Collaborative Learning of Different Types of Healthcare Data from Heterogeneous IoT Devices

Lei Zhao, *Student Member, IEEE,* Lin Cai*, *Fellow, IEEE,* Wu-Sheng Lu, *Life Fellow, IEEE*

***Abstract*—In the realm of healthcare data analysis, privacy concerns have been tackled by the Federated Learning (FL) framework. However, in the situation that heterogeneous healthcare Internet of Things (IoT) devices collect different types of data, applying FL becomes difficult. To train a model leveraging diverse healthcare IoT devices, we propose an advanced collaborative learning framework to fill the gap. With the proposed collaborative learning framework, individual IoT devices project their sensed features into a carefully developed latent space, which are transmitted to a central server. For privacy preservation, the latent local features are encoded within this space, while the samples' labels remain securely stored in the individual IoT devices. Collaboratively, the deep neural network model is trained by both the central server and the diverse IoT devices. The central server handles the computationally intensive training processes, while the individual IoT devices evaluate the model's performance and initiate back-propagation based on their locally stored labels. Experimental results demonstrate that the proposed collaborative learning framework achieves performance similar to centralized training and significantly outperforms individual training while preserving data privacy.**

***Index Terms*—Collaborative Learning, Heterogeneous Healthcare Informatics, Latent Features.**

I. Introduction

The rapid proliferation of Internet of Things (IoT) devices in healthcare has resulted in the collection of vast volumes of data, fueling a wide range of medical applications such as remote health monitoring, fitness programs, disease detection, and elderly care [1] [2]. However, individual healthcare IoT devices often face limitations in terms of data capacity and computing resources [3], making it arduous to train deep learning models to yield meaningful insights. Consequently, many smart healthcare applications have turned to the cloud-based approach for machine learning model training [4]. Nevertheless, cloud solutions raise valid concerns regarding data privacy and escalating maintenance costs [5]. The potential compromise of sensitive private healthcare information remains a pressing issue [6] [7]. Furthermore, the investments required to maintain a cloud data center and the challenges associated with obtaining permissions for storing and processing healthcare data add further complexity to the situation.

Federated Learning (FL) has emerged as an attractive approach for healthcare data thanks to its ability to facilitate collaboration among data holders while keeping sensitive information locally [8] [9]. By employing the FL framework, individual devices can perform local processing, while the central server handles coordination and aggregation [10]. Extensive research has focused on developing advanced FL algorithms to enhance learning performance, with emphasis on privacy preservation [11] [12] and learning efficiency [13].

However, the existing FL approaches predominantly revolve around scenarios where the same type of local data is used to train a unified global model. In healthcare, the integration of heterogeneous data from diverse devices, such as medical imaging equipment, wearable sensors, electronic health records, genomics data, and precision medicine interventions, is essential for training comprehensive models that enable accurate diagnosis, personalized treatment, and improved healthcare outcomes [14]. Consequently, the conventional FL framework encounters challenges when attempting to facilitate collaborative learning with heterogeneous healthcare IoT device data. Furthermore, the computational demands imposed on individual IoT devices during the training process may prove impractical, particularly given the limited capacity of wearable healthcare IoT devices. Therefore, minimizing the computational loads on healthcare IoT devices becomes imperative to ensure the feasibility of collaborative learning scenarios.

In this work, we present a collaborative learning framework that harnesses the capabilities of heterogeneous healthcare IoT devices effectively. Our approach focuses on developing unified latent features for diverse local data from various IoT devices. Ensuring privacy is of utmost importance, and thus, we keep the generation of local latent features on the respective devices, preventing the reverse engineering of the original data. To further enhance privacy, we preserve the sample labels locally within the proposed collaborative learning framework, safeguarding sensitive information. At the central server, a deep neural network model is trained collaboratively with the individual IoT devices, leveraging the locally preserved labels and transformed representations of the local samples.

The significance of our proposed collaborative learning framework lies in its ability to effectively collect diverse healthcare IoT devices with heterogeneous local data sets. This capability is crucial, given the challenges of conducting collaborative training with varied local samples. Moreover, the framework offers a compelling advantage by substantially reducing the local computation burden, which is vital for resource-limited healthcare IoT devices. By generating lower-dimensional local latent features from the original data, our framework also alleviates the communication burden, promoting seamless collaboration among different healthcare IoT devices. These extracted essential local information not only facilitates collaboration but also enhances training efficiency and scalability. We leverage latent features to share valuable information while concealing sensitive details from the original data. The local generation of latent features represents data

L. Zhao, L. Cai, and W-S. Lu are with Dept. of Electrical & Computer Engineering, University of Victoria, 3800 Finnerty Road, Victoria, BC, V8P 5C2, Canada. *Corresponding author: Lin Cai (E-mail: cai@ece.uvic.ca).

in a more abstract and concise manner, effectively filtering out noise and irrelevant information, resulting in a more robust and accurate shared model. The shared model, trained on these latent features, demonstrates heightened resilience to small changes in the input data, offering significant benefits for healthcare applications dealing with noisy or incomplete data.

To comprehensively evaluate the performance of our collaborative learning framework, we conducted extensive simulations. The numerical results demonstrate that our proposed approach achieves performance comparable to centralized training methods while significantly outperforming individual training. Through our research, we aim to revolutionize the healthcare landscape by harnessing the collective potential of heterogeneous IoT devices. By embracing a unified latent feature space and prioritizing data privacy, our approach opens new avenues for intelligent healthcare applications that effectively utilize the rich and diverse data sources provided by healthcare IoT devices.

The rest of this paper is organized as follows. In Section II, we provide an overview of related works in the field. Section III presents the system model and formulates the collaborative training problem. In Section IV, we elaborate on our proposed collaborative learning framework with an adaptive local feature space. Theoretical analysis is provided in Section V to support the effectiveness of our approach. To demonstrate the training efficiency under various settings, we present the simulation results in Section VI, followed by the concluding remarks in Section VII.

