clients and multimodal clients.

***Index Terms*—collaborative work, semisupervised learning, edge computing, multimodal sensors**

I. Introduction

In recent years, we have witnessed a rapid growth in personal data generated from many different aspects in people's daily lives, such as mobile devices and IoT devices. Powered by the enormous amount of personal data, machine-learning (ML) techniques, especially Deep Neural Networks (DNN), have shown great capabilities of conducting complex tasks such as image recognition, natural language processing, human activity recognition, and so forth. Traditionally, ML systems are centralized and need to collect and store personal data on a server to train DNN models, which causes privacy issues. The long-debated privacy issues in centralized ML systems have motivated researchers to design and implement machine learning in decentralized fashions. Federated learning (FL) [1], which allows different parties to jointly train DNN models without releasing their local data, is a system paradigm that has gained much popularity in both research communities and real-world ML applications.

In FL systems, DNN models are trained on clients at the edge of networks instead of on servers in the cloud. This makes FL systems specifically suitable for privacy sensitive applications such as smart home [2]–[4] based on IoT technologies. For example, Wu *et al.* [5] propose an FL framework that uses personalization to address the device, statistical and model heterogeneity issues in IoT environments. Pang *et al.* [6] propose an FL framework using reinforcement learning to adjust the model aggregation strategy on models trained with IoT data. As a distributed system paradigm, FL provides a feasible and scalable solution for realizing ML on resource-constrained IoT devices [7].

IoT applications often deploy different types of sensors or devices that generate data from different modalities (*e.g.*, sensory, visual, and audio) [8]. For example, in one smart home, activities of a person can be recorded by body sensors in a smartwatch worn by the person, and also by a video camera in the room at the same time. Meanwhile, for smart homes with different device setups, some of them may have multimodal local data (*i.e.*, *multimodal clients*) while the others may have unimodal local data (*i.e.*, *unimodal clients*). One way to apply FL to these IoT applications is to implement individual services for different modalities. However, many centralized ML systems [9]–[13] have shown that combining data from different modalities can improve their performance. Therefore, it is necessary to design and implement FL systems in a way that supports multimodal IoT data and different device setups.

To work on multimodal data, one approach in existing FL systems uses data fusion [14] to mix representations from different modalities before a final decision layer into a new representation space. This requires all the data (*i.e.*, training and testing) in the system to be aligned multimodal data, which means that all the clients need to have data from all modalities in the system. In addition, the labelled data in the system also need to be from all modalities, in order to support supervised learning on the new representation space. This does not work on systems with unimodal clients and increases the complexity of data annotation. Another approach [15] extracts representations from different modalities locally and requires the clients to send the representations to the server in order to align different modalities. This may break the privacy guarantee provided by FL since the representations can be used to recover local data, especially when the server has taken part in the training of the model that extracts the representations. Allowing FL to work on clients with arbitrary data modalities (*i.e.*, unimodal or multimodal) and with labelled data that come from single modalities, however, still remains a challenge.

In this paper, we propose a multimodal FL framework that takes advantage of aligned multimodal data on clients. Although acquiring alignment information for multimodal data across different clients is challenging, our assumption is that data from different modalities (*e.g.*, sensory data and visual data) on a *multimodal client* inherently have some alignment information (*e.g.*, through synchronized local timestamps of sensory data samples and video frames on that client), based on which we can train models to extract multimodal representations from the data. We utilize multimodal autoencoders [9], [10] to encode the data into shared or correlated hidden representations. To enable the server in our framework to aggregate trained local autoencoders into a global autoencoder, we propose a multimodal version of the FedAvg algorithm [1] that can combine local models trained on data from both unimodal and multimodal clients.

As it is difficult to have adequate labels on clients in real-world FL systems [14], [16], we focus on semi-supervised scenarios wherein local data on the clients are unlabelled and the server has an auxiliary labelled dataset. We use the global autoencoder and the auxiliary labelled dataset on the server to train a classifier for activity recognition tasks [17], [18] and evaluate its performance on a variety of multimodal datasets (*e.g.*, sensory and visual). Compared with existing FL systems [14], [15], our proposed framework does not share representations of local data to the server. Additionally, instead of requiring the clients and the server to have aligned data from all modalities, our framework conducts local training on both multimodal and unimodal clients, and only needs unimodal labelled data on the server. Our experimental results indicate that our proposed framework can improve the classification performance (F_1 score) of FL systems in comparison to unimodal FL, and allows us to use unimodal labelled data to train models that can be applied to multimodal testing data.

We make the following contributions in this paper:

- We propose a multimodal FL framework that works on data from different modalities and clients with different device setups, and a multimodal FedAvg algorithm.
- Complementing the existing knowledge on the benefit of using multimodal data in centralized ML, we find that introducing data from more modalities into FL also leads to better classification performance.
- We show that classifiers trained on labelled data on the server from one modality can achieve decent classification F_1 scores on testing data from other modalities.
- We show that combining contributions from both unimodal and multimodal clients can further improve the classification F_1 scores.

II. Related work

A. Federated learning

McMahan *et al.* [1] propose federated learning (FL) as an alternative system paradigm to centralized ML. In an FL system, a server acts as an coordinator to select clients and to send a global DNN model to the clients. The clients use their own data to locally train the model and then send the resulting models back to the server, on which these models are aggregated into a new global model. The system repeats this process for a number of rounds until the performance of the global model on a given task converges. The privacy of the clients' data is protected since the data are never shared with others. Given its decentralized feature, FL is especially suitable for edge computing [19], [20], which moves computation to the place where data are generated.

Canonical FL systems focus on supervised learning that requires all local data on FL clients to be labelled. In edge computing, data generated from IoT devices can only be accessed by the data subjects, since FL clients do not share data to third parties. These data subjects (*i.e.*, end users of an FL system) may not have time or abilities to annotate their data with labels of a given task, especially when the task requires expert knowledge (*e.g.*, labelling timer-series sensory data with clinical knowledge). Therefore, one key challenge of deploying FL in real-world IoT environments is the lack of labelled data on clients for local training. In order to address this issue, recent research in FL has been focusing on unsupervised and semi-supervised FL frameworks through data augmentation [16], [21]–[29] to generate pseudo labels for local data, or through unsupervised learning to extract hidden representations from unlabelled local data [17], [18]. For example, van Berlo *et al.* [17] propose to learn hidden representations through convolutional autoencoders from unlabelled local data on FL clients. Their results show that the learned representations can empower downstream tasks such as classifications. Zhao *et al.* [18] propose a semi-supervised FL framework for human activity recognition and compared the performance of different autoencoders. Their framework shows better performance than data augmentation schemes do. Our work in this paper follows the path of the latter category. Compared with the existing research, we enable semi-supervised FL to learn from multiple data modalities.

B. Heterogeneity in federated learning

Heterogeneity is one of the most challenging issues [30], [31] in FL because models are locally trained on clients. Different clients may vary in terms of computational capabilities, model structures, distribution of data, or distribution of features. Among all these issues, the heterogeneity in distribution of data (*i.e.*, non-IID local data) has attracted most research efforts [32]–[35]. Smith *et al.* [32] apply multi-task learning to addressing the issue of training on non-IID data in FL. Instead of training one global model for all clients, they treat each client as a different task and train separate models for them. Similarly, Li *et al.* [34] extend federated multi-task learning to an online fashion and allow new clients to join the system. To address the heterogeneity in the distribution of features when shifting FL from one domain to another, Chen *et al.* [35] propose to use transfer learning to align the features in lower-stream layers (*e.g.*, fully connected layers before final output layers). In order to learn from heterogeneous models (*i.e.*, DNN models with different structures), Lin *et al.* [36]

propose to use knowledge distillation [37] to train global models of FL based on the output probability distribution from local models, instead of directly averaging the parameters of them. Existing research, however, neglected the heterogeneity in data modalities in FL, which is commonplace in many scenarios such as edge computing, IoT environments, and mobile computing.

