Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Federated model with contrastive learning and adaptive control variates for Human activity recognition*

Ignatius IWAN, Bernardo nugroho YAHYA‡, Seoklyong LEE

Industrial and Management Engineering Department, Hankuk University of Foreign Studies, Gyeonggi 17035, Republic of Korea

Email: mania087@gmail.com, bernardo@hufs.ac.kr, sllee.hufs.ac.kr

Received Sept. 13, 2024; Revision accepted Jan. 8, 2025; Crosschecked

Abstract: The recent attention to privacy issues demands a communication-safe method for training Human Activity Recognition (HAR) models on client activity data. Federated Learning (FL) has become a compelling technique to facilitate model training between the server and clients while preserving data privacy. However, classical FL methods often assume Independent and Identically Distributed (IID) data among clients. This assumption does not hold true in practical scenarios. Human activity in real world scenarios exhibits variation, resulting in skewness where identical activities are executed uniquely across clients. It leads to local model objectives drifting away from the global model objective, thereby impeding overall convergence. To address this challenge, we propose FedCoad, a novel federated model leveraging contrastive learning with adaptive control variates to handle the skewness among HAR clients. Model contrastive learning minimizes the gap in representation between global and local models to help global model convergence. During local model updates, the adaptive control variates penalize the local model updates with regard to the model weight and the rate of change from the control variates update. Our experiments show that FedCoad outperforms other state-of-the-art FL algorithms in HAR benchmark datasets.

Key words: Federated learning; Human activity recognition; Contrastive learning; Deep learning
https://doi.org/10.1631/FITEE.2400797 **CLC number:**

Unedited

1. Introduction

Sensor-based Human Activity Recognition (HAR) using wearable devices is critical in user-centered applications like healthcare (Wu et al., 2022) , smart environments (Bianchi et al., 2019) , and fall detection (Mrozek et al., 2020) . Federated Learning (FL) (McMahan et al., 2017) offers a decentralized and privacy-preserving solution for training models across devices distributed among clients. The FL method works well when the client data are independent and identically distributed (IID) as most of the clients will have similar quantities, labels, and features of data. However, humans perform various activities and even have their unique patterns. Therefore, real-world HAR data are inherently non-IID since they exhibit some skewness that can degrade model performance.

There are three types of HAR skewness in real-world conditions (Presotto et al., 2022) :*feature distribution skew*, where two clients are doing the same activity with different patterns (e.g., walking patterns differ between younger and older individuals); *quantity distribution skew*, where clients have imbalanced labeled data volumes; *label distribution skew*, where two clients have different sets of labels. For example, an athlete has more activity labels related to sports than has an office worker. These conditions lead to divergent local objectives during training and hinder global model

‡ Corresponding author

* Project supported by Hankuk University of Foreign Studies Research Fund and the Ministry of Trade, Industry and Energy (MOTIE) and Korea Institute for Advancement of Technology (KIAT) through the International Cooperative R&D program (No. P0022316)

ORCID: Ignatius IWAN, https://orcid.org/0009-0002-4447-0058; Bernardo nugroho YAHYA, https://orcid.org/0000-0002-7121-2436; Seoklyong LEE, https://orcid.org/0000-0002-8630-5395

© The Author(s) 2025

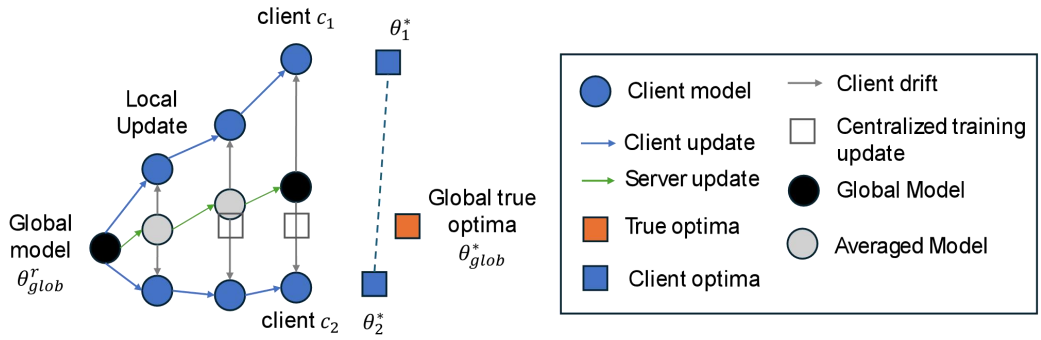


Fig. 1 Client update variance affects the global model to reach global true optima.

optimization (Karimireddy et al., 2020; X. Li et al., 2019).

Existing methods have sought to address these non-IID challenges. Some methods focus on the model aggregation stage to handle non-IID issues, such as implementing momentum for model aggregation normalization (Hsu et al., 2019) or clustering clients with similar data distributions for grouped updates (J. Guo et al., 2024; Presotto et al., 2022) . However, normalization during model aggregation may fail, as the local updates are already skewed and clustering operations require significant computational resources, especially as the number of clients grows. Other methods focus on the local training stage, such as FedProx which applies ℓ_2-norm regularization to constrain local updates (T. Li et al., 2018) and SCAFFOLD which uses control variates to reduce update variance (Karimireddy et al., 2020) . While effective in some cases, these methods risk local model overfitting due to the limited diversity of client data in HAR.

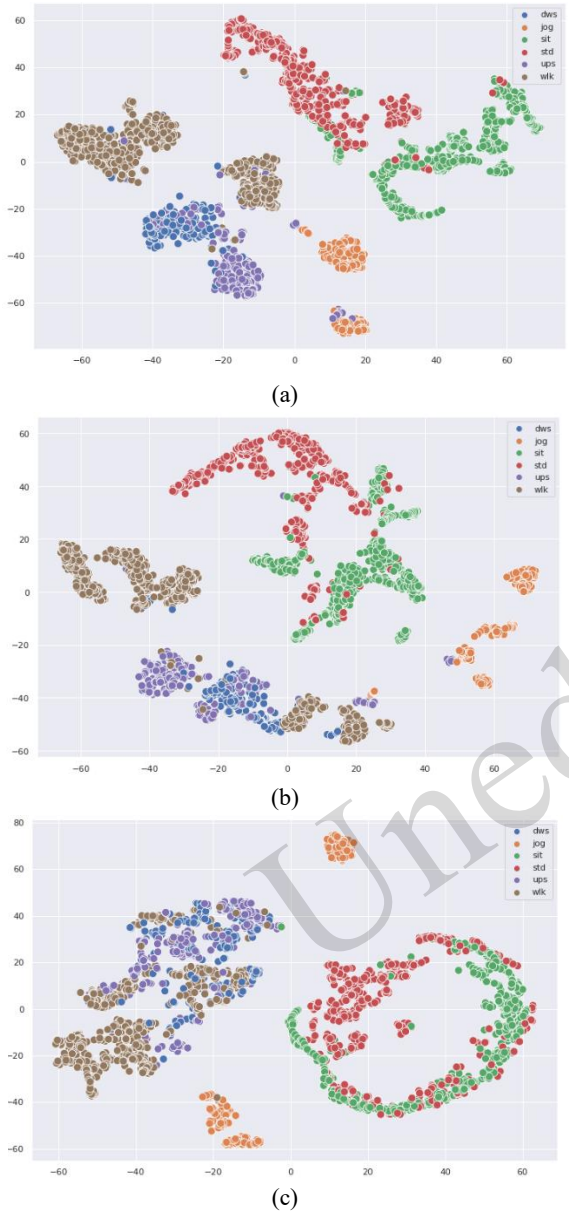
Fig. 1 illustrates the problem where two heterogeneous clients train together in an FL manner compared to a centralized training update. Unlike centralized training, which uses a comprehensive dataset to achieve the global optimum, FL relies on locally skewed datasets. Local models trained on heterogeneous data diverge from the global objective, and the aggregated global model often fails to generalize. In FL, the local clients learn only from their own dataset and overlook other clients' underlying data distributions, leading to updates that are misaligned with the true global optima.