II. Related Works

IoT devices have become prevalent tools for collecting data in intelligent healthcare applications [15]. However, due to privacy concerns surrounding healthcare data, uploading such data to cloud data centers for model training is undesirable [16] [17]. Preserving data locally is particularly favored, especially in healthcare applications where privacy is of utmost importance. However, resource-limited IoT devices face significant challenges in processing data locally. The constrained computing power and memory capacity of these devices can hinder their ability to handle large volumes of data efficiently. In additional, the diversity in local data samples presents substantial challenges for effective collaboration. Integrating data from various sources, each with its unique characteristics, can complicate the collaboration process and hinder the seamless sharing of information among IoT devices. Overcoming these challenges is essential to harnessing the full potential of IoT devices in data processing and collaboration.

Numerous research works have embraced FL framework to train global models by utilizing decentralized data from multiple clients [3]. Within the healthcare domain, diverse approaches have been employed to address various challenges. Some works have specifically focused on using FL to ensure privacy and security in collaborations among medical institutions. For example, references [18] and [19] adopted FL to address privacy concerns by avoiding the sharing of raw data or model details [20] [21]. These efforts primarily focus on extracting knowledge from electronic health records. One notable approach is seen in [22], where tensor factorization models were employed to convert vast electronic health records into meaningful phenotypes for data analysis. Additionally, a two-stage federated natural language processing method [23] facilitates the utilization of clinical notes from different hospitals or clinics. Moreover, [24] introduced a community-based FL algorithm that accommodates the decentralized non-IID (Independent and Identically Distributed) and privacy-sensitive characteristics of electronic medical records. This work clusters distributed data into clinically meaningful communities to learn one model capturing similar diagnoses and geographical locations. While these methods have demonstrated efficacy in natural language processing tasks and with specific healthcare data from electronic medical records, they may not be directly suitable for healthcare IoT devices due to their distinct characteristics and limitations.

Other works focus on specific healthcare data, such as [25], which investigated brain structural relationships across diseases and clinical cohorts using FL. A general decentralized optimization framework [26] was developed to collaboratively train sparse support vector machines to perform binary classification on the diagnosis of heart failure among multiple data holders. How to decompose the approximated neural network function to enable collaboration among IoT devices over the first shallow components was proposed in [27] [28]. However, the diversity of the training data has not been fully explored in the above works, as their training data typically follow the same structure.

Conventional FL framework faces significant challenges in handling heterogeneous local data sets [29] [9]. All FL clients are required to share the same global model [30] and perform multiple local model updates before communicating with the central server [31]. However, this approach is not suitable for serving multiple types of IoT devices with varying local sample dimensions, which is crucial in healthcare IoT applications. Each healthcare IoT device serves unique functions, and collaboration among different types of devices is essential to generate useful services. Hence, there is a pressing need to develop an innovative framework that enables seamless collaboration among diverse healthcare IoT devices.

Moreover, traditional FL framework often assume that individual devices are capable of conducting local updates of deep neural network models, which may not be feasible in healthcare IoT applications [32] [33] [10]. The resource capacity of individual IoT devices is often limited and varies from device to device, so multiple local updates can lead to severe delay variance, causing convergence difficulties with non-IID local data distributions with heterogeneous devices [24] [34] [35]. Existing improvements to the FL framework, such as FedProx [9], introduced a proximal term to local objective functions, and other works employed variance reduction to correct client drift in local updates [33] to address challenges from the biased estimate of the global gradient. However, these approaches cannot well address the complexities of collaborating heterogeneous healthcare IoT devices. To overcome these challenges and foster effective collaboration among diverse healthcare IoT devices, a novel framework that accounts for feature space diversity, varying device capacities, and non-

IID data distributions is required. By addressing these critical issues, we can unlock the potential of collaborative learning in healthcare IoT applications.

In this paper, we develop a novel collaborative learning framework designed specifically for addressing the challenges posed by the diverse local feature spaces of heterogeneous healthcare IoT devices. By leveraging this framework, we substantially reduce the work loads for individual IoT devices during the collaborative training procedure. Our proposed collaborative learning framework effectively tackles the issue of diverse local feature spaces, which makes our approach promising for a wide range of practical healthcare applications, where the efficient utilization of heterogeneous IoT devices is important. Simulation results show that the proposed collaborative learning process is efficient and feasible.

III. System Model and Problem Formulation

A. System Model

In this section, we propose a collaborative learning framework as shown in Fig. 1 specifically designed to address the challenges posed by the diversified local feature space in healthcare IoT intelligence. With the widespread use of IoT devices, including wearable devices in smart healthcare, a massive volume of data is collected. However, individual healthcare IoT devices have inherent limitations, such as constrained processing, storage, and power capacities. Therefore, it is crucial to simplify the training tasks on each individual device.

To tackle these challenges, our framework emphasizes collaboration among healthcare IoT devices to achieve robust performance in supporting AI applications. This collaborative approach enables the creation of a unified latent feature space that captures the combined knowledge from multiple devices, resulting in improved learning capabilities and enhanced performance. We aim to unlock the full potential of healthcare IoT intelligence while effectively addressing the constraints imposed by the diversified local feature space. Through collaboration, we pave the way for harnessing the rich and diverse data sources provided by IoT devices.

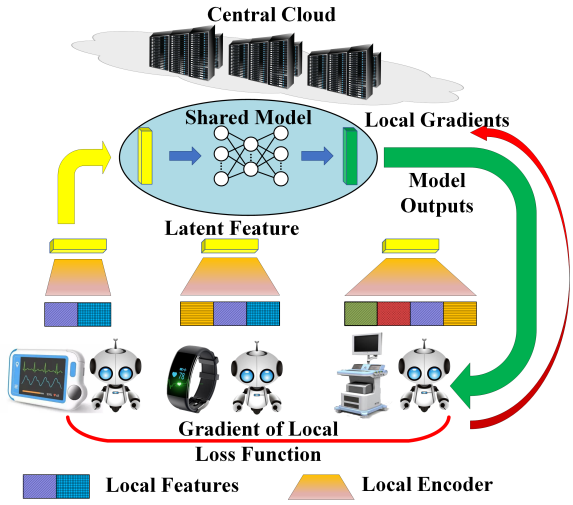


Fig. 1. Collaborative learning framework for healthcare IoT intelligence.