The recent study by Liu *et al.* [15] applies FL on data from two modalities (*i.e.*, images and texts) and treats each modality individually, which is the same as running two individual FL instances. In the study, to align the two modalities on a server, representations of local data need to be uploaded to the server. This breaks the privacy guarantee of FL because the server has the global model that generates the representations from raw data and could recover the raw data if it has those representations. The framework proposed by Liang *et al.* [14] can work on multimodal data only when the clients' local data, the server's labelled data, and testing data are all aligned data from both modalities. Instead of aligning the representations from different modalities, it conducts early fusion (*i.e.*, element-wise multiplication) on the representations. Thus unimodal data cannot contribute to the local training and the trained model cannot be used on unimodal data. Compared to the existing work, we use the alignment information in local data to learn to extract shared or correlated hidden representations from multiple modalities. Our scheme does not require sending representations of local data to the server, which contradicts the motivation of using FL. In addition, it allows models to be trained and used on unimodal data.

C. Multimodal deep learning

When training deep learning models for a certain task, the used data can be generated from a variety of modalities (*e.g.*, recognizing human activities from IoT sensory data or videos). In order to utilize these data, multimodal deep learning has attracted much attention from researchers. Ngiam *et al.* [9] propose to use deep autoencoders [38] to learn multimodal representations from audio and visual data. The alignment between the two modalities is done by reconstructing the output for both modalities from the hidden representation generated by either modality. Wang *et al.* [10] compare different multimodal representation learning techniques and propose to combine both deep canonical correlation analysis [39] and autoencoders to map data from different modalities into highly correlated representations instead of one common representation. These techniques have demonstrated that data from different modalities can complement each other when learning representations and improve the overall performance of an ML system. Many applications such as audio-visual speech recognition [9], activity and context recognition [12], [13], and textual description generation for images [40], have been implemented based on multimodal deep learning. The recent survey by Baltrušaitis *et al.* [41] provides a detailed analysis and taxonomy of multimodal deep learning. In this paper, we apply multimodal representation learning to FL to address the heterogeneity issue in local data modalities.

III. Methodology

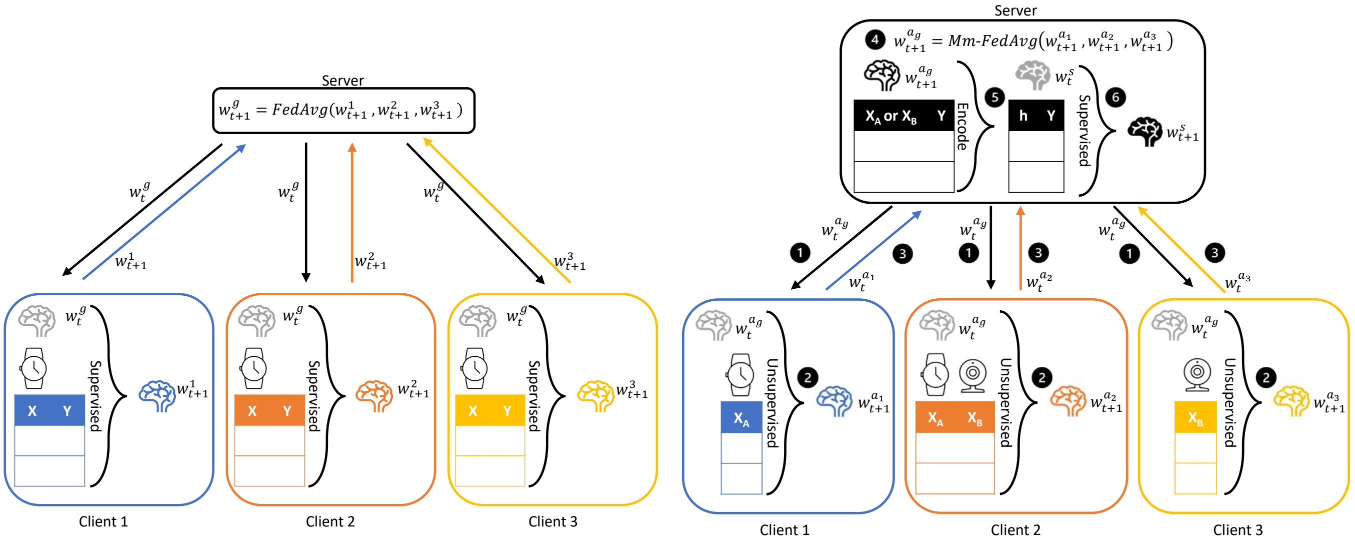
Our goal is to enable FL to work on clients that have different local data modalities. We first introduce the overall design of our framework. We then describe the key techniques that we use to extract representations from multimodal data and the algorithms that we designed to aggregate local models trained on both unimodal and multimodal clients.

A. Framework overview

A canonical FL system, as shown in Fig. 1a, only works on clients that have local data from the same modality and requires the data to be labelled for supervised learning.

We propose an FL framework wherein clients' unlabelled local data can be from either one single modality or multiple modalities. In our framework, as shown in Fig. 1b, unimodal clients (*e.g.*, Clients 1 and 3) only deploy one type of devices due to reasons such as budget or privacy. Multimodal clients (*e.g.*, Client 2) deploy both types of devices and thus have multimodal local data. On a multimodal client, we assume that there is alignment information between the data from two modalities, based on which we can align the hidden representations of two modalities. For example, a person's activity can be captured by the accelerometers in a smartwatch and by an IP camera in the room at the same time. A record of video call contains both the visual information and audio information of a speech. This kind of matching information is the key to align the hidden representations of multimodal data since they describe the same underlying activities or events.

To address the lack of labelled data in FL systems using IoT devices, similar to existing semi-supervised FL frameworks [17], [18], on clients we assume that no labelled local data are available. Thus we learn to extract hidden representations from unlabelled data. On multimodal clients, we train local models to extract shared or correlated representations between different modalities since we have aligned pairs of multimodal data. On unimodal clients, we train models to extract representations from one single modality. Local models from both types of clients are sent to the server and are aggregated into a global model by using a multimodal version of the FedAvg algorithm [1]. The server uses the global model to encode a labelled dataset from either modality into a labelled representation dataset, based on which a classifier is trained through supervised learning. We believe that, as the service provider, the server can provide such an auxiliary dataset with labels that requires expert knowledge about the task of the service. For example, in many existing human activity datasets, labelling activities with sensory data can be done through controlled laboratory trials with the assistance from video cameras and pre-defined trial scripts [42]. The clients receive both the global model and the classifier from the server during each communication round and can use them on their local data for classifications. Alg. 1 describes the the process of multimodal federated learning.



(a) Canonical federated learning

(b) Multimodal federated learning

Fig. 1: In canonical federated learning (a), during round t, a server sends a global model w^g_t to selected clients that have data from the same modality. Client k conducts supervised learning to generate a local model w^k_{t+1}. Local models are aggregated on the server by using the FedAvg algorithm. In multimodal federated learning (b), a server sends a global model $w^{a_g}_t$ to selected clients to learn to extract multimodal representations (Sec. III-B) on unlabelled local data. The server uses multimodal FedAvg (Sec. III-C) to aggregate local models into a new global model $w^{a_g}_{t+1}$ and uses it to encode a labelled dataset (modality A or B) to a labelled representation dataset (h, Y). A classifier w^s_{t+1} is then trained on (h, Y), which can be used by all clients.