An additional approach to mitigate the local model overfitting issue by giving an auxiliary perspective, like supplementary data to generalize, or convergence direction correction, can be helpful to prevent local model overfitting. Recently, contrastive learning has emerged as a promising solution to enhance generalization by encouraging models to align similar representations. It can be used in unsupervised or supervised settings. Unsupervised approaches (C. Guo et al., 2024) leverage unlabeled data, while supervised methods (C. Li et al., 2021) teach the model to create different representations for data with distinct labels. However, both types are affected by the amount of available data that can be stored in wearable devices, and each client's data reflects only individual patterns. Another approach, MOON (Q. Li et al., 2021) introduces model contrastive learning that uses the global model as an external resource for the local training guide to prevent local model overfitting. Despite its advantages, MOON focuses solely on the training phase and is susceptible to misdirection by the biased global model.

To investigate the assumption above, we conducted experiments using the Motionsense dataset by training a 1D CNN. T-SNE was used to visualize hidden vectors. Fig. 2(a) shows that a centralized model achieves well-separated representations, capturing diverse activity classes. However, unlike the result of Q. Li et al. (2021), the global model in Fig. 2(b) reveals overlapping representations in the global model trained with FL, caused by local models overfitting to skewed datasets Fig. 2(c). Therefore, sole reliance on the global model aggregation to correct local client updates may fail to guide the local model training.

To address those limitations, we propose FedCoad, a method that seamlessly unifies model

Unedited



(a) (b) (c)

Fig. 2 T-SNE visualization in the Motionsense dataset for (a) Centralized; (b) FedAvg; (c) Local model.

contrastive learning with dynamic control variates to enhance local model training stability and achieve better convergence for the global model. Unlike existing methods, which address local model divergence in either the training phase (e.g., MOON, FedProx) or the parameter update phase (e.g., SCAFFOLD), FedCoad tackles both simultaneously. During training, model contrastive learning aligns local and global model representations, mitigating discrepancies while maintaining local objectives. In the update phase, adaptive control variates penalize deviations from the global update trajectory, preventing overfitting and ensuring convergence. By addressing the divergence in both the training and parameter update phases, FedCoad ensures that local models avoid overfitting and achieve superior performance in non-IID environments. To summarize, the contributions of this work are as follows:

1. We propose a new FL method that maximizes the synergy of model contrastive learning and control variates to prevent local model divergence in a non-IID environment.
2. The proposed FedCoad handles the local model divergence in both the ocal training and the local model update phases.
3. Our proposed method can significantly reduce the communication exchange cost compared to the HAR state-of-the-art method.
4. We performed extensive experiments and ablation studies in popular HAR benchmark datasets and showed the superiority of the proposed FedCoad.

The rest of this paper is structured as follows. Section 2 summarizes related studies in FL for non-IID environments. Section 3 explains the inner workings of FedCoad. Section 4 introduces the dataset and describes the experimental setup and results. Section 5 concludes the paper.

2. Related studies

In this section we give an overview of popular FL algorithms, summarized in Table 1. FedAvg (McMahan et al., 2017) was the first and is the most popular FL algorithm to train a global model with multiple clients. The popularity of FedAvg has catalyzed the adoption of FL technology within the realm of HAR research, as shown by recent studies (Sozinov et al., 2018), which have started to explore the applicability of FL to HAR for distributed training. However, one of the problems in FL for HAR is that each client's data are different and they follow non-IID patterns. A recent study by (X. Li et al., 2019) found that the non-IID condition affects FedAvg and decreases its performance because each

Unedited

Table 1 List of the state-of-the-art FL algorithms and current limitation

Reference	Methodology	Limitation
(McMahan et al., 2017)	FedAvg	Performance decreases due to the clients' skewness in non-IID environments
(T. Li et al., 2018)	FedProx	The penalty from regularization may affect the local model convergence
(Hsu et al., 2019)	FedAvgM	Encountered issues when local model is skewed before global model aggregation
(Karimireddy et al., 2020)	SCAFFOLD	The training objectives lack a previous local model as reference to prevent overfitting
(Q. Li et al., 2021)	MOON	Global model may fail to guide the local model learning in some cases
(C. Li et al., 2021)	METAHAR	Limited labeled data may introduce bias to local model representation.

local model trains on its own skewed dataset. Subsequently, some studies have attempted to improve FedAvg by handling non-IID conditions with improvement categorizations such as (1) on aggregation (Wang et al., 2020), (Hsu et al., 2019) and (2) on local training (Karimireddy et al., 2020;, Q. Li et al., 2021; T. Li et al., 2018).

The work that handles the global model aggregation step involves the server normalizing the aggregation updates during the global model update. FedMA (Wang et al., 2020) matches and averages weights in a layer-wise manner using a Bayesian non-parametric method. FedAvgM (Hsu et al., 2019) introduces momentum when aggregating the local model weights for the global model updates. However, normalizing the weights during model aggregation may fail as their local model updates already overfit to their skewed local environment before the aggregation. Another way is to separate the clients based on their characteristics and provide FL training for similar clients. For instance, the server can cluster the clients (J. Guo et al., 2024; Presotto et al., 2022b) based on performance comparison or similarity metrics. Instead of having a single global model, each cluster has its cluster model, which is exchanged and updated by the clients in the respective cluster. However, clustering-based methods have a computational issue since the clustering operation performs iterative comparisons between clients. Moreover, when the number of clients increases, the computation requirement will scale to the number of clients.

The work that handles local training involves the clients controlling the local updates to prevent the local model from being too skewed to their local condition by using a regularization term. FedProx (T. Li et al., 2018) presents a penalty term for the local training objective. The term uses ℓ_2 -norm to calculate the distance between the local model weights and the global model weights during local model training. Rather than regularizing the local model during training, SCAFFOLD (Karimireddy et al., 2020) controls the local model parameter updates by using control variates. Both a server and clients have their control variates. After the training, the client uses the difference between those control variates to adjust the parameters of the local model. Generally, all methods in the second category try to control the local model update but their capability to prevent the local model from overfitting is limited as they have the perspective only from their skewed dataset. Also, both methods regularize the local model only in either the local training or local model parameters update.

Instead of directly learning from the client-labeled data that can introduce bias, some FL studies adopted contrastive learning (C. Guo et al., 2024; C. Li et al., 2021) that compares data points and obtains a general representation inside the underlying structure of the data. In general, contrastive learning methods can be deployed in unsupervised and supervised settings. The unsupervised setting uses the unlabeled data of the clients to guide the model training. For example, ModCL (C. Guo et al., 2024) is one of the HAR state-of-the-art methods that first pre-train the model on unlabeled data using inter-modality and intra-modality consistency in a contrastive manner before adapting to the clients. Although successful, this method requires the client to have a lot of unlabeled data because learning can be unstable with small amounts of data. Clients' wearable devices often have little storage and insufficient computation to train with massive data over a long period. The supervised setting uses the

Unedited

Table 2 Notations occurred in this paper

Notation	Description	Notation	Description
C	Set of clients	r	Round of the federated learning
$C_i \in C$	Client at index i	x	Feature instances of a dataset
N	Total number of clients	y	Label instances of a dataset
D	Set of datasets	ρ_{glob}	Control variate of the global model
$\mathcal{D}_i \in D$	Dataset of client at index i	ρ_i	Control variate of client at index i
θ	Model weight	Δ_i	Weight difference between the previous and updated control variate of client at index i
θ_i^r	Local model weight of client i at round r	δ_{sup_i}	Supervised cross entropy loss of client at index i
θ_{glob}^r	Global model weight of at round r	δ_{con_i}	Model contrastive loss of client at index i
$F_\theta(.)$	Output of the whole model with weight θ	δ_{local_i}	Total local loss of client at index i
$J_\theta(.)$	Output of the model with weight θ before the output layer	δ_{reg_i}	Regularized total local loss of client at index i
$P_{\mathcal{D}_i}(x)$	Feature distribution of dataset $\mathcal{D}_i$	α	Learning rate
$P_{\mathcal{D}_i}(y)$	Label distribution of dataset $\mathcal{D}_i$	μ	Hyperparameter for controlling the influence of model contrastive loss
		τ	Temperature of model contrastive loss

Unedited

labeled data information and guides the model to learn representations that can differentiate them through contrastive learning. For example, MetaHAR (C. Li et al., 2021) pre-trains a global encoder model with pairwise loss before fine-tuning the global encoder into the clients. However, there are usually only small amounts of client-labeled data and it is often biased towards individual patterns.