In the proposed framework, each healthcare IoT device performs a few training steps on its local feature adaptive encoder parameters and then transmits the encoded latent features to a central server. The central server aggregates the locally encoded latent features in each round and performs a forward computation using the shared deep neural network parameters. The resulting outputs of the shared neural network model are then distributed to the individual IoT devices. At this stage, each IoT device computes the local gradients of the cross-entropy loss based on the received outputs from the neural network model and their local labels.

By employing this training framework, the deep neural network model is collaboratively trained with the central server and the various healthcare IoT devices. The intensive training computations are handled by the central server, while the individual IoT devices evaluate the model and initiate back-propagation based on their respective local labels. Additionally, the framework ensures enhanced privacy of local data by utilizing encoded latent local features, while the samples' labels are securely retained by the individual IoT devices.

B. Problem Setup

There are E healthcare IoT devices denoted as $\{s_i, i = 1, 2, \cdots, E\}$. The healthcare IoT devices maintain local data sets to train their local encoder model parameters. The shared deep neural network model parameters are denoted by $\boldsymbol{w}$ which is tuned in the central server. The IoT device s_i maintains the local model parameter $\boldsymbol{W}_i$ which constructs a latent space for the local sample features.

The data sample collected by healthcare IoT device s_i is denoted by $\boldsymbol{x}_p \in R^{d_i \times 1}$ and the local data set is $\boldsymbol{D}_i \in R^{d_i \times n_i}$, where there are n_i local training samples and the overall sample number $n = \sum_{i=1}^{E} n_i$. There is no overlap among different local data sets. All data samples in the local data set $\boldsymbol{D}_i$ of healthcare IoT device s_i construct the local objective function $f_i(\boldsymbol{w}, \boldsymbol{W}_i, \boldsymbol{x}_p)$. The optimization problem in a collaborative objective is formulated as

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad f(\boldsymbol{w}) = \sum_{i=1}^{E} \frac{n_i}{n} \sum_{\boldsymbol{x}_p \in \boldsymbol{D}_i} f_i(\boldsymbol{w}, \boldsymbol{W}_i, \boldsymbol{x}_p). \quad (1)$$

However, for the local training procedure in healthcare IoT device s_i, it tries to minimize its own objective function which leads to a local optimal solution. Due to the heterogeneity of the local training data sets, local objective functions are different from each other. The local training procedures provide different directions to update the model which causes difficulty to converge from the global view.

More importantly, the local feature space is different, namely $\{d_i\}_{i=1}^{E}$ are different leading to varying input dimension of the local models. Therefore, minimizing the local objective function and average the results as shown in traditional FL framework is not feasible in this case with different types of local healthcare data sets.

Furthermore, the target to minimize the local objective function may not be practical by the healthcare IoT device s_i itself if the model $\boldsymbol{w}$ becomes too complex. Collaborative learning framework is proposed in following to address the

TABLE I
DESCRIPTION OF NOTATION

NOTATION	DESCRIPTION
E	The total number of IoT devices
$\boldsymbol{D}_i$	The local data set of IoT device s_i
$\boldsymbol{x}_p$	The original feature sample vector with index p
n	The overall sample number
n_i	The sample number possessed by IoT device s_i
$\boldsymbol{w}$	The overall parameters in the shared NN model
$f(\boldsymbol{w})$	The collaborative objective function
$f_i(\boldsymbol{w}, \boldsymbol{W}_i, \boldsymbol{x}_p)$	The local objective function of IoT device s_i
q	The unified latent feature dimension
$\boldsymbol{W}_i$	The local model parameters tuned by IoT device s_i
$\boldsymbol{V}_i$	The latent local data set of samples of IoT device s_i
$\boldsymbol{v}_p$	The latent feature sample vector with index p
$\Phi(\cdot)$	Activation function in neurons
$\Psi(\cdot)$	Normalization function among hidden layers of the shared NN model
d_i	The original feature space with IoT device s_i
$\hat{\boldsymbol{D}}_{\boldsymbol{i}}$	The reconstructed local data set of IoT device s_i
$\boldsymbol{X}_i$	The covariance matrix w.r.t. the local data set $\boldsymbol{D}_i$
λ_j	The j-th eigenvalue of $\boldsymbol{X}_i$
m	The number of hidden layers in the shared NN model
p_l	The number of neurons in the l-th hidden layer of the shared NN model
$\tilde{\boldsymbol{W}}_l$	The weights between the l-th hidden layer and the $(l+1)$-th hidden layer of the shared NN model
B	The latent sample batch size
$\tilde{\boldsymbol{V}}$	One batch of latent samples
$\boldsymbol{A}_l$	The pre-activation values w.r.t. the l-th hidden layer of shared model over $\tilde{\boldsymbol{V}}$
$\boldsymbol{\mu}_l$, $\boldsymbol{\sigma}_l$	The mean and standard deviation w.r.t. columns of $\boldsymbol{A}_l$
Ψ_l	The normalization function among hidden layers
$\boldsymbol{\gamma}_l$, $\boldsymbol{\zeta}_l$	Two parameter vectors in Ψ_l
$\boldsymbol{o}_p$	The output vector of the forward computation w.r.t. $\boldsymbol{v}_p$
α_k	Learning rate for shared NN model
$\varphi(\cdot)$	Quadratic auxiliary function
$\boldsymbol{g}(\boldsymbol{w}_k)$	The gradient information w.r.t. the parameters in shared NN model in the k-th round

limited capacities of individual healthcare IoT devices and the diversified local sample features.