Algorithm 1 Multimodal Federated Learning

```
Require: $K$: number of clients; $C$: fraction of clients to choose; $D = (X, Y)$: labelled dataset from either modality (A or B)
initializes $w^{a_g}_0$, $w^s_0$ at $t = 0$
for all communication round $t$ do
    $S_t \leftarrow$ randomly selected $K \cdot C$ clients
    $W_t \leftarrow \emptyset$
    for all client $k \in S_t$ do
        $w^{a_k}_{t+1} \leftarrow$ Multimodal Local Training$(k, w^{a_g}_t)$    ▷ on client $k$
        $W_t \leftarrow W_t \cup w^{a_k}_{t+1}$
    end for
    $w^{a_g}_{t+1} \leftarrow$ Multimodal FedAvg$(W_t)$    ▷ on the server
    $h \leftarrow w^{a_g}_{t+1}.encoder(X)$    ▷ using the encoder for the modality of $X$
    $D'_t \leftarrow (h, Y)$
    $w^s_{t+1} \leftarrow$ Cloud Training$(D'_t, w^s_t)$    ▷ on the server
end for
```

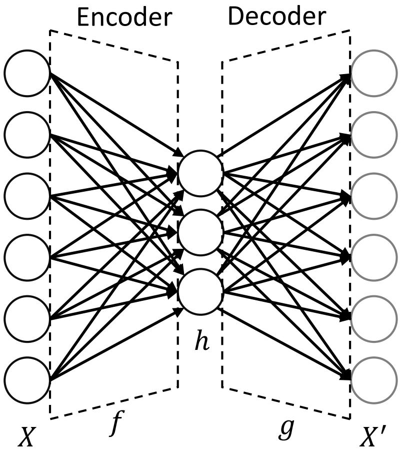
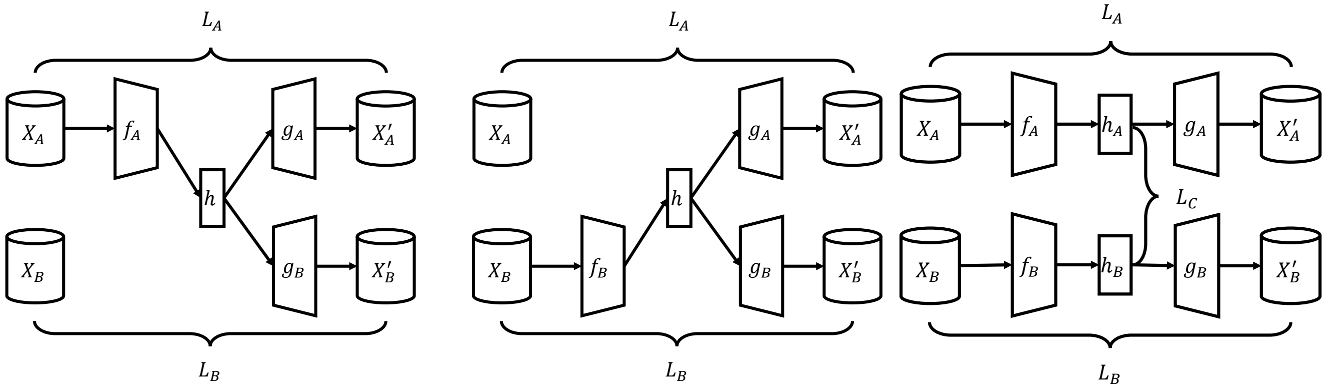


Fig. 2: A simple autoencoder structure. An encoder f maps input data X into a hidden representation h. A decoder g maps h into a reconstruction X'.

B. Learning to extract representations

The key part of the local training in our proposed framework is how to learn representations from unlabelled unimodal data or multimodal data. We first introduce canonical autoencoders, which we train to extract hidden representations from unimodal data. Then we introduce two types of multimodal autoencoders, which learn to extract shared and correlated hidden representations from different modalities.

1) Autoencoders: Autoencoders [38] are one of the most commonly used DNNs in unsupervised ML. A typical autoencoder, as shown in Fig. 2, has two building blocks, which are an *encoder* (f) and a *decoder* (g). The encoder maps unlabelled data (X) into a hidden representation (h). The decoder tries to generate a reconstruction (X') of the input data from the representation. When training an autoencoder, the objective is to minimize the difference between X and



(a) Split autoencoder

(b) Canonically correlated autoencoder

Fig. 3: In split autoencoders (a), for aligned input (X_A, X_B) from two modalities, data from one modality are input into its encoder to generate an h, which is then used to reconstruct the data for both modalities through two decoders. Each single modality has a loss function (*i.e.*, L_A and L_B) and the overall objective of training is to minimize $L_A + L_B$. In a canonically correlated autoencoder (b), data from both modalities are input into their encoders to generate two representations. Two parameter matrices are used to maximize the canonical correlation between the paired representations h_A and h_B. The overall objective of the training is to minimize $\lambda(L_A + L_B) + L_C$, where λ is a trade-off parameter and L_C is the negative value of the canonical correlation.

X', which is measured by a loss function $L(X, X')$, such as the mean squared error (MSE). The assumption is that if the reconstruction error is small, then it means that the hidden representation contains the most useful information in the original input. Therefore, minimizing the error will make the encoder to learn to extract such useful information.

2) Split autoencoders: Canonical autoencoders only work on data from the same modality. In order to extract shared representations from multimodal data, Ngiam *et al.* [9] propose a split autoencoder (SplitAE) that takes input data from one modality and encode the data into a shared h for two modalities. With the shared h, two decoders are used to generate the reconstructions for two modalities. Fig. 3a shows the structures of SplitAEs for two data modalities. The premise is that the data from two modalities have to be matching pairs, which means that they present the same underlying activities or events. Since the encoders for both modalities aim to extract hidden representations, we want the representations to be not only specific to an individual modality. Instead, we hope that the extracted representations from both encoders can reflect the general nature of the activities or events in question.

For modalities A and B, given a pair of matching samples (X_A, X_B) (*e.g.*, accelerometer data and video data of the same activity), the SplitAE (f_A, g_A, g_B) for input modality A is:

$$\underset{f_A, g_A, g_B}{\arg\min}\; L_A(X_A, X'_A) + L_B(X_B, X'_B) \tag{1}$$

X'_A and X'_B are the reconstructions for two modalities. L_A and L_B are the loss functions for two modalities, respectively. By minimizing the compound loss in Eq. 1, the learned encoder f_A will extract representations that are useful for both modalities. Similarly, for input modality B, its SplitAE is (f_B, g_A, g_B).

3) Deep canonically correlated autoencoders: In order to combine deep canonical correlation analysis [39] and autoencoders together, Wang *et al.* [10] propose a deep canonically correlated autoencoder (DCCAE). Instead of mapping multimodal data into shared representations, DCCAE keeps an individual autoencoder for each modality and tries to maximize the canonical correlation between the hidden representations from two modalities. Fig. 3b shows the structure of a DCCAE for two modalities.

For modalities A and B, given aligned input (X_A, X_B), the DCCAE (f_A, g_A, f_B, g_B) is:

$$\underset{f_A, g_A, f_B, g_B, U, V}{\arg\min}\; \lambda(L_A + L_B) + L_C \tag{2}$$

$$L_C = -\mathrm{tr}(U^{\intercal} f_A(X_A) f_B(X_B)^{\intercal} V) \tag{3}$$

Parameter matrices U and V are canonical correlation analysis directions. Similarly to SplitAE, one of the objectives of DCCAE is to minimize the reconstruction losses. In addition, it uses another objective to increase the canonical correlation between the generated representations from two modalities (*i.e.*, minimizing its negative value L_C). The two objectives are balanced by a parameter λ. By this means, DCCAE maps multimodal data into correlated representations rather than shared representations.