In the image domain, the MOON technique involves model contrastive learning using the global model as another local training guide to prevent local model overfit and learning generalized representations. Even though MOON (Q. Li et al., 2021) falls into the supervised contrastive learning category, this model tries to handle this issue by looking at the global model representation as another perspective to adjust the local model updates. As the global model learns from various clients, it contains general knowledge that can handle heterogeneity. MOON proposes conducting model contrastive learning by comparing the global model and the previous round local model output representation as the drift control. This approach is successful in the image domain but the global model may fail to guide the local model update in HAR, as the global model itself is the averaging result of skewed client local models.

Our proposed approach falls into the second category which handles non-IID by improving the client's local training. To handle the limitation of model contrastive learning, this approach penalizes the local training loss by regulating them with control variates that take the previous local and global model updates into consideration.

3. Methodology

This section introduces the problem statement in this study and describes the FedCoad framework. Table 2 lists the notations that are used throughout this paper.

3.1 Preliminary

In FL, there are N clients, denoted as $C = \{C_1, ..., C_N\}$. Client C_i has a local dataset $\mathcal{D}_i = \{(x, y)\}$, where x is a feature data and y the activity label. Let's consider a set of datasets $D = \{\mathcal{D}_1, ..., \mathcal{D}_N\}$ where each dataset corresponds to a client in C. D is non-IID if there exists a pair of datasets $\mathcal{D}_i, \mathcal{D}_k \in D$ that follows one of the conditions below:

Feature distribution skew: $P_{\mathcal{D}_i}(x) \neq P_{\mathcal{D}_k}(x)$ is a condition where there exists an inequality between the probability distributions of $\mathcal{D}_i$ and $\mathcal{D}_k$ since the feature data in $\mathcal{D}_i$ have a significantly marginal distribution compared to the feature data in $\mathcal{D}_k$. This can happen in HAR, since everyone may perform activities in different ways. For example, an elderly

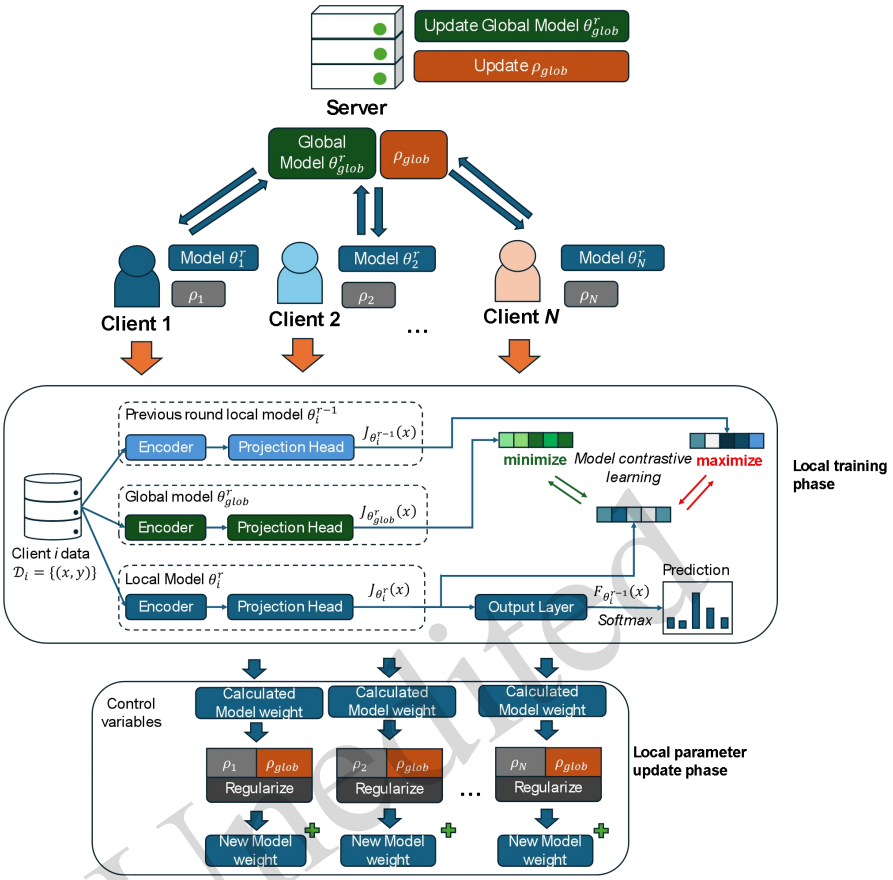


Fig. 3 Framework of FedCoad

person may have a slower walking pattern than an adolescent person.

Label distribution skew: $P_{\mathcal{D}_i}(y) \neq P_{\mathcal{D}_k}(y)$ is a condition where there exists an inequality between the probability distributions of $\mathcal{D}_i$ and $\mathcal{D}_k$ because the labels in $\mathcal{D}_i$ have a significantly marginal distribution compared to those in $\mathcal{D}_k$. In HAR, each person may have different daily activities. For example, office workers may spend more time sitting than an athlete.

Quantity distribution skew: The total number of data $|\mathcal{D}_i|$ is significantly different from $|\mathcal{D}_k|$. In HAR, since each person has a different schedule for the same activity, they have a different number of samples.

With the conditions above, the objective is to build a model with weight θ over the dataset $\mathcal{D} \triangleq \cup_{i\in\{1..N\}}\mathcal{D}_i$ without exchanging any of the client's personal training data. This work also assumes that each client learns only classes available during local training, because updating model parameters for unseen classes may harm the performance of FL (Diao et al., 2020). So, the objective is to minimize loss according to Eq (1):

$$\min_{\theta} \mathcal{L}(\theta) = \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} L_i(\theta), \qquad (1)$$

where $L_i(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\left[\delta_i\left(\theta; (x, y)\right)\right]$ is the empirical loss of C_i.

3.2 Method

This section explains FedCoad components such as the model architecture used for model contrastive learning, the inner workings of FedCoad during the local model training and parameter update phase, and the FL workflow itself in FedCoad.

3.2.1 Model architecture

The model has three components (Fig. 3): the base encoder which is responsible for extracting

Unedited

representation from data inputs, the projection head which is an additional layer that maps the encoder representation to another space with a determined dimension, and an output layer that predicts the class label based on projection head output. To formalize the formula, in this study we used the notations $F_\theta(.)$ to denote the whole model and $J_\theta(.)$ to denote the model before the output layer. For example, $J_\theta(x)$ refers to the mapped representation of data input x.