IV. COLLABORATIVE LEARNING WITH ADAPTIVE LOCAL FEATURE SPACE

A. Adaptive Latent Local Feature Design

The local adaptive latent feature design plays a crucial role in enabling the collaboration of heterogeneous healthcare IoT devices. The local adaptive latent feature design aims to capture the unique characteristics of each individual IoT device's data. It involves the development of a customized encoder architecture that transforms the local features into a latent space representation. By leveraging this adaptive latent local feature design, our framework enables collaborative training with the central server and various IoT devices. Moreover, the local adaptive latent feature design also contributes to privacy preservation. By operating in the latent space, sensitive information is inherently abstracted and protected. This ensures that the privacy of healthcare data is maintained throughout the collaboration and training process.

The unified latent feature dimension is denoted by q. The local model $\boldsymbol{W}_i$ of IoT s_i extracts the local data sample features from d_i dimension to q dimension which is defined as $\boldsymbol{W}_i = \begin{bmatrix}\boldsymbol{c}_1 & \cdots & \boldsymbol{c}_q\end{bmatrix}_{d_i \times q}$. The local model parameter matrix $\boldsymbol{W}_i$ can span a q-dimensional latent space. The projection of the n_i local data samples on the q-dimensional space spanned by the columns in $\boldsymbol{W}_i$ are represented by $\boldsymbol{V}_i = \begin{bmatrix}\boldsymbol{v}_1 & \cdots & \boldsymbol{v}_{n_i}\end{bmatrix}_{q \times n_i}$, which are the latent features. Each $\boldsymbol{v}_i$ is the linear combination of the q picked out basis vectors in $\boldsymbol{W}_i$, and the scalar for each basis vectors is the associated component in $\boldsymbol{v}_i$, i.e., $\boldsymbol{v}_i = \sum_{j=1}^{q} v_i^j \boldsymbol{c}_j$.

We apply a neural architecture to tune the local model parameters needed for adaptive latent feature learning. For the local data set $\boldsymbol{D}_i$ as the input to the neural network (NN), the post-activation of the hidden layer is the q-dimensional representations of the n_i data samples, i.e., the columns of $\boldsymbol{V}_i$. For simplicity representation, we consider an example with one hidden layer and linear activation functions, to generate the unified latent feature in q-dimensional space, the hidden layer should contain q neurons.

The local model parameter matrix $\boldsymbol{W}_i$ contains all the weights connecting the units in the input layer and the units in the hidden layer which transforms the data sample in IoT s_i from the d_i-dimensional space to the q-dimensional latent space which is represented by the post-activation value of the hidden layer as $\boldsymbol{V}_i = \Phi(\boldsymbol{W}_i^T \boldsymbol{D}_i)$. Then, the weights of the links connecting the hidden layer units and the output layer is denoted by $\hat{\boldsymbol{W}}_i \in R^{q \times d_i}$.

To ensure that there is no information loss when the original local features are transformed into the latent features, the latent representation $\boldsymbol{V}_i$ must be able to be transformed back into the original d_i-dimensional space. The reconstructed local data set of IoT device s_i is formulated as $\hat{\boldsymbol{D}}_i = \Phi(\hat{\boldsymbol{W}}^T \boldsymbol{V}_i)$, which should be pushed towards the original local data set $\boldsymbol{D}_i$, i.e., IoT device s_i targets to minimize the Frobenius norm of the residual matrix $\| \boldsymbol{D}_i - \hat{\boldsymbol{D}}_i \|_F^2$. With identity activation function assumption, we design the local parameter matrix $\boldsymbol{W}_i$ as one orthogonal matrix to be able to expand the latent space without information loss with its column vectors where $\boldsymbol{W}_i^{-1} = \boldsymbol{W}_i^T$. The latent feature transformation is denoted by

$$\boldsymbol{W}_i^{-1} \boldsymbol{D}_i = \boldsymbol{W}_i^{-1} \boldsymbol{W}_i \boldsymbol{V}_i = \boldsymbol{V}_i, \tag{2}$$

and then transform $\boldsymbol{V}_i$ back to the d-dimensional space is denoted by

$$\boldsymbol{D}_i = \boldsymbol{W}_i \boldsymbol{V}_i = \boldsymbol{W}_i \boldsymbol{W}_i^{-1} \boldsymbol{D}_i. \tag{3}$$

Following the idea in (2) and (3), we design $\hat{\boldsymbol{W}}_i = \boldsymbol{W}_i^T$ and $\hat{\boldsymbol{D}}_i$ is connected to the original local data set $\boldsymbol{D}_i$ by

$$\hat{\boldsymbol{D}}_{\boldsymbol{i}} = \hat{\boldsymbol{W}}_i^T \boldsymbol{V}_i = \boldsymbol{W}_i (\boldsymbol{W}_i^T \boldsymbol{W}_i)^{-1} \boldsymbol{W}_i^T \boldsymbol{D}_i. \tag{4}$$

Then, the objective function for the local training task of IoT device s_i to generate the unified latent features is defined as

$$\begin{aligned} \underset{\boldsymbol{W}_i}{\text{minimize}} \quad & \frac{1}{n_i} \sum_{\boldsymbol{x}_p \in \boldsymbol{D}_i} ||\boldsymbol{W}_i \boldsymbol{W}_i^T \boldsymbol{x}_p - \boldsymbol{x}_p||_2^2 \\ \text{subject to} \quad & \boldsymbol{W}_i^T \boldsymbol{W}_i = \boldsymbol{I}. \end{aligned} \tag{5}$$

Since

$$||\boldsymbol{W}_i \boldsymbol{W}_i^T \boldsymbol{x}_p - \boldsymbol{x}_p||_2^2 = -\boldsymbol{x}_p^T \boldsymbol{W}_i \boldsymbol{W}_i^T \boldsymbol{x}_p + \boldsymbol{x}_p^T \boldsymbol{x}_p, \tag{6}$$

then, the local learning objective is given by

$$-\frac{1}{n_i} \sum_{\boldsymbol{x}_p \in \boldsymbol{D}_i} \boldsymbol{x}_p^T \boldsymbol{W}_i \boldsymbol{W}_i^T \boldsymbol{x}_p + \frac{1}{n_i} \sum_{\boldsymbol{x}_p \in \boldsymbol{D}_i} ||\boldsymbol{x}_p||_2^2. \tag{7}$$