C. Multimodal federated averaging

During each round t, the server sends a global multimodal autoencoder $w_t^{a_g}$ to selected clients. A selected client is either unimodal or multimodal and the local training on $w_t^{a_g}$ depends

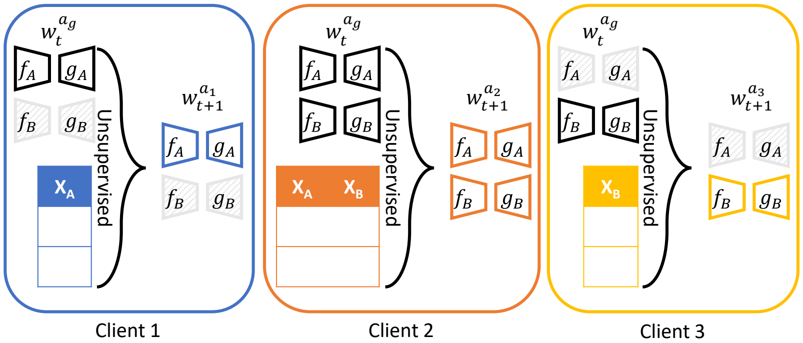


Fig. 4: Multimodal local training. Clients only update the f and g that are related to the modalities of their data.

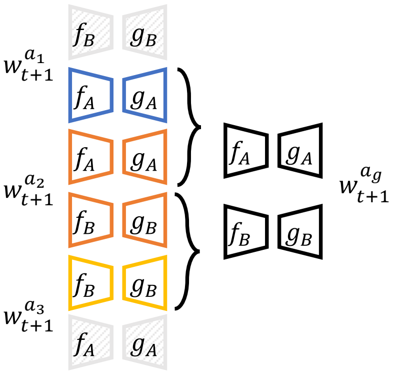


Fig. 5: Multimodal FedAvg on the server. Only the updated parts of each local model will be aggregated.

Algorithm 2 Multimodal FedAvg (Mm-FedAvg)

```
Require: $W_t$: local multimodal autoencoders at round $t$; $\alpha$: multimodal weight parameter; $n^k$: number of samples on client $k$; $m^k$: data modality of client $k$;
$W_t^A \leftarrow \{w^{a_k} | w^{a_k} \in W_t \wedge m^k = A\}$
$W_t^B \leftarrow \{w^{a_k} | w^{a_k} \in W_t \wedge m^k = B\}$
$W_t^{AB} \leftarrow \{w^{a_k} | w^{a_k} \in W_t \wedge m^k = AB\}$
$n_A \leftarrow \sum_{w^{a_k} \in W_t^A} n^k + \alpha \sum_{w^{a_k} \in W_t^{AB}} n^k$
$n_B \leftarrow \sum_{w^{a_k} \in W_t^B} n^k + \alpha \sum_{w^{a_k} \in W_t^{AB}} n^k$
$(f_A, g_A) \leftarrow \sum_{w^{a_k} \in W_t^A} \frac{n^k}{n_A}(f_A, g_A)^k + \alpha \sum_{w^{a_k} \in W_t^{AB}} \frac{n^k}{n_A}(f_A, g_A)^k$
$(f_B, g_B) \leftarrow \sum_{w^{a_k} \in W_t^B} \frac{n^k}{n_B}(f_B, g_B)^k + \alpha \sum_{w^{a_k} \in W_t^{AB}} \frac{n^k}{n_B}(f_B, g_B)^k$
$w_{t+1}^{a_g} \leftarrow (f_A, g_A, f_B, g_B)$
```

on the modality of data on the client. As shown in Fig. 4, a multimodal client (*e.g.*, Client 2) locally updates the encoders and decoders for both modalities. A unimodal client (*e.g.*, Client 1 or 3) only updates the encoder and decoder for its data modality through standard autoencoder training. The encoder and decoder for the other modality will be frozen during the local training.

We propose a multimodal FedAvg (Mm-FedAvg) algorithm to aggregate autoencoders received from both unimodal clients and multimodal clients. Fig. 5 shows which parts of different local autoencoders are used when generating a new global model. Given a global multimodal autoencoder $w_t^{a_g}$ at round t represented as $(f_A, g_A, f_B, g_B)_t$, $(f_A, g_A)_t$ is the encoder and decoder for modality A. Similarly, a local multimodal autoencoder updated by client k is $w_t^{a_k}$ and the client's modality m_k is one of A, B and AB. The Mm-FedAvg algorithm is shown in Alg. 2.

When aggregating local models from multimodal clients and unimodal clients, the contribution from multimodal clients is controlled by a weight parameter α. Increasing α can give more weights to multimodal clients because they play a key role in aligning two modalities, which helps unimodal clients benefit from the data from another modality.

IV. Evaluation

We evaluate our proposed framework on different multimodal datasets including sensory data, depth camera data, and RGB camera data through simulations. The research questions that we want to answer are as follows:

- Q1. Does introducing data from multiple modalities into FL improve its performance?
- Q2. Does a classifier trained on labelled data from one modality work on testing data from other modalities?
- Q3. Does learning from both unimodal and multimodal clients provide better performance than only learning from multimodal clients?

A. Datasets

As human activity recognition (HAR) is a domain that often relies on multimodal data, we used three HAR datasets that contain IoT data from different modalities in our experiments. Table I shows the modalities, X sizes, h sizes, and the number of classes in the datasets.

1) Different sensory modalities: The Opportunity (Opp) challenge dataset [42] contains 18 short-term and non-repeated kitchen activities including *opening & closing doors, fridges, dishwashers, and drawers, cleaning tables, drinking from cups,*

TABLE I: USED MULTIMODAL DATASETS

Dataset	Modality	X size	h size	Classes
Opp	Acce Gyro	24 15	10	18
mHealth	Acce Gyro Mag	9 6 6	4	13
UR Fall	Acce RGB Depth	3 512 8	2,4	3

toggling switches, and *null activities*. Its multimodal data are measured by on-body sensors including accelerometers, gyroscopes, and magnetic sensors. We use the accelerometer data (Acce) measured in $milligrams$ and gyroscope data (Gyro) measured in $degrees/s$ as the two modalities in our experiments. Following the experimental setup used by Hammerla *et al.* [43], we use the runs ADL4 and ADL5 of subjects 2 and 3 as testing data ($118k$ samples) and the remaining runs (except for ADL2 of subject 1) as training data ($525k$ samples). For *NaN* data in a sequence, we use their previous value in the sequence to replace them [42]. As the training data are from 15 runs, when generating local data for a client, the size of the randomly sampled sequence is $1/15$ of the training data.

The mHealth dataset [44] contains 13 daily living and exercise activities including *standing still, sitting & relaxing, lying down, walking, climbing stairs, waist bending forward, frontal elevation of arms, knees bending, cycling, jogging, running, jumping front & back*, and *null activities*. The activities are measured by multimodal on-body sensors including accelerometers, ECG sensors, gyroscopes, and magnetometers. We use the accelerometer data (Acce) measured in $meters/s^2$, gyroscope data (Gyro) measured in $degrees/s$, and magnetometer data (Mag) measured as local magnetic field in our experiments and test the combinations of each two of them. For each replicate of our simulations, we use the Leave-One-Subject-Out method to randomly choose one participant and use her data as testing data. The other 9 participants' data are used as training data. The average number of samples from a participant is $122{\pm}18k$ ($mean{\pm}std$). The size of the randomly sampled sequence for a client is $1/9$ of the training data.