3.2.2 Local training phase

In local model training, the local model needs to optimize two losses (Fig. 3): classification loss and model contrastive loss. Classification loss refers to cross-entropy and model contrastive loss refers to metric learning loss which operates on the similarity between the representations of the current local model and global model along with the previous local model. When a client C_i starts to conduct local training, it downloads the global model θ_{glob}^r from the server and trains its local model θ_i^r at round r. For every data input x, the model obtains the representation of x from the current local model being updated $g = J_{\theta_i^r}(x)$, the representation of x from the global model $g_{glob} = J_{\theta_{glob}^r}(x)$, and the representation of x from the local model at the previous round $g_{prev} = J_{\theta_i^{r-1}}(x)$. With the assumption that the global model can produce better representations (Q. Li et al., 2021), the goal is to minimize the distance between g and g_{glob} while increasing the dissimilarity between g and g_{prev}. Therefore, to encourage g to have hard-to-learn distinctions with g_{prev}, the calculation of client C_i model contrastive loss (δ_{con_i}) follows Eq (2):

$$\delta_{con_i} \leftarrow -log \frac{e^{\frac{sim(g,\ g_{glob})}{\tau}}}{e^{\frac{sim(g,\ g_{glob})}{\tau}} + e^{\frac{sim(g,\ g_{prev})}{\tau}}}, \quad (2)$$

where τ denotes the temperature parameter that controls the penalty strength on the distance between the local model and the previous local model. The total local loss of client C_i is the sum of its classification loss (δ_{sup_i}) and model contrastive loss (Eq (3)):

$$\delta_{local_i} = \delta_{sup_i}\left(\theta_i^r; (x, y)\right) + \mu\delta_{con_i}(\theta_i^r; \theta_i^{r-1}; \theta_{glob}^r; x), \quad (3)$$

where μ is a hyper-parameter that controls the influence of contrastive loss.

3.2.3 Local parameter update phase

As clients train only with their local data, their updates tend to drift from the global true optima and pursue their local minima (Karimireddy et al., 2020). Therefore, assigning the new parameter directly after the local training hinders global model convergence. To handle this, clients need to regularize their local parameter updates after the local training. Karimireddy et al. (2020) used control variates to adjust the local model update to minimize client drift. Following in their footsteps, in this study we incorporated control variates such as local control variates denoted $\rho_1, ..., \rho_N$ and a global control variate denoted ρ_{glob}.

The purpose of the control variate is to adjust the local model update so that it does not pursue the local minima. The local control variate represents the update speed or the rate of changes between a client-trained local model and the global model. As the global control variate is the average of the local control variates, it embodies overall client update speed. Therefore, FedCoad prevents local clients from converging too fast to their local minima by factoring the divergence between overall clients (global control variance) and their update speed (local control variates).

At the first round, the server initializes the global control variate using random weight and makes copies for the client local control variate. So, the regularized local loss for the local client updates follows Eq (4):

$$\delta_{reg_i} = \delta_{local_i} + \rho_{glob} - \rho_i, \quad (4)$$

therefore, the local objective follows Eq (5):

$$\min_{\theta_i^r} \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\left[\delta_{sup_i}\left(\theta_i^r; (x, y)\right) + \mu\delta_{con_i}\left(\theta_i^r; \theta_i^{r-1}; \theta_{glob}^r; x\right) + \rho_{glob} - \rho_i\right], \quad (5)$$

Unedited

For every C_i, the parameter weight updates follow Eq (6):

$$\theta_i^r \longleftarrow \theta_i^r - \alpha \nabla \delta_{reg_i}, \quad (6)$$

where α is the learning rate and $\nabla \delta_{total}$ is the local model update. Next, C_i updates ρ_i according to Eq. (7):

$$\rho_i \leftarrow \rho_i - \rho_{glob} + (\theta_{glob}^r - \theta_i^r)/(E * \alpha), \quad (7)$$

where E is the number of local epochs and B is the batch size. After all clients send their local model updates and their local control variate rate of change Δ_i, the server will update ρ_{glob} according to Eq. (8):

$$\rho_{glob} = \sum_{k=1}^{N} \Delta_k. \quad (8)$$

3.2.4 FL algorithm

Using all the components mentioned above, FedCoad aims to build a global model by using a server to arrange FL training with clients. Details of the workflow are shown in Algorithm 1 and Algorithm 2. In the first round ($r = 1$), the server initializes the first global model weight θ_{glob}^0 and server control variates ρ_{glob} (Algorithm 1 lines 2-3). Then, the server lists N number of clients that will join the training and sends θ_{glob}^0, and other hyperparameters to C (Algorithm 1 line 6). After receiving the server transmission, each client C_i performs $LocalTraining$ using their local model θ_i^r and ρ_{glob} (Algorithm 1, line 7). Also, if it is the first round ($r = 1$), C_i will initialize their local control variates ρ_i (Algorithm 2, line 3). Inside the $LocalTraining$, client C_i calculates the prediction loss δ_{sup_i} using cross-entropy and model contrastive loss δ_{con_i}. To obtain δ_{sup_i}, C_i runs θ_i^r on their local data $\mathcal{D}_i$ to get local model prediction $F_{\theta_i^r}$ and compares it with the true label y (Algorithm 2, line 8). For the δ_{con_i}, C_i runs θ_i^r, θ_{glob}^r, and θ_i^{r-1} on $\mathcal{D}_i$ to obtain $\mathcal{g}$, $\mathcal{g}_{glob}$, and $\mathcal{g}_{prev}$ before C_i calculates δ_{con_i} (Algorithm 2, lines 9-12). Next, C_i sums δ_{con_i} and δ_{sup_i} as δ_{local_i} (Algorithm 2, line 13) which then will be penalized using ρ_{glob} and ρ_i (Algorithm 2, line 14). The penalized loss δ_{reg_i} is then used for updating θ_i^r (Algorithm 2, line 15). After the $LocalTraining$, C_i updates ρ_i values and its rate of change Δ_i (Algorithm 2, lines 18-20). Subsequently, each C_i sends the updated θ_i^r and Δ_i to the server (Algorithm 2, line 21). Finally, the server receives a list of $\{\theta_1^r, \theta_2^r, .., \theta_N^r\}$ and $\{\Delta_1, \Delta_2, ..., \Delta_N\}$ to update θ_{glob}^r and ρ_{glob} (Algorithm 1, lines 8-9).

Algorithm 1 FedCoad server workflow

Input:	number of communication rounds R, batch size B, number of local epochs E, clients C, client datasets $\mathcal{D}$, learning rate α,
Output:	final global model θ_{glob}^r
1	Server:
2	initialize θ_{glob}^0
3	initialize ρ_{glob}
4	**for** $r = 1, ..., R$ **do**
5	**for** $i = 1, ..., N$ **do**
6	send θ_{glob}^r and ρ_{glob} to C_i
7	$\theta_i^r, \Delta_i \leftarrow$ LocalTraining$(i, \theta_i^r, \rho_{glob})$
8	$\theta_{\text{glob}}^{r+1} = \sum_{k=1}^{N} \frac{\lvert\mathcal{D}_k\rvert}{\lvert\mathcal{D}\rvert} \theta_k^r$
9	$\rho_{glob} \leftarrow$ update global control variates using Eq (8)
10	**end for**
11	**end for**
12	return θ_{glob}^R

Algorithm 2 Local training workflow

Input:	client index i, client model weight θ_i^r, global control variate ρ_{glob}
Output:	updated model weight θ_i^r, local control variate rate of change Δ_i
1	LocalTraining$(i, \theta_i^r, \rho_{glob})$:
2	**if** $r = 1$ **then**
3	$\rho_i \longleftarrow \rho_{glob}$
4	**end if**
5	$\theta_i^r \longleftarrow \theta_{glob}^r$
6	**for** epoch $e = 1,2, ..., E$ **do**
7	**for** each batch $b = \{x, y\}$ of $\mathcal{D}_i$ **do**
8	$\delta_{sup_i} \leftarrow CrossEntropy(F_{\theta_i^r}, y)$
9	$\mathcal{g} = J_{\theta_i^r}(x)$
10	$\mathcal{g}_{glob} = J_{\theta_{glob}^r}(x)$
11	$\mathcal{g}_{prev} = J_{\theta_i^{r-1}}(x)$
12	$\delta_{con_i} \leftarrow$ calculate loss using Eq (2)

Unedited

$\sigma_{local_i} \leftarrow$ calculate sum of local loss using Eq (3)
$\delta_{reg_i} \leftarrow$ calculate penalized loss using Eq (4)
update local model weight θ_i^r using Eq (6)
end for
end for
$\rho_{temp} = \rho_i$
$\rho_i \leftarrow$ update local control variates using Eq (7)
$\Delta_i = \rho_i - \rho_{temp}$
return θ_i^r, Δ_i to the server

4. Experiments

This section describes the experiment and analysis results. FedCoad was evaluated on popular HAR benchmark datasets and compared with results from previous studies. The supplementary materials cover other details such as the performance measure, comparison with pretraining methods, the embedding visualization, and communication exchange comparison (see supplementary materials, Sections 1.1 – 1.4).