The second term is constant w.r.t. $\boldsymbol{W}_i$, we can focus on the first term to minimize the objective function, i.e.,

$$\boldsymbol{x}_p^T \boldsymbol{W}_i \boldsymbol{W}_i^T \boldsymbol{x}_p = \text{trace}(\boldsymbol{W}_i \boldsymbol{W}_i^T \boldsymbol{x}_p \boldsymbol{x}_p^T). \tag{8}$$

The original objective can be transformed into

$$\begin{aligned} \underset{\boldsymbol{W}_i}{\text{minimize}} \quad & -\text{trace}\left(\boldsymbol{W}_i \boldsymbol{W}_i^T \frac{1}{n_i} \sum_{\boldsymbol{x}_p \in \boldsymbol{D}_i} \boldsymbol{x}_p \boldsymbol{x}_p^T\right), \\ \text{subject to} \quad & \boldsymbol{W}_i^T \boldsymbol{W}_i = \boldsymbol{I}. \end{aligned} \tag{9}$$

The eigen-decomposition of the covariance matrix w.r.t. the local data set $\boldsymbol{D}_i$ is defined as

$$\boldsymbol{X}_i = \frac{1}{n_i} \sum_{\boldsymbol{x}_p \in \boldsymbol{D}_i} \boldsymbol{x_p} \boldsymbol{x_p^T} = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U^T}, \tag{10}$$

which leads to

$$-\text{trace}\left(\boldsymbol{W}_i \boldsymbol{W}_i^T \boldsymbol{X}_i\right) = -\text{trace}\left(\boldsymbol{W}_i^T \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^T \boldsymbol{W}_i\right). \tag{11}$$

We choose q eigenvectors from $\boldsymbol{U}$ as the q columns in the local model parameter matrix $\boldsymbol{W}_i$, then,

$$-\text{trace}\left(\boldsymbol{W}_i^T \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^T \boldsymbol{W}_i\right) = -\sum_{j=1}^{q} \lambda_j, \tag{12}$$

where $\{\lambda_j\}_{j=1}^q$ are the q corresponding eigenvalues. The objective becomes to

$$\begin{aligned} \underset{\boldsymbol{\lambda}}{\text{minimize}} \quad & -\sum_{j=1}^{q} \lambda_j \\ \text{subject to} \quad & \boldsymbol{X}_i = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U^T}. \end{aligned} \tag{13}$$

The proposed collaborative learning framework harnesses the capabilities of heterogeneous healthcare IoT devices by utilizing local models $\{\boldsymbol{W}_i\}_{i=1}^E$ as code books which is kept locally.

Privacy Preservation: By adopting the local adaptive latent feature approach, this framework ensures data privacy as the local models $\{\boldsymbol{W}_i\}_{i=1}^E$ are not shared, thereby preventing the reconstruction of the original features from the latent features in the absence of local model information. Healthcare IoT devices can leverage their unique data characteristics and mitigate privacy risks by reducing the direct exposure of sensitive data. This process effectively obfuscates and conceals sensitive details in the local data.

Latent features represent a more abstract and condensed form of the original data, capturing essential patterns and relationships without divulging specific details about individual data samples. This abstraction adds an extra layer of privacy protection and allows devices to collaborate securely without compromising sensitive information. By embracing this local adaptive latent feature approach, the framework facilitates efficient collaboration among healthcare IoT devices, fostering a collaborative ecosystem where devices can collectively improve the quality of trained models. This advancement lays the groundwork for the development of advanced and intelligent healthcare applications, benefiting patients and practitioners alike.

While achieving absolute robustness against malicious inputs is challenging, our goal is to make it significantly more difficult and costly for attackers to compromise the shared model. The proposed collaboration framework greatly enhances the model's resistance to adversarial attacks during training. In our proposed framework, each IoT device constructs its own latent feature space, designed to focus on the most critical information from the original samples. The use of latent features aids in reducing the impact of adversarial perturbations and improves the model's generalization, making it harder for attackers to identify vulnerabilities in the collaborative process. Moreover, when transmitting obfuscated latent features to train the shared model, the original information is concealed from potential attackers, further increasing the difficulty for malicious users to craft effective adversarial examples. By combining these strategies, our collaborative framework establishes a stronger defense against adversarial attacks and provides a more secure and robust environment for collaborative machine learning among IoT devices.

B. Global Shared Neural Network Function

For generality, we assume the global shared neural network function is formulated by one m layer neural network. The latent features of the data samples are transmitted to multiple neurons by linearly combining the link weights that connect each input node and the neurons to obtain their pre-activation value and compute the post-activation value by the activation function in each neuron. Then the successive neuron layers feed the post-activation value into one another until the output layer. We define each layer contains $p_1, p_2, \cdots, p_m$ neurons. The post-activation outputs of hidden layers are denoted by m vectors $\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_m$ with dimension $p_1, p_2, \cdots, p_m$, respectively. The weights between the l-th hidden layer and the $(l+1)$-th hidden layer are denoted by a connection matrix $\tilde{\boldsymbol{W}}_l \in R^{p_l \times p_{l+1}}$.