2) Sensory-Visual modalities: The UR Fall Detection dataset [45] contains 70 video clips recorded by a RGB camera (RGB) and a depth camera (Depth) of human activities including *not lying, lying on the ground*, and *temporary poses*. Each video frame is labelled and paired with sensory data from accelerometers (Acce) measured in $grams$. We use this dataset for our experiments on sensory-visual and visual-visual modality combinations. For the modality RGB, similar to the work by Srivastava *et al.* [46], we use a pre-trained ResNet-18 [47] to convert each frame into a feature map. For the modality Depth, we use the extracted features including *HeightWidthRatio, MajorMinorRatio, BoundingBoxOccupancy, MaxStdXZ, HHmaxRatio, Height, Distance*, and *P40Ratio*, which are provided in the dataset. The size of h is 2 with Acce and is 4 without it. For each replicate of our simulations, we randomly sample $1/10$ data (*i.e.*, 7 video clips) as testing data and use the rest as training data. The average number of frames in a video clip is 164 ± 82 ($mean \pm std$). From the training data, the size of a randomly sampled sequence for a client is $1/9$ of the training data.

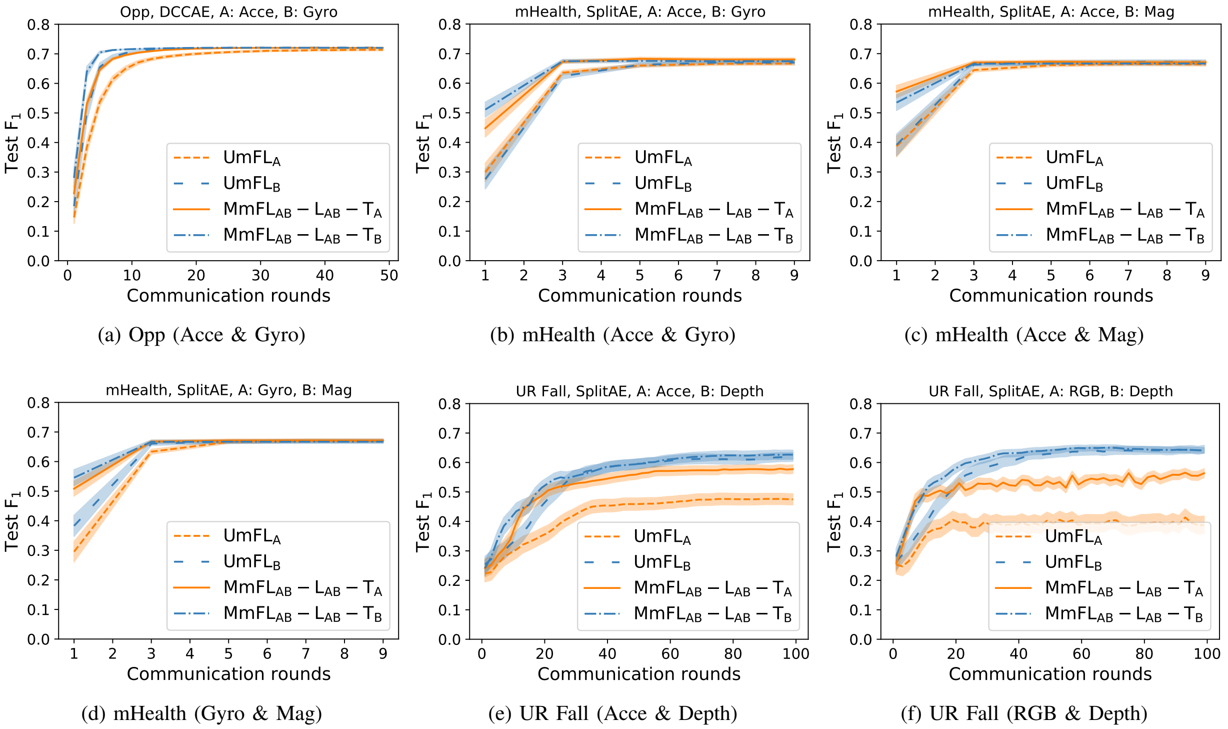
B. Simulation setup

In each replicate of our simulation, the server conducts at most 100 communication rounds with the clients and selects 10% clients for local training (2 epochs with a 0.01 or a 0.001 learning rate, whichever provides better performance) in each round, after which the cloud training (5 epochs with a 0.001 learning rate) is conducted. The labelled dataset on the server is randomly sampled from the training dataset and its size is the same as the size of a client's local data. For DCCAE, we set $\lambda = 0.01$ as suggested by Wang *et al.* [10]. For the multimodal weight parameter α, we tested $\{1, 2, 10, 50, 100, 500\}$ and found that $\alpha = 100$ provides the best performance. For each individual simulation setup, we use different random seeds to run 64 replicates.

1) Baselines: To answer Q1, we consider a system in which clients have multimodal data and a server has two labelled unimodal datasets. Without multimodal representation learning, a baseline scheme can only use data from one modality, which we refer to as **UmFL** (30 unimodal clients, 1 label modality). Comparing UmFL with our multimodal scheme (30 multimodal clients, 2 label modalities) will reveal whether introducing more modalities in FL improves its performance. We test both of them on the data from the modality of UmFL.

To answer Q2, we consider a system wherein clients have multimodal data and a server has a labelled dataset from one modality. A baseline scheme trains a global unimodal autoencoder for each modality with the same size of h. The classifier of the baseline is trained on the labelled data from one modality with the help from the autoencoder on that modality. We directly test the classifier on data from the other modality, since the sizes of h from two modalities are the same. This baseline does not use the alignment information to do any multimodal local training. It is for the ablation study on the multimodal local training and multimodal FedAvg component. We refer to this baseline as **Abl** (30 unimodal clients for each modality, 1 label modality). Comparing Abl with our scheme (30 multimodal clients, 1 label modality) will indicate whether the multimodal component brings any improvement to the performance.

To answer Q3, we consider a system that has both unimodal clients and multimodal clients. The server in the system has a labelled dataset from one modality. A baseline scheme only chooses multimodal clients (30 clients) to update the global autoencoders. Comparing it with other schemes that use both multimodal and unimodal clients for local update will show whether our proposed Mm-FedAvg improves the performance of the system.



(a) Opp (Acce & Gyro)

(b) mHealth (Acce & Gyro)

(c) mHealth (Acce & Mag)

(d) mHealth (Gyro & Mag)

(e) UR Fall (Acce & Depth)

(f) UR Fall (RGB & Depth)

Fig. 6: Comparison between UmFL and MmFL. MmFL schemes have higher or same level of converged F_1 scores on UR Fall datasets than UmFL schemes do. On all three datasets, MmFL converges faster than UmFL does.

2) Models: We implement all the deep learning components through the PyTorch library [48]. For training autoencoders on time-series data, we use long short-term memory (LSTM) [49] autoencoders [46] in our experiments for local training and use the bagging strategy [50] to train our models with random batch sizes and sequence lengths. An LSTM autoencoder takes a time-series sequence (*e.g.*, sensory data, video frames) as its input. The hidden states generated by the LSTM encoder unit are used as the hidden representations of the input samples in the sequence. On the server side, we use a simple classifier that has one multilayer perceptron (MLP) layer connected to one LogSoftmax layer as the model for supervised learning. On the mHealth dataset, we introduce a Dropout layer (rate=0.5) before the MLP layer of the classifier to prevent overfitting.