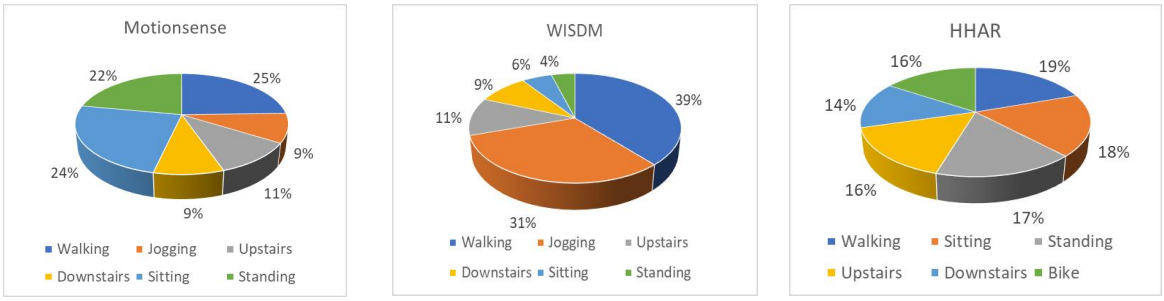
4.1 Dataset

Three benchmark datasets were used in this study:

Wireless Sensor Data Mining (WISDM) (Kwapisz et al., 2011) was a study to inspect problems in collecting sensor data embedded in mobile devices. There were 36 subjects who performed daily living activities such as standing, walking, sitting, walking upstairs, jogging, and walking downstairs. Fig. 4a shows the distribution of activities in the dataset. The subjects used a smartphone in the front pocket of their trousers while performing each activity at various times. The data were gathered from the accelerometer inside the smartphone with a sampling frequency of 20 Hz.

Motionsense (Malekzadeh et al., 2019) includes results from a study aimed at advancing the development of privacy-preservation in sensor data transmission systems. There were 24 subjects including 10 women and 14 men in the experiment. The subjects' age ranged from 18 to 46 and they had a body weight of from 48 to 102 kg. An iPhone 6s was used as the mobile device and was placed in each subject's front trouser pocket. The subjects performed activities including walking upstairs, sitting, walking, standing, jogging, and walking downstairs. Fig. 4b shows the distribution of activities in the dataset. This study used only the accelerometer signal from the mobile device for compatibility.

Heterogeneity Activity Recognition (HHAR) (Stisen et al., 2015) was a study to investigate the effect of heterogeneous mobile devices in terms of sensors, device models, and workloads on the performance of activity recognition. HHAR contains data from 9 subjects aged between 25 and 30 when performing daily living activities including walking, biking, walking upstairs, walking downstairs, sitting, and standing. Fig. 4c shows the distribution of activities in the dataset. Smartwatches were strapped around the subjects' arms, and smartphones of variable frequency were placed on their waists. The following devices were used during the HHAR study: Samsung Galaxy S3 Mini (100 Hz), Samsung Galaxy S Plus (50 Hz), LG Nexus 4 (200 Hz),



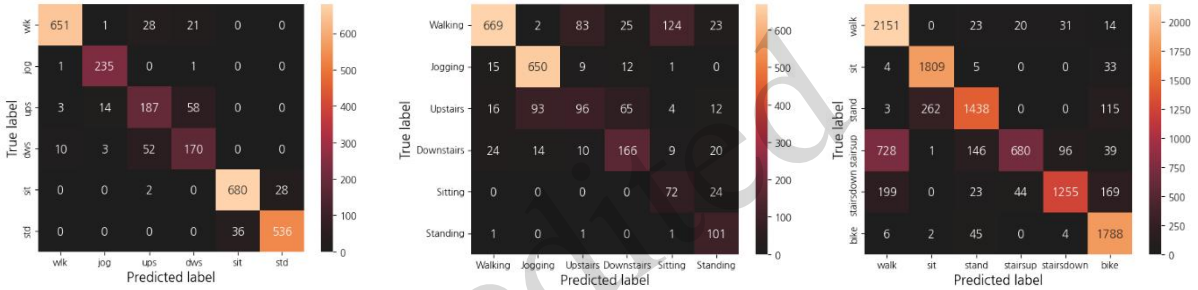
(a) (b) (c)

Fig. 4 Class distribution in the (a) Motionsense; (b) WISDM; (c) HHAR datasets.

Unedited

Table 3 The performance results of FedCoad and other FL algorithms on benchmark datasets.

Method	Motionsense		WISDM		HHAR	
	Accuracy %	F1-score %	Accuracy %	F1-score %	Accuracy %	F1-score %
FedAvg	91.44	88.32	73.94	67.69	80.86	79.07
FedProx	95.24	92.16	75.61	70.33	79.85	77.63
SCAFFOLD	92.48	90.29	75.42	71.20	82.93	82.02
FedAvgM ($\beta = 0.7$)	86.23	83.13	73.48	66.52	77.10	74.53
FedAvgM ($\beta = 0.997$)	81.76	78.00	64.25	55.74	65.92	71.09
MOON (μ=1.0)	91.76	88.32	74.62	66.26	80.36	78.24
MOON (μ=5.0)	91.65	87.90	73.36	64.59	77.58	74.64
MOON (μ=10.0)	91.94	88.37	72.63	63.26	80.24	78.25
FedCoad (μ=1.0)	**95.52**	**92.70**	76.58	**73.55**	83.78	82.81
FedCoad (μ=5.0)	93.47	90.61	**76.67**	73.17	**84.31**	**83.22**
FedCoad (μ=10.0)	94.78 %	91.67 %	76.46 %	71.91 %	84.06 %	82.91 %



(a) (b) (c)

Fig. 5 FedCoad confusion matrix on (a) Motionsense; (b) WISDM; (c) HHAR datasets.

Samsung Galaxy S3 (150 Hz), Samsung Galaxy Wear (100 Hz), and LG G (200 Hz). During the data collection, all subjects were asked to perform each activity for 5 min. This study used only the triaxial accelerometer signals from the smartphone for compatibility with other datasets.

4.2 Experimental setup

The experiment was performed on a computer with Intel Core I9-9900K @3.60 GHz, GPU NVIDIA RTX 2080, and the neural network was built using Python and Pytorch libraries. This study compared FedCoad with the state-of-the-art approaches FedAvg (McMahan et al., 2017), FedAvgM (Hsu et al., 2019), SCAFFOLD (Karimireddy et al., 2020), FedProx (T. Li et al., 2018), and MOON (Q. Li et al., 2021). The model used 1D-CNN as the base encoder, followed by two fully connected layers as the projection head, and one fully connected layer as the output layer.

For the data preprocessing, all data were segmented into a time window with 50% overlap and a sequence length of 200. For example, Motionsense was captured at 50 Hz so sensor readings with a duration of 4 s were taken as one sample window. During the evaluation, 80% of the subjects were used as training clients while the rest were used for server private evaluation data. Each algorithm used Adam as the optimizer with a learning rate of 0.001 with a weight decay of 0.00001. The number of batch sizes was set to 8, the number of communication rounds was set to 100, and the local epochs were set to 5. Also, the methods that used contrastive learning temperature τ were set to 0.5.