To perform the normalization and scaling for the inputs to each hidden layer in the shared NN model, we apply the batch normalization on the pre-activation values for each hidden layer. We define the batch size as B and one latent sample batch is denoted by $\tilde{\boldsymbol{V}} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_B]$. The pre-activation values of the given batch $\tilde{\boldsymbol{V}}$ in the first hidden layer are denoted by

$$\boldsymbol{A}_0 = \tilde{\boldsymbol{W}}_0^T \tilde{\boldsymbol{V}} = [\boldsymbol{a}_0^1, \boldsymbol{a}_0^2, \cdots, \boldsymbol{a}_0^B]. \tag{14}$$

The mean of the pre-activation values over the given batch is denoted by $\boldsymbol{\mu}_0$ and the corresponding standard deviation is denoted by $\boldsymbol{\sigma}_0$ for the first hidden layer are given by

$$\boldsymbol{\mu}_0 = \frac{1}{B} \sum_{i=1}^{B} \boldsymbol{a}_0^i, \qquad \boldsymbol{\sigma}_0^2 = \frac{1}{B} \sum_{i=1}^{B} (\boldsymbol{a}_0^i - \boldsymbol{\mu}_0)^2 + \epsilon, \tag{15}$$

where the operation is piece-wise and ϵ is a small value to avoid the zero variance. After computing the first and second order momentum of the pre-activation values in the first hidden layer, the normalization procedure is defined as

$$\Psi_0(\tilde{\boldsymbol{W}}_0^T \boldsymbol{v}_p) = \frac{\boldsymbol{\gamma}_0}{\boldsymbol{\sigma}_0} \cdot (\boldsymbol{a}_0^p - \boldsymbol{\mu}_0) + \boldsymbol{\zeta}_0, \tag{16}$$

where $\boldsymbol{\gamma}_0$ and $\boldsymbol{\zeta}_0$ are two parameter vectors which need to be tuned during the back-propagation.

We use $\Phi_l(\cdot)$ to represent the activation function in the neurons in the l-th hidden layer and $\forall l \in \{1, \cdots, m-1\}$. For simplicity, we apply $\boldsymbol{w}$ to represent overall parameters in the shared NN model. The latent feature $\boldsymbol{v}_p \in R^{q\times 1}$ is fed into the shared NN model at the input layer, where the connection matrix $\tilde{\boldsymbol{W}}_0 \in R^{q\times p_1}$ linearly combining the input features and deliver the result through the activation function to the first hidden layer formulated as $\boldsymbol{h}_1 = \Phi_0(\Psi_0(\tilde{\boldsymbol{W}}_0^T \boldsymbol{v}_p))$.

We define the connection matrix $\tilde{\boldsymbol{W}}_{m+1} \in R^{p_m\times K}$ connecting the m-th hidden layer and the output layer, the forward computation is formulated as

$$\boldsymbol{o}_p = \Phi_{m+1}(\Psi_{m+1}(\tilde{\boldsymbol{W}}_{m+1}^T \Phi_m(\Psi_m(\cdots \Phi_0(\Psi_0(\tilde{\boldsymbol{W}}_0^T \boldsymbol{v}_p)))))), \tag{17}$$

where $\boldsymbol{o}_p \in R^{K\times 1}$ denotes the output vector in the output layer.

Since the K outputs in vector $\boldsymbol{o}_p$ could be any number, we need to control the magnitude and the sign of the output value for any possible input data sample. Each output in $\boldsymbol{o}_p$ represents the score of a given data sample that belongs to one category. The category with highest score is the class the model has predicted for the given data sample. The original goal is to pick up the index with the largest output number from $\boldsymbol{o}_p = [o_1, \cdots, o_K]$, i.e.,

$$j^* = \underset{1\le j\le K}{\text{argmax}}\{o_1, o_2, \cdots, o_K\}. \tag{18}$$

Then, based on the idea that

$$\underset{1\le j\le K}{\text{argmax}}\{o_1, o_2, \cdots, o_K\} = \underset{1\le j\le K}{\text{argmax}}\{e^{o_1}, e^{o_2}, \cdots, e^{o_K}\}, \tag{19}$$

the output number o_j can be replaced by e^{o_j}. Furthermore, the logarithm of the sum of all $\{e^{o_j}\}_{j=1}^K$ approximates to the largest number of all numbers in $\boldsymbol{o}_p$ as

$$\log(\sum_{j=1}^{K} e^{o_j}) \approx \max_{1\le j\le k}\{o_1, o_2, \cdots, o_K\}. \tag{20}$$

We use $\hat{y}_i$ to represent the probability that the given data sample belongs to category i, i.e., $P(\boldsymbol{y}_p|o_i)$, which interprets the outputs of the shared NN model as probabilities of the input data sample belonging to the corresponding category. To formulate all the outputs of the shared NN model to be non-negative and sum to 1, $\hat{y}_i$ is designed as

$$\hat{y}_i = P(\boldsymbol{y}_p|o_i) = \frac{e^{o_i}}{\sum_{j=1}^K e^{o_j}}, \tag{21}$$

where $\boldsymbol{y}_p$ is the one-hot vector to indicate the true category that the given data sample belongs to. And we use vector $\hat{\boldsymbol{y}}_p$ to represent the estimated probability results for the given data sample where $\hat{y}_i$ is its i-th component, which is denoted as

$$\hat{\boldsymbol{y}}_p = P(\boldsymbol{y}_p|\boldsymbol{o}_p) = \left\{\frac{e^{o_i}}{\sum_{j=1}^K e^{o_j}}\right\}_{i=1}^K. \tag{22}$$

C. Initiation of Back-propagation by Local Loss

Then, the central server sends the estimated results $\hat{\boldsymbol{y}}_p$ to the corresponding IoT device which has the true label $\boldsymbol{y}_p$. We compare the estimated $\hat{\boldsymbol{y}}_p$ with the reality $\boldsymbol{y}_p$ by checking how probable $\boldsymbol{y}_p$ is. Maximizing this conditional probability $P(\boldsymbol{y}_p|\boldsymbol{o}_p)$ is equivalent to minimize the negative logarithm of this conditional probability $-\log\left(P(\boldsymbol{y}_p|\boldsymbol{o}_p)\right)$.