C. Metrics

We test the classifier on the server against a labelled testing dataset. We use a sliding time window with length of 2,000 to extract time-series sequences (without overlap) from the testing dataset. We use the encoder of w^{a_g} for the modality of the testing data to convert the sequences into representations and test them on the classifier w^s. We calculate the F_1 score of each class within a sequence as:

$$F_1 = \frac{2 * TP}{2 * TP + FP + FN} \tag{4}$$

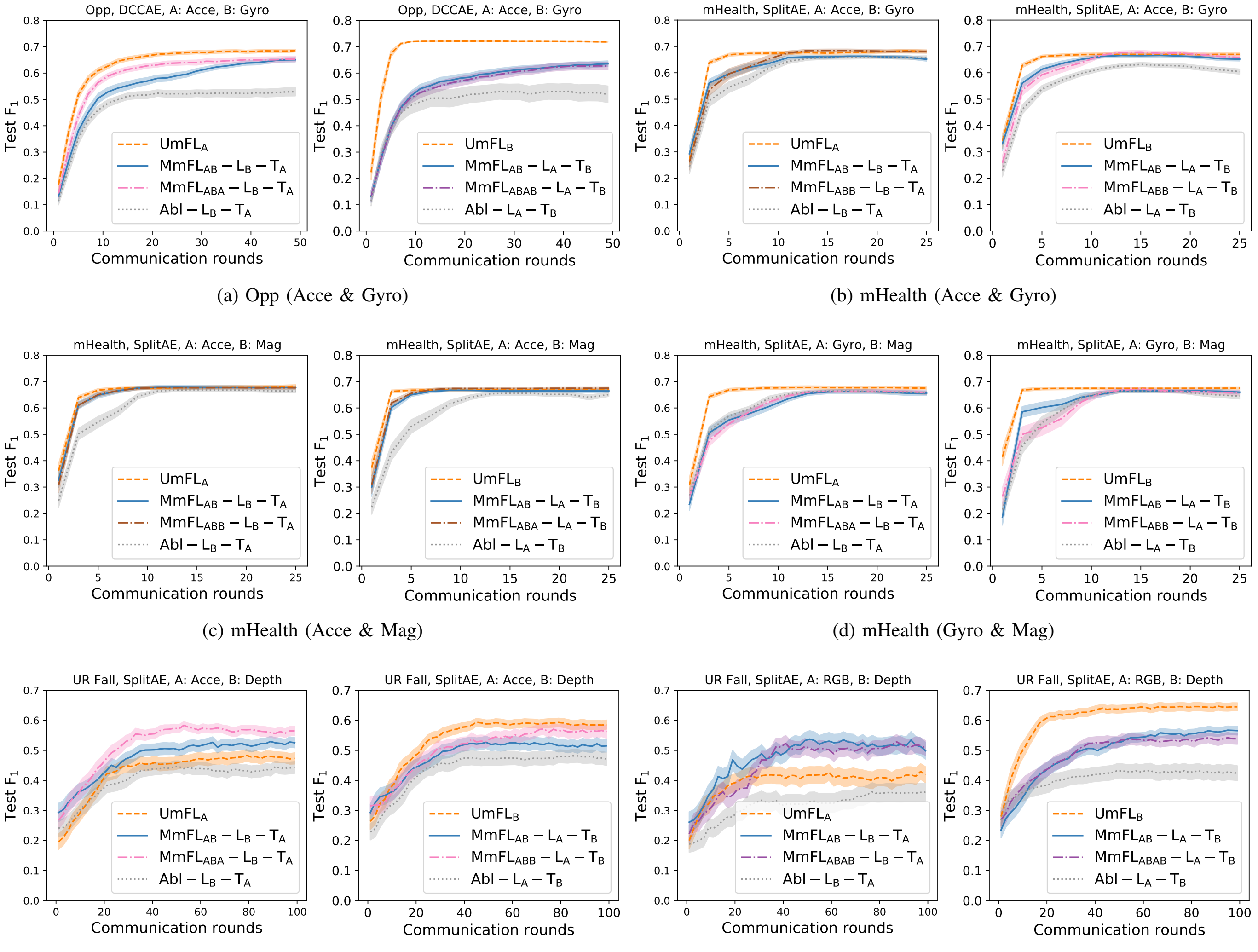
TP, FP, and FN are the numbers of true positive, false positive, and false negative classification results, respectively. The weighted average F_1 score of all classes within the sequence (with the number of ground truth samples of a class being its weight) is the F_1 score on the sequence. And the average F_1 score of all sequences is the F_1 score of the classifier. We evaluate the F_1 score of the classifier every other communication round until it converges and calculate its average value and standard error from 64 replicates. On each dataset, we evaluate both SplitAE and DCCAE and keep the one that has better F_1 scores.

V. Results

We find that by using data from multiple modalities, the F_1 score of the classifier is higher than that by using data from one single modality. With the help of multimodal representations, the classifier trained on labelled data from one modality can be used on the data from another modality and achieve acceptable F_1 scores. In addition, combining local autoencoders from both unimodal and multimodal clients can achieve higher F_1 scores than only using multimodal clients.

A. Multimodal data improve F_1 scores

On the Opp dataset, as shown in Fig. 6a, the F_1 scores of multimodal schemes (MmFL) that are trained on labelled datasets from two modalities (L_{AB}) converge faster than $UmFL_A$ and $UmFL_B$ do when being tested on each modality (T_A and T_B). Although the converged F_1 scores are the same for both UmFL and MmFL, using multimodal data speeds up the convergence.



(a) Opp (Acce & Gyro)

(b) mHealth (Acce & Gyro)

(c) mHealth (Acce & Mag)

(d) mHealth (Gyro & Mag)

(e) UR Fall (Acce & Depth)

(f) UR Fall (RGB & Depth)

Fig. 7: F_1 scores of MmFL with labelled data from one modality (*e.g.*, $\mathrm{L_B}$) and test data from the other modality (*e.g.*, $\mathrm{T_A}$). MmFL schemes achieve higher converged F_1 scores or faster convergence than baselines (*i.e.*, Abl schemes) in most cases. Combining contributions from both unimodal and multimodal clients (*e.g.*, $\mathrm{MmFL_{ABA}}$) can further improve the F_1 scores.

On the mHealth dataset (Fig. 6b– 6d), the results on three modality combinations show similar trends. On each testing modality, the converged F_1 scores of MmFL schemes are similar to those of their unimodal counterparts. However, the F_1 scores of MmFL schemes converge faster than UmFL schemes do.

On the UR Fall dataset, the sizes of X from Acce and RGB are 3 and 512, respectively. Thus $h = 2$ is the largest representation size that we can use for the modality combination Acce & RGB and it is not large enough to encode useful representations from RGB data. Therefore we only show the results from the other two modality combinations (Fig. 6e & 6f). The F_1 scores of MmFL schemes are higher than those of UmFL schemes when the schemes are tested against Acce data or RGB data. When being tested against Depth data, MmFL schemes converge faster than UmFL schemes do. Even the modalities of data in UR Fall are more heterogeneous (*i.e.*, sensory & visual) than those in Opp or mHealth (*i.e.*, sensory & sensory), multimodal FL can still align their representations, thereby introducing more data to improve the F_1 score of the FL system.

Similar to the results of existing studies on centralized ML systems, our results demonstrate that, in FL systems, combining different modalities through multimodal representation learning can achieve higher F_1 scores or faster convergence than only using unimodal data. Compared with existing work using early fusion [14], the labelled data source on the server in our framework does not have to be aligned multimodal data. It can be individual unimodal datasets that are collected separately. This suggests that we can scale up FL systems across different modalities by utilizing the alignment information contained in local data on multimodal clients.

B. Labels can be used across modalities

To answer Q2, we use labelled data from one modality for supervised learning on the server and test the trained classifier on the other modality that does not have any labels in the system. Fig. 7 shows the F_1 scores of MmFL with different modalities for labelled data (*e.g.*, $\mathrm{L_B}$) and testing data (*e.g.*, $\mathrm{T_A}$), in comparison with a baseline scheme (Abl) for the ablation study and a unimodal scheme for the modality of the testing data (*e.g.*, $\mathrm{UmFL_A}$).