4.3 Performance comparison

In this section we report the performance of all the listed algorithms for those clients with fully labeled data. All algorithms were run five times to account for variation. The experiment varied the hyperparameter $\mu = \{1.0, 5.0, 10.0\}$ for both FedCoad and MOON according to the range used in Q. Li et al. (2021). For FedAvgM, the momentum value β was set to 0.7 and 0.997 following the original paper (Hsu et al., 2019). Table 3 shows the accuracy and F1-score results of all the listed algorithms.

Unedited

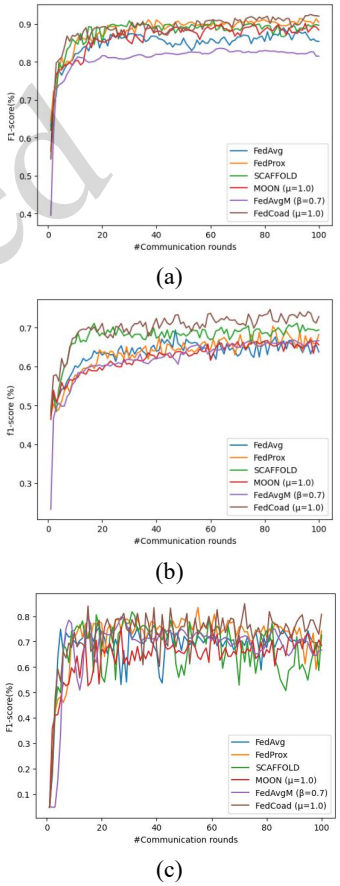
Table 4 The number of rounds for FL methods to achieve a similar F1-score as running FedAvg for 100 rounds. '\' means the method fails to outperform FedAvg convergence speed.

Methods	Motionsense		WISDM		HHAR	
	#rounds	speedup	#rounds	speedup	#rounds	speedup
FedAvg	100	1×	100	1×	100	1×
FedAvgM	\	\	51	1.96×	7	14.28×
FedProx	11	9.1×	19	5.26×	11	9.1×
SCAFFOLD	9	11.11×	**8**	**12.5×**	9	11.11×
MOON (μ=1.0)	15	6.67×	48	2.08×	27	3.7×
MOON (μ=5.0)	73	1.37×	94	1.06×	32	3.12×
MOON (μ=10.0)	40	2.5×	99	1.0×	**4**	**25×**
FedCoad (μ=1.0)	**7**	**14.3×**	**8**	**12.5×**	10	10×

From the accuracy metric, FedCoad outperformed MOON which used model contrastive learning, and SCAFFOLD which used control variates. For example, FedCoad (μ =1.0) outperformed MOON (μ =10.0) by 3.58% in Motionsense and outperformed SCAFFOLD by 1.16% in WISDM. FedCoad (μ=1.0) also slightly outperformed FedProx by 0.28% in Motionsense and 0.97% in WISDM. In the HHAR dataset, FedCoad (μ =5.0) achieved the highest accuracy and outperformed SCAFFOLD by 1.38% and FedProx by 4.46%.

Next, we used the F1-score which accounts for class imbalance in HAR to estimate the FedCoad improvement. From the F1-score results, methods that include only model contrastive learning failed to improve the performance on vanilla FedAvg. For example, MOON (μ=1.0) achieved 66.26% in terms of the F1-score while FedAvg was 67.69% in WISDM which shows model contrastive learning alone is insufficient for HAR. On the other hand, the methods that include regularization in the local training step, such as FedProx, SCAFFOLD, and FedCoad, outperformed FedAvg. Specifically, FedCoad, FedProx, and SCAFFOLD can perform competitively among other methods. For example, the FedCoad (μ=1.0) F1-score outperformed FedAvg by 4.38% and 5.86% in Motionsense and WISDM, respectively.

For hyperparameter μ, FedCoad with $\mu = 1.0$ and $\mu = 5.0$ achieved the highest F1-score and accuracy in all benchmark datasets. In the Motionsense dataset, FedCoad with $\mu = 1.0$ achieved accuracy of 95.52% and an F1-score of 92.70%, which surpassed those of other methods. On the other hand, FedCoad with μ=5.0 outperformed other methods in the HHAR dataset with accuracy of 84.31% and an F1-score of 83.22%. For the WISDM dataset, both $\mu = 1.0$ and $\mu = 5.0$ gave the same performance in terms of accuracy and F1-score. Therefore, $\mu = 1.0$ and $\mu = 5.0$ are suitable values for FedCoad.



(a)

(b)

(c)

Fig. 6 The F1-score in various number of communication rounds for (a) Motionsense; (b) WISDM; (c) HHAR dataset

Fig. 5a-5c shows the correctness of FedCoad class predictions for all benchmark datasets. In Motionsense, FedCoad could correctly predict the

Unedited

true classes with a minimal amount of error. For the other datasets, FedCoad produced some classification errors for activities with similar locomotion. For example, in Fig. 5b and Fig. 5c, FedCoad shows some errors distinguishing between 'upstairs' and 'downstairs' in WISDM and between 'sit' and 'stand' in HHAR. However, FedCoad correctly predicted the true classes for most of the cases in both datasets.

From the results of FedCoad, combining model contrastive learning with control variates improved the performance compared to the other methods. For example, FedCoad (μ =1.0) with F1-scores of 92.70% and 73.55% outperformed SCAFFOLD by 2.41% and 2.35% in Motionsense and WISDM, respectively.

4.4 Convergence speed comparison

In this section, we compare all the methods against FedAvg in terms of their F1-score convergence rates. The number of rounds needed for all methods to achieve the same F1-score as FedAvg for 100 rounds is shown in Table 4. Fig. 6a-6c compares the convergence speed of all methods. FL baselines with regularization can achieve significant speed up compared to FedAvg. For example, in Table 4, FedCoad and SCAFFOLD achieved 12.5× speedup compared to FedAvg in WISDM. At the start of the round, the speed of F1-score improvement in FedCoad was similar to FedAvg and other FL methods. Nevertheless, FedCoad achieved a higher F1-score than FedAvg in later rounds. Furthermore, FedCoad achieved the highest speedup in two out of three datasets (Motionsense and WISDM), demonstrating its superiority.

On the other hand, all methods had unstable convergence in the HHAR dataset (Fig. 6c)**.** Unlike the other datasets, the clients in the HHAR dataset had different devices such as the Samsung Galaxy S3 or LG Nexus 4. Therefore, heterogeneity from the devices affected global model aggregation. Nevertheless, the proposed FedCoad had an upward and stable trend from round 80 to round 100 and achieved the highest F1-score performance in the end compared to the other methods.

In all benchmark datasets, FedCoad and SCAFFOLD performed competitively in terms of convergence rate. For MotionSense, FedCoad was superior to all other methods (14.3x faster than FedAvg). However, in WISDM, SCAFFOLD and FedCoad were both 12.5x faster than FedAVG.

4.5 Handling real-world HAR skewness

In real-world HAR implementation, there are three skewness scenarios (Presotto et al., 2022) to consider. In this study, we simulated the skewness using the benchmark dataset that previously followed laboratory settings to make it follow non-IID in a real-world setting. Figs. 7a-7d show the effect of each skewness scenario on the clients' data distribution. The following steps were taken to simulate the skewness:

(1) Feature distribution skew (FS): the experiment used the skewness already present in each dataset, as each client already had a different pattern.

(2) Label distribution skew (LS): 0 to 2 labels were randomly removed to simulate the variation of available activities of each client (C. Li et al., 2021).