This formulation is purely from the idea that try to maximize the estimated probability of the ground truth. This idea can be connected with the cross-entropy loss function by the ground truth one-hot vector $\boldsymbol{y}_p$ as

$$\begin{aligned} -\log(P(\boldsymbol{y}_p|\boldsymbol{o}_p)) &= -\sum_{j=1}^{K} y_j \log(P(\boldsymbol{y}_p|o_j)) \\ &= -\sum_{j=1}^{K} y_j \log(\hat{y}_j). \end{aligned} \tag{23}$$

This expectation concept can be interpreted as picking up the negative logarithm of the ground truth probability $-\log(\hat{y}_{j^*})$ where only j^*-th component of the one-hot vector $\boldsymbol{y}_p$ equals to 1, the rest part of $\boldsymbol{y}_p$ are 0. The loss w.r.t. latent sample $\boldsymbol{v}_p$ is defined as

$$\begin{aligned} f(\boldsymbol{w}, \boldsymbol{v}_p) &= -\sum_{j=1}^{K} y_j \log\left(\frac{e^{o_j}}{\sum_{i=1}^K e^{o_i}}\right) \\ &= \log(\sum_{i=1}^{K} e^{o_i}) - \sum_{j=1}^{K} y_j o_j \\ &= \log(\sum_{i=1}^{K} e^{o_i}) - o_{j^*}. \end{aligned} \tag{24}$$

Then, the IoT device evaluates the local gradients based on its local labels and the outputs of the shared NN function as

$$\{\partial_{o_i} f(\boldsymbol{w}, \boldsymbol{v}_p)\}_{i=1}^K = \left\{\frac{e^{o_i}}{\sum_{j=1}^K e^{o_j}} - y_i\right\}_{i=1}^K. \tag{25}$$

Then, the local gradients are transmitted to the central server to update the parameters in the shared NN model.

D. Backward Computation in Shared Neural Network

The back-propagation for the shared NN model parameters tuning in the central server initiated by $\{\partial_{o_i} f(\boldsymbol{w}, \boldsymbol{v}_p)\}_{i=1}^K$ provided by the IoT devices. Following the chain rule, we define the back-propagated value at the output of the activation function of the l-th hidden layer as $\nabla_{\Phi_l} f$, and the back-propagated value at the output of Ψ_i is given by $\nabla_{\Psi_l}(\nabla_{\Phi_l} f)$. There are two parameters defined normalization function $\Psi_l(\cdot)$, i.e., $\boldsymbol{\gamma}_l$ and $\boldsymbol{\zeta}_l$. The gradients with respect to the two parameters over the given batch with size B are computed as

$$\begin{aligned} \nabla_{\boldsymbol{\zeta}_l}(\nabla_{\Psi_l}(\nabla_{\Phi_l} f)) &= \left\{\sum_{p=1}^{B} \frac{\partial_{\Phi_l} f}{\partial \Psi_{l,p}^i} \cdot \frac{\partial \Psi_{l,p}^i}{\partial \boldsymbol{\zeta}_l^i}\right\}_{i=1}^{p_l}, \\ \nabla_{\boldsymbol{\gamma}_l}(\nabla_{\Psi_l}(\nabla_{\Phi_l} f)) &= \left\{\sum_{p=1}^{B} \frac{\partial_{\Phi_l} f}{\partial \Psi_{l,p}^i} \cdot \frac{\partial \Psi_{l,p}^i}{\partial \boldsymbol{\gamma}_l^i}\right\}_{i=1}^{p_l}. \end{aligned} \tag{26}$$

We define

$$\hat{\boldsymbol{a}}_{l,p} = \frac{1}{\boldsymbol{\sigma}_l} \cdot (\boldsymbol{a}_{l,p} - \boldsymbol{\mu}_l), \tag{27}$$

and we obtain that

$$\hat{\boldsymbol{a}}_{l,p} = \left\{\sum_{p=1}^{B} \frac{\partial \Psi_{l,p}^i}{\partial \boldsymbol{\gamma}_l^i}\right\}_{i=1}^{p_l}. \tag{28}$$

To continue back-propagate in the l-th hidden layer w.r.t. function $\Psi_l(\cdot)$, we compute the gradient w.r.t. $\hat{\boldsymbol{a}}_{l,p}$ denoted by

$$\nabla_{\hat{\boldsymbol{a}}_{l,p}}(\nabla\Psi_l) = \left\{\frac{\partial\Psi_{l,p}^i}{\partial\hat{a}_{l,p}^i}\right\}_{i=1}^{p_l}. \tag{29}$$

Then, from the definition in (15) and (27), the back-propagation passes to $\boldsymbol{a}_l$ which is formulated by

$$\begin{aligned}&\nabla_{\boldsymbol{a}_l}(\nabla_{\Psi_l}(\nabla_{\Phi_l}f))\\ &= \left\{\frac{1}{\sigma_i}\frac{\partial_{\Phi_l}f}{\partial\hat{a}_{l,p}^i} + \frac{1}{m}\frac{\partial_{\Phi_l}f}{\partial\mu_l^i} + 2\cdot\frac{(a_{l,p}^i-\mu_l^i)}{m}\frac{\partial_{\Phi_l}f}{\partial\sigma_l^{i2}}\right\}_{i=1}^{p_l}.\end{aligned} \tag{30}$$

Then, based on the already-executed $\nabla_{\Psi_l}(\nabla_{\Phi_l}f)$, we obtain

$$\nabla_{\hat{\boldsymbol{a}}_{l,p}}(\nabla_{\Psi_l}(\nabla_{\Phi_l}f)) = \left\{\gamma_l^i\frac{\partial_{\Phi_l}f}{\partial\Psi_{l,p}^i}\right\}_{i=1}^{p_l}. \tag{31}$$