On the Opp dataset with DCCAE (Fig. 7a), using only multimodal clients (*i.e.*, $\mathrm{MmFL_{AB}}$) achieves higher converged F_1 scores than baseline schemes do, which means that the multimodal representation learning on clients indeed aligns two modalities. When training classifiers on labelled Gyro data and testing them on Acce data (*i.e.*, $\mathrm{MmFL_{AB}}$-$\mathrm{L_B}$-$\mathrm{T_A}$), the F_1 score is close to that of a unimodal scheme using Acce data (*i.e.*, $\mathrm{UmFL_A}$), which demands labelled Acce data on the server.

On the mHealth dataset (Fig. 7b–7d), the converged F_1 score of baseline schemes and unimodal schemes is close to each other. This means that the different modalities may be correlated even without being aligned (similar to the findings reported by Malekzadeh *et al.* [51]). This might be due to the fact that except for 1 accelerometer on the chest, 6 sensors for different modalities in the mHealth dataset were attached to 2 body parts (*e.g.*, left-ankle and right-lower-arm). Thus the readings of different modalities from the same body part might be correlated. $\mathrm{MmFL_{AB}}$ schemes still improve the converged F_1 scores compared to Abl schemes and have faster convergence in two modality combinations (*i.e.*, Acce & Gyro, Acce & Mag).

On the UR Fall dataset (Fig. 7e–7f), $\mathrm{MmFL_{AB}}$ schemes have higher F_1 scores than baselines do. It is worth to note that, when using labelled Depth data (*i.e.*, $\mathrm{L_B}$), the test F_1 scores on Acce and RGB data (*i.e.*, $\mathrm{MmFL_{AB}}$-$\mathrm{L_B}$-$\mathrm{T_A}$ schemes in Fig. 7e & 7f) are even higher than those when using labelled data from these two testing modalities (*i.e.*, $\mathrm{UmFL_A}$). In Sec. V-A, results in Fig. 6e & 6f show that the unimodal schemes using Depth data have higher F_1 scores than those using Acce or RGB data. Therefore, for MmFL with SplitAE, using labelled Depth data for the supervised learning on the server leads to higher F_1 scores than those using Acce or RGB data's own labels.

Our results show that, with the help of multimodal representation learning on FL clients, we can use the trained global autoencoder to share the label information from one modality to other modalities by mapping them into shared or related representations. The test F_1 scores on the other modalities can be close to or even better than those of unimodal FL schemes using labels from the modalities. This allows us to scale up FL systems even with limited source of unimodal labelled data. In addition, we can potentially improve the testing performance of a modality by aligning it with other modalities that have labels, instead of directly mapping it to labels.

C. Training on mixed clients

To understand how mixed clients with different device setups (*i.e.*, unimodal clients and multimodal clients), which is a more realistic scenario for FL systems, affect the F_1 scores, for each $\mathrm{MmFL_{AB}}$ scheme with 30 multimodal clients, we run one mixed-client scheme that has 10 more clients for modality A (*i.e.*, $\mathrm{MmFL_{ABA}}$), one that has 10 more clients for modality B (*i.e.*, $\mathrm{MmFL_{ABB}}$), and one that has 10 more clients for each modality (*i.e.*, $\mathrm{MmFL_{ABAB}}$). We compare them and keep the one that has the highest F_1 scores.

In Fig. 7a, the $\mathrm{MmFL_{ABA}}$-$\mathrm{L_B}$-$\mathrm{T_A}$ scheme on the Opp dataset further speeds up the convergence of test F_1 scores compared to $\mathrm{MmFL_{AB}}$, which means that combining contributions from both unimodal and multimodal clients by using Mm-FedAvg is better than using only multimodal clients. On the mHealth dataset (Fig. 7b & 7c), the mixed-client schemes slightly improve the test F_1 scores in two experiments. Similarly, on the UR Fall dataset (Fig. 7e), $\mathrm{MmFL_{ABA}}$ and $\mathrm{MmFL_{ABB}}$ schemes show improved F_1 scores in the experiments of the Acce & Depth combination.

The results indicate that using Mm-FedAvg to combine models from both multimodal (with higher weights) and unimodal clients can provide higher F_1 scores or faster convergence than only using multimodal clients. Thus, when there are a limited number of multimodal clients in a mixed-client FL system, we can utilize unimodal clients to boost the local training.

VI. Discussions

In this paper, we have proposed a multimodal FL framework on IoT data. We now discuss how the framework can be used in real-world FL systems and what potential research topics are in the space of multimodal FL.

A. Heterogeneity beyond data distributions

Training in FL is mainly conducted on clients. In a real-world FL system, each client's local data are generated on an individual level rather than a population level, which means that heterogeneity between clients is commonplace. Some heterogeneity such as data distributions has been well studied and solving it can help keep the performance of FL systems stable across different clients. Other heterogeneity, such as data modalities, is also an important issues in implementing FL systems. As shown in our results, solving such heterogeneity can make FL systems scalable across different modalities, thereby increasing the amount of available data. In an FL system using IoT devices, it is difficult to force all clients to deploy devices that have the same data modality, because users may have different budgets for devices or privacy concerns on the devices installed in their homes. Therefore, multimodal FL plays an important role in realizing those promised FL systems that aim to work with hundreds of thousands of clients. In this paper, we focused on the modality heterogeneity issue and the other types of heterogeneity are out of our scope, which is the limitation of this paper. For future research, we plan to investigate how multimodal FL performs with the influence from the other types of heterogeneity in aspects such as data distributions and DNN model structures.

B. Sharing label information across modalities

The lack of labelled data on FL clients has recently motivated researchers to design semi-supervised FL systems. In many cases, only the service provider (*i.e.*, the FL server) has the ability and expertise to provide labelled data. The existing research on semi-supervised FL assumes that the labelled data on the server and the local data on clients are from the same modality. In this paper, we have shown that our framework allows label information from one modality to be used by other modalities. This can potentially contribute to reducing the cost of data annotation on the server when implementing real-world semi-supervised FL systems. Some modalities (*e.g.*, sensory data) may not be easy to directly annotate on. However, by using the matching information on FL clients, we can align these modalities with other modalities that are easy to acquire annotations (*e.g.*, visual data) on the server. By this means, we can enable clients from all modalities in the system to utilize the label information through multimodal representations. It may also allow us to deploy fewer privacy-intrusive devices (*e.g.*, cameras) in people's homes since we only need some clients to have multimodal data for alignment.

C. Utilizing mixed FL clients

One of our contributions in this paper is the Mm-FedAvg algorithm that combines locally updated autoencoders from both unimodal and multimodal clients. By giving multimodal clients more weights, combining contributions from mixed clients has higher F_1 scores than only using multimodal clients. Thus only a part of the clients in the system needs to be multimodal clients. Currently, all the multimodal clients in the framework use the same type of autoencoder (*i.e.*, either all SplitAE or all DCCAE) and the unimodal clients' can directly update a part of the autoencoders. In reality, this assumption may need to be changed due to different local data distributions or computational capabilities. Therefore, we suggest that more flexible multimodal averaging algorithms using techniques such as knowledge distillation [36] should be investigated. It would allow FL systems to use different local autoencoders for multimodal representation learning. In addition, mechanisms that can evaluate the quality of models trained on different data modalities and can dynamically adjust the weights of multimodal clients are necessary, which will allow us to optimise the combined contributions.