(3) Quantity distribution skew (QS): each client has only small amounts of labeled data in real life (Tang et al., 2021) so in the experiments, each client used only 10% of the available data as a training set.

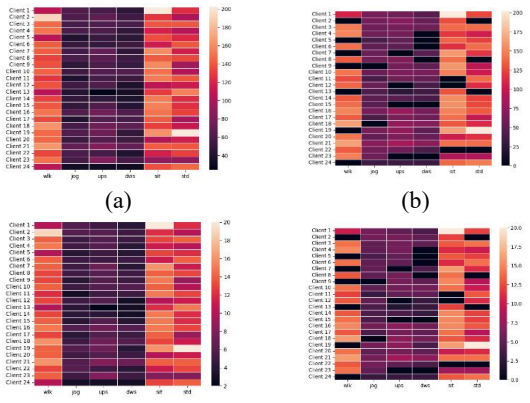


Fig. 7 The data distribution of each party in different skewness scenarios (a) FS; (b) FS+LS; (c) FS+QS; (d) FS+LS+QS using the Motionsense dataset

4.5.1 Feature skew and label skew (FS+LS)

In the FS+LS condition, some of the clients' original training label sets are removed. Therefore, the clients have an uneven label set for training and each client's local objectives can misguide the local

Unedited

Table 5 The F1-score and reduction of FL methods in the FS+LS scenario

Method	Motionsense		WISDM		HHAR	
	F1-score %	Reduction %	F1-score %	Reduction %	F1-score %	Reduction %
FedAvg	86.80	1.52	66.58	1.11	71.32	7.75
FedProx	**90.87**	1.29	**69.44**	0.89	59.64	17.99
SCAFFOLD	88.10	2.19	66.42	4.78	73.28	8.74
FedAvgM (β=0.7)	82.10	**1.03**	68.01	**-1.49**	68.21	6.32
MOON (μ=1.0)	85.27	3.05	65.24	1.02	**73.84**	**4.4**
FedCoad (μ=1.0)	90.58	2.12	63.77	9.78	72.06	10.75

Table 6 The F1-score and reduction of FL methods in the FS+QS scenario

Method	Motionsense		WISDM		HHAR	
	F1-score %	Reduction %	F1-score %	Reduction %	F1-score %	Reduction %
FedAvg	85.07	3.25	64.08	3.61	80.11	-1.04
FedProx	86.73	5.43	66.64	3.69	**87.04**	**-9.41**
SCAFFOLD	**90.12**	0.17	**67.97**	3.23	85.87	-3.85
FedAvgM (β=0.7)	83.20	**-0.07**	62.61	3.91	80.88	-6.35
MOON (μ=1.0)	87,90	0.42	66.53	**-0.27**	78.29	-0.05
FedCoad (μ=1.0)	89.84	2.86	67.62	5.93	85.47	-2.66

Table 7 The F1-score and reduction of FL methods in the FS+LS+QS scenario

Method	Motionsense		WISDM		HHAR	
	F1-score %	Reduction %	F1-score %	Reduction %	F1-score %	Reduction %
FedAvg	76.14	12.18	63.95	3.74	76.91	2.16
FedProx	85.03	7.13	65.35	4.98	**86.96**	**-9.33**
SCAFFOLD	84.69	5.6	67.33	3.87	75.51	6.51
FedAvgM (β=0.7)	77.62	5.51	61.48	5.04	78.92	-4.39
MOON (μ=1.0)	77.57	10.75	63.85	2.41	75.93	2.31
FedCoad (μ=1.0)	**88.54**	4.16	**69.49**	4.06	78.31	4.50

Unedited

model training. Table 5 shows that the performance of all FL methods was reduced significantly in the FS+LS condition. For example, the performance of SCAFFOLD decreased by 5.24% and MOON by 2.83% on average across all datasets. A notable exception was FedProx which decreased by only 1.29% in Motionsense and 0.89% in WISDM. However, FedProx had the worst performance in HHAR, showing a reduction of 17.99%. Even though FedCoad achieved competitive performance along with other methods in Motionsense and WISDM, it showed a severe performance reduction of 7.55% on average across all datasets.

4.5.2 Feature skew and quantity skew (FS+QS)

In the FS+QS condition, all clients retain their original training label set but the quantity decreases to 10% of the original size. So, the risk of overfitting in local model training increases since the amount of training data is reduced. Compared to the FL+LS condition, there were no significant performance reductions among all FL methods. For example, FedCoad performance decreased by 2.04% on average across all datasets (Table 6). For this condition, FedProx had the most significant performance reduction of 5.43%. However, there is an odd phenomenon in HHAR. Instead of decreasing, the performance of all FL methods increased over their performance in the FS condition. Nevertheless, FedCoad was robust enough to train a global model in FS+QS conditions.

4.5.3 All skewness scenarios combined (FS+QS+LS)

In the FS+QS+LS condition, some of the clients' original training label set is removed and the quantity of the remaining training data decreases by 10%. In the FS+QS+LS condition, most FL methods experienced a larger performance reduction than in the FS+QS condition (Table 7). For example, FedCoad performance was reduced by 4.24% on

average across all datasets in FS+QS+LS compared to 2.04% across all datasets in the FS+QS condition. It seems the FS+LS and FS+QS+LS conditions caused large performance reductions using all FL methods. Therefore, LS skewness was the main cause of performance reduction among other skewness scenarios. Based on performance, FedCoad performed best in FS+QS+LS conditions. For example, it achieved the highest F1-scores of 88.54% in Motionsense and 69.49% in WISDM.

5. Discussion

In this section we delve into the experimental results and provide insights into the factors contributing to the observed outcomes. Without the simulated skewness described in Section 4.3, the proposed FedCoad achieved competitive performance compared to methods like MOON and SCAFFOLD, which address heterogeneity either during local training or through local model parameter updates. For instance, in the WISDM dataset, FedCoad (μ=1.0) outperformed MOON (μ=1.0) by 1.96% and SCAFFOLD by 1.16% in accuracy. This suggests that FedCoad and similar baseline methods are sufficient for training a global model in the FS condition (dataset subject-level skewness). However, the F1-score results indicate that FedCoad (μ=1.0) predictions are more balanced across classes, with an F1-score improvement of 7.29% over MOON (μ=1.0).

When introducing simulated skewness (Section 4.5) that reflects real-world conditions, the performance of several FL baseline methods declined significantly. For example, in the Motionsense dataset under FS+LS+QS conditions, methods like MOON, which focuses on heterogeneity during local training, and SCAFFOLD which focuses on parameter updates, experienced more severe degradation than FedCoad. MOON, which uses only model contrastive learning during local training, showed a 10.75% reduction in F1-score compared to its performance without the simulated skewness (Section 4.3). This highlights the insufficiency of handling heterogeneity solely during local training. Similarly, FedProx and SCAFFOLD, which address heterogeneity during parameter updates, showed F1-score decreases of 7.13% and 5.6%, respectively, indicating that regularizing local model weights after training mitigates some of the effects of HAR skewness as it directly reduces the bias from skewed local training objectives.

FedCoad, which addresses heterogeneity during both local training and local model parameter updates, showed superior performance under these conditions. It achieved the highest F1-score of 88.54% with the smallest performance decrease (4.16%) compared to other baseline methods. These results underscore FedCoad's resilience and robustness in handling non-IID environments. Additional experiments further validated its strengths, showing faster convergence rates and the ability to train model encoders that generate distinguishable feature embeddings (see supplementary materials, Section 1.3), which are essential for classification tasks.

Furthermore, the ablation studies described in the supplementary materials reveal the contribution of FedCoad's components (see supplementary materials, Section 2.1) and their synergistic effect on performance. Experiments on client availability (see supplementary materials, Section 2.3), also highlighted FedCoad’s scalability. For example, when comparing scenarios with 20% and 100% client availability, FedAvg's F1-score improved by only 1.41%, whereas FedCoad's improved by 4.44%.

Despite its advantages, FedCoad has limitations for real-world implementation. It currently does not use unlabeled data, which is abundant on client devices and could enhance local training and adaptation. Additionally, FedCoad incurs higher communication costs due to the exchange of global and local control variates during training. Nevertheless, its strong performance in simulated skewness environments suggests its potential for real-world applications. Future work should focus on addressing these limitations, such as incorporating semi-supervised learning to leverage unlabeled data and optimizing communication efficiency to reduce costs.

6. Conclusions

In this work, we studied the problem of HAR in the FL setting and built a generalized global model that can be adapted by participating clients. The

Unedited

skewness among clients in the real world poses significant challenges to implementing FL for HAR. To handle this, we proposed FedCoad, a new approach to align global and local model representation using model contrastive learning to bridge and control variates to regularize the local model update. From our experiments, FedCoad outperformed other methods in tests of skewed dataset settings (non-IID) in the benchmark datasets. Even though FedCoad had disadvantages in FS+LS conditions, it exceled in FS+QS+LS which implies that FedCoad could be more robust in a real-world scenario. Furthermore, the performance of FedCoad was comparable to that of pretraining methods that require additional fine-tuning data (see supplementary materials, Section 1.2). The results also showed that FedCoad can reduce the communication cost by up to 90.3% compared to the state-of-the-art ModcL (supplementary materials, Section 1.4).

Despite the improvement, FedCoad has a limitation in the unlabeled data partition and personalization scheme. It lacks the use of unlabeled data on the client side that is beneficial for local training and adaptation steps to optimize the performance in client environments. Extreme heterogeneity in hardware capabilities or data imbalance among clients may also impact FedCoad's stability and performance. Moreover, FedCoad's additional memory requirements for control variates may pose challenges to low-powered edge devices with minimal storage.

In the future work, this work can be extended to use unlabeled data with semi-supervised learning on the client's side for leveraging the unlabeled data to optimize performance and explore personalization strategies that could further tailor the model to individual client environments. Developing an adaptive weighting mechanism could also help in the server aggregation step to combine the local model updates from clients with extreme data imbalance. Additionally, optimizing or compressing control variates could reduce memory demands, making FedCoad more suitable for edge devices.

Contributors

Ignatius IWAN designed the research and drafted the paper. All the authors revised and improved the paper.

Data Availability

Datasets in this experiment can be acquired through the following link : WISDM (https://archive.ics.uci.edu/dataset/507/wisdm+smartphone+and+smartwatch+activity+and+biometrics+dataset); Motionsense (https://github.com/mmalekzadeh/motion-sense) and HHAR (https://archive.ics.uci.edu/dataset/344/heterogeneity+activity+recognition)

Conflict of Interest

All the authors declare that they have no conflict of interest.

Unedited

References

Bianchi V, Bassoli M, Lombardo G, et al., 2019. IoT wearable sensor and deep learning: an integrated approach for personalized human activity recognition in a smart home environment. *IEEE Internet Things J*, 6(5):8553-8562. https://doi.org/10.1109/JIOT.2019.2920283

Diao EM, Ding J, Tarokh V, 2020. HeteroFL: computation and communication efficient federated learning for heterogeneous clients. Proc 9th Int Conf on Learning Representations.

Guo CR, Zhang YW, Chen YQ, et al., 2024. Modality consistency-guided contrastive learning for wearable-based human activity recognition. *IEEE Internet Things J*, 11(12):21750-21762. https://doi.org/10.1109/JIOT.2024.3379019

Guo JL, Li ZT, Liu AF, et al., 2024. REC-fed: a robust and efficient clustered federated system for dynamic edge networks. *IEEE Trans Mob Comput*, 23(12):15256-15273. https://doi.org/10.1109/TMC.2024.3452312

Hsu TMH, Qi H, Brown M, 2019. Measuring the effects of non-identical data distribution for federated visual classification. https://arxiv.org/abs/1909.06335.

Karimireddy SP, Kale S, Mohri M, et al., 2020. SCAFFOLD: stochastic controlled averaging for federated learning. Proc 37th Int Conf on Machine Learning, article 476.

Kwapisz JR, Weiss GM, Moore SA, 2011. Activity recognition using cell phone accelerometers. *ACM SIGKDD Explor Newsl*, 12(2):74-82. https://doi.org/10.1145/1964897.1964918

Li CL, Niu D, Jiang B, et al., 2021. Meta-HAR: federated representation learning for human activity recognition. Proc Web Conf, p.912-922. https://doi.org/10.1145/3442381.3450006

Li QB, He BS, Song D, 2021. Model-contrastive federated learning. Proc IEEE/CVF Conf on Computer Vision

and Pattern Recognition, p.10713-10722. https://doi.org/10.1109/CVPR46437.2021.01057

Li T, Sahu AK, Zaheer M, et al., 2020. Federated optimization in heterogeneous networks. Proc 3rd Conf on Machine Learning and Systems.

Li X, Huang KX, Yang WH, et al., 2020. On the convergence of fedavg on non-IID data. Proc 8th Int Conf on Learning Representations.

Malekzadeh M, Clegg RG, Cavallaro A, et al., 2019. Mobile sensor data anonymization. Proc Int Conf on Internet of Things Design and Implementation, p.49-58. https://doi.org/10.1145/3302505.3310068

McMahan HB, Moore E, Daniel R, et al., 2017. Communication-efficient learning of deep networks from decentralized data. Proc 20th Int Conf on Artificial Intelligence and Statistics, p.1273-1282.

Mrozek D, Koczur A, Małysiak-Mrozek B, 2020. Fall detection in older adults with mobile IoT devices and machine learning in the cloud and on the edge. *Inf Sci*, 537:132-147. https://doi.org/10.1016/j.ins.2020.05.070

Presotto R, Civitarese G, Bettini C, 2022. FedCLAR: federated clustering for personalized sensor-based human activity recognition. Proc IEEE Int Conf on Pervasive Computing and Communications, p.227-236. https://doi.org/10.1109/PerCom53586.2022.9762352

Sozinov K, Vlassov V, Girdzijauskas S, 2018. Human activity recognition using federated learning. Proc IEEE Int Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications, p.1103-1111. https://doi.org/10.1109/BDCloud.2018.00164

Stisen A, Blunck H, Bhattacharya S, et al., 2015. Smart devices are different: assessing and MitigatingMobile sensing heterogeneities for activity recognition. Proc 13th ACM Conf on Embedded Networked Sensor Systems, p.127-140. https://doi.org/10.1145/2809695.2809718

Tang CI, Perez-Pozuelo I, Spathis D, et al., 2021. SelfHAR: improving human activity recognition through self-training with unlabeled data. *Proc ACM Interact Mob Wearable Ubiquitous Technol*, 5(1):36. https://doi.org/10.1145/3448112

Wang HY, Yurochkin M, Sun YK, et al., 2020. Federated learning with matched averaging. Proc 8th Int Conf on Learning Representations.

Wu JH, Sun J, Song J, et al., 2022. Health assessment method based on multi-sign information fusion of body area network. *Inf Sci*, 618:136-149. https://doi.org/10.1016/j.ins.2022.10.033

List of supplementary materials

1 Extension of experiment section

2 Ablation studies

Table S1 Performance comparison with pretraining methods

Table S2 Communication exchange cost for one round in Motionsense

Table S3 Ablation studies of FedCoad components

Fig S1 T-SNE of (a) centralized and (b) Fedcoad in Motionsense dataset

Fig S2 T-SNE of (a) centralized and (b) Fedcoad in WISDM dataset

Fig S3 T-SNE of (a) centralized and (b) Fedcoad in HHAR dataset

Fig. S4 Effect of temperature (τ)

Fig. S5 Effect of client availability on FedAvg and FedCoad performance

Unedited