VII. Conclusions

As a new system paradigm, federated learning (FL) has shown great potentials to realize deep learning systems in the real world and protect the privacy of data subjects at the same time. In this paper, we propose a multimodal and semi-supervised framework that enables FL systems to work with clients that have local data from different modalities and clients with different device setups (*i.e.*, unimodal clients and multimodal clients). Our experimental results demonstrate that introducing data from multiple modalities into FL systems can improve their classification F_1 scores. In addition, it allows us to apply models trained on labelled data from one modality to testing data from other modalities and achieve decent F_1 scores. It only requires a part of the clients to be multimodal in order to align different modalities. We believe that our contributions can help machine-learning system designers who want to implement FL in complex real-world scenarios such as IoT environments, wherein data are generated from different modalities. For future research, we plan to investigate broader applications of our framework in domains apart from multimodal human activity recognition.

Acknowledgement

This work was supported by the UK Dementia Research Institute.

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[2] U. M. Aïvodji, S. Gambs, and A. Martin, "IOTFLA : A Secured and Privacy-Preserving Smart Home Architecture Implementing Federated Learning," in *Proceedings of the 2019 IEEE Security and Privacy Workshops (SPW)*, 2019, pp. 175–180.

[3] B. Liu, L. Wang, M. Liu, and C.-Z. Xu, "Federated Imitation Learning: A Novel Framework for Cloud Robotic Systems With Heterogeneous Sensor Data," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3509–3516, 2020.

[4] Y. Zhao, H. Haddadi, S. Skillman, S. Enshaeifar, and P. Barnaghi, "Privacy-preserving activity and health monitoring on databox," in *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, 2020, p. 49–54.

[5] Q. Wu, K. He, and X. Chen, "Personalized Federated Learning for Intelligent IoT Applications: A Cloud-Edge Based Framework," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 35–44, 2020.

[6] J. Pang2021, Y. Huang, Z. Xie, Q. Han, and Z. Cai, "Realizing the Heterogeneity: A Self-Organized Federated Learning Framework for IoT," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3088–3098, 2021.

[7] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A Survey on Federated Learning for Resource-Constrained IoT Devices," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[8] A. Brunete, E. Gambao, M. Hernando, and R. Cedazo, "Smart Assistive Architecture for the Integration of IoT Devices, Robotic Systems, and Multimodal Interfaces in Healthcare Environments," *Sensors*, vol. 21, no. 6, 2021.

[9] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal Deep Learning," in *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 689–696.

[10] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On Deep Multi-View Representation Learning," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 2015, pp. 1083–1092.

[11] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Berkeley MHAD: A Comprehensive Multimodal Human Action Database," in *Proceedings of the 2013 IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 2013, pp. 53–60.

[12] V. Radu, C. Tong, S. Bhattacharya, N. D. Lane, C. Mascolo, M. K. Marina, and F. Kawsar, "Multimodal Deep Learning for Activity and Context Recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, Jan. 2018.

[13] T. Xing, S. S. Sandha, B. Balaji, S. Chakraborty, and M. Srivastava, "Enabling Edge Devices that Learn from Each Other," in *Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking*, 2018, pp. 37–42.

[14] P. P. Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency, "Think Locally, Act Globally: Federated Learning With Local And Global Representations," 2020, arXiv: 2001.01523.

[15] F. Liu, X. Wu, S. Ge, W. Fan, and Y. Zou, "Federated Learning for Vision-and-Language Grounding Problems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 11 572–11 579.

[16] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, "Federated Semi-supervised Learning with Inter-client Consistency," 2020, arXiv: 2006.12097.

[17] B. van Berlo, A. Saeed, and T. Ozcelebi, "Towards Federated Unsupervised Representation Learning," in *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, 2020, p. 31–36.

[18] Y. Zhao, H. Liu, H. Li, P. Barnaghi, and H. Haddadi, "Semi-supervised Federated Learning for Activity Recognition," 2021, arXiv: 2011.00851.

[19] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[20] J. Chen and X. Ran, "Deep Learning With Edge Computing: A Review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.

[21] Y. Liu, X. Yuan, R. Zhao, Y. Zheng, and Y. Zheng, "RC-SSFL: Towards Robust and Communication-efficient Semi-supervised Federated Learning System," 2020, arXiv: 2012.04432.

[22] Z. Zhang, Z. Yao, Y. Yang, Y. Yan, J. E. Gonzalez, and M. W. Mahoney, "Benchmarking Semi-supervised Federated Learning," 2021, arXiv: 2008.11364.

[23] Z. Long, L. Che, Y. Wang, M. Ye, J. Luo, J. Wu, H. Xiao, and F. Ma, "FedSiam: Towards Adaptive Federated Semi-Supervised Learning," 2021, arXiv: 2012.03292.

[24] W. Zhang, X. Li, H. Ma, Z. Luo, and X. Li, "Federated Learning for Machinery Fault Diagnosis with Dynamic Validation and Self-supervision," *Knowledge-Based Systems*, vol. 213, p. 106679, 2021.

[25] Y. Kang, Y. Liu, and T. Chen, "FedMVT: Semi-supervised Vertical Federated Learning with MultiView Training," 2020, arXiv: 2008.10838.

[26] B. Wang, A. Li, H. Li, and Y. Chen, "GraphFL: A Federated Learning Framework for Semi-Supervised Node Classification on Graphs," 2020, arXiv: 2012.04187.

[27] D. Yang, Z. Xu, W. Li, A. Myronenko, H. R. Roth, S. Harmon, S. Xu, B. Turkbey, E. Turkbey, X. Wang *et al.*, "Federated Semi-Supervised Learning for COVID Region Segmentation in Chest CT using Multi-National Data from China, Italy, Japan," *Medical Image Analysis*, vol. 70, p. 101992, 2021.

[28] A. Saeed, T. Ozcelebi, and J. Lukkien, "Multi-task Self-Supervised Learning for Human Activity Detection," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 2, Jun. 2019.

[29] 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, 2021.

[30] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and Open Problems in Federated Learning," 2021, arXiv: 1912.04977.

[31] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[32] V. Smith, C. K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated Multi-Task Learning," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[33] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated Learning with Non-IID Data," 2018, arXiv: 1806.00582.

[34] R. Li, F. Ma, W. Jiang, and J. Gao, "Online Federated Multitask Learning," in *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 215–220.

[35] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[36] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble Distillation for Robust Model Fusion in Federated Learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 2351–2363.

[37] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in A Neural Network," 2015, arXiv: 1503.02531.

[38] P. Baldi, "Autoencoders, Unsupervised Learning and Deep Architectures," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, Bellevue, Washington, USA, 2012, pp. 37–49.

[39] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep Canonical Correlation Analysis," in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, USA, 2013, pp. 1247–1255.

[40] A. Karpathy and L. Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3128–3137.

[41] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal Machine Learning: A Survey and Taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2019.

[42] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. d. R. Millán, and D. Roggen, "The Opportunity Challenge: A Benchmark Database for On-Body Sensor-based Activity Recognition," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 2033–2042, 2013.

[43] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016, pp. 1533–1540.

[44] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga, "mHealthDroid: A Novel Framework For Agile Development of Mobile Health Applications," in *Proceedings of the 6th International Work-Conference on Ambient Assisted Living and Daily Activities*, 2014, pp. 91–98.

[45] B. Kwolek and M. Kepski, "Human Fall Detection on Embedded Platform Using Depth Maps and Wireless Accelerometer," *Computer Methods and Programs in Biomedicine*, vol. 117, no. 3, pp. 489–501, 2014.

[46] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised Learning of Video Representations Using LSTMs," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 2015, p. 843–852.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[49] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[50] Y. Guan and T. Plötz, "Ensembles of Deep LSTM Learners for Activity Recognition using Wearables," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1–28, Jun. 2017.

[51] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "DANA: Dimension-Adaptive Neural Architecture for Multivariate Sensor Data," 2020, arXiv: 2008.02397.