Then, the back-propagation has been updated to $\hat{\boldsymbol{a}}_{l,p}$ which is directly associated with $\boldsymbol{\mu}_l, \boldsymbol{\sigma}_l^2$ as shown in (27). Since $\boldsymbol{\mu}_l, \boldsymbol{\sigma}_l^2$ come from one batch of the passing pre-activation values, we obtain

$$\nabla_{\boldsymbol{\sigma}_l^2}\nabla_{\hat{\boldsymbol{a}}_l}(\nabla_{\Psi_l}(\nabla_{\Phi_l}f)) = \left\{\sum_{p=1}^{B}\frac{\partial_{\Phi_l}f}{\partial\Psi_{l,p}^i}\frac{\partial\Psi_{l,p}^i}{\partial\hat{a}_{l,p}^i}\frac{\partial\hat{a}_{l,p}^i}{\partial\sigma_l^{i2}}\right\}_{i=1}^{p_l}. \tag{32}$$

Because $\nabla_{\boldsymbol{\sigma}_l}\hat{\boldsymbol{a}}_l = -1/\boldsymbol{\sigma}_l^2$ and $\nabla_{\boldsymbol{\sigma}_l}\boldsymbol{\sigma}_l^2 = 2\boldsymbol{\sigma}_l$, we get $\nabla_{\boldsymbol{\sigma}_l^2}\hat{\boldsymbol{a}}_l = -1/2\boldsymbol{\sigma}_l^3$, which leads to

$$\nabla_{\boldsymbol{\sigma}_l^2}\nabla_{\hat{\boldsymbol{a}}_l}\Psi_l = -\left\{\frac{1}{2\sigma_l^{i3}}\sum_{p=1}^{B}\gamma_l^i\cdot(a_{l,p}^i-\mu_l^i)\right\}_{i=1}^{p_l}. \tag{33}$$

The gradient with respect to $\boldsymbol{\mu}_l$ is given by

$$\begin{aligned}&\nabla_{\boldsymbol{\mu}_l}\nabla_{\hat{\boldsymbol{a}}_l}(\nabla_{\Psi_l}(\nabla_{\Phi_l}f))\\ &= \left\{\left(\sum_{p=1}^{B}\frac{\partial_{\Phi_l}f}{\partial\hat{a}_{l,p}^i}\frac{\partial\hat{a}_{l,p}^i}{\partial\mu_l^i}\right) + \frac{\partial_{\Phi_l}f}{\partial\sigma_l^{i2}}\frac{\partial\sigma_l^{i2}}{\partial\mu_l^i}\right\}_{i=1}^{p_l}.\end{aligned} \tag{34}$$

Since $\nabla_{\boldsymbol{\mu}_l}\hat{\boldsymbol{a}}_l = -1/\boldsymbol{\sigma}_l$ and

$$\nabla_{\boldsymbol{\mu}_l}\boldsymbol{\sigma}_l^2 = -\frac{2}{B}\sum_{p=1}^{B}(\boldsymbol{a}_{l,p}-\boldsymbol{\mu}_l), \tag{35}$$

combining (31) and (33), we get

$$\begin{aligned}&\nabla_{\boldsymbol{\mu}_l}\nabla_{\hat{\boldsymbol{a}}_l}(\nabla_{\Psi_l}(\nabla_{\Phi_l}f))\\ &= -\sum_{p=1}^{B}\frac{\boldsymbol{\gamma}_l\cdot\nabla_{\Psi_l}(\nabla_{\Phi_l}f)}{\boldsymbol{\sigma}_l}\\ &+ \frac{\boldsymbol{\gamma}_l}{\boldsymbol{\sigma}_l^3B}(\sum_{p=1}^{B}\nabla_{\Psi_l}(\nabla_{\Phi_l}f)\cdot(\boldsymbol{a}_{l,p}-\boldsymbol{\mu}_l))\cdot(\sum_{p=1}^{B}(\boldsymbol{a}_{l,p}-\boldsymbol{\mu}_l)).\end{aligned} \tag{36}$$

Therefore, the back-propagation updated on the pre-activation output $\boldsymbol{a}_{l,p}$ is connected to $\nabla_{\Psi_l}(\nabla_{\Phi_l}f)$ as

$$\begin{aligned}&\nabla_{\boldsymbol{a}_l}(\nabla_{\Psi_l}(\nabla_{\Phi_l}f))\\ &= \frac{\boldsymbol{\gamma}_l(\nabla_{\Psi_l}(\nabla_{\Phi_l}f))}{\boldsymbol{\sigma}_l} - \frac{\boldsymbol{\gamma}_l}{\boldsymbol{\sigma}_lB}\sum_{p=1}^{B}(\nabla_{\Psi_l}(\nabla_{\Phi_l}f))\\ &+ \frac{\boldsymbol{\gamma}_l}{\boldsymbol{\sigma}_l^3B^2}(\sum_{p=1}^{B}(\nabla_{\Psi_l}(\nabla_{\Phi_l}f))\cdot(\boldsymbol{a}_{l,p}-\boldsymbol{\mu}_l))\cdot\sum_{p=1}^{B}(\boldsymbol{a}_{l,p}-\boldsymbol{\mu}_l)\\ &- \frac{(\boldsymbol{a}_{l,p}-\boldsymbol{\mu}_l)}{\boldsymbol{\sigma}_l^3B}\sum_{p=1}^{B}\boldsymbol{\gamma}_l(\nabla_{\Psi_l}(\nabla_{\Phi_l}f))\cdot(\boldsymbol{a}_{l,p}-\boldsymbol{\mu}_l).\end{aligned} \tag{37}$$

As we apply $\boldsymbol{w}_k$ to denote all the parameters in the shared NN model in the k-th round, $\boldsymbol{g}(\boldsymbol{w}_k)$ refers to the gradients information w.r.t. the parameters in the shared NN model in the k-th round. We use α_k to denote the step length parameter along the search direction and the current shared NN model updating is $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k\boldsymbol{g}(\boldsymbol{w}_k)$. And we applied detailed analysis to justify the learning rate α_k and convergence in Section V.

V. Convergence Analysis

Optimization techniques invest substantial effort in understanding and improving the region near the optimizers. This region holds paramount importance as it determines convergence and influences the overall performance of optimization algorithms. By focusing on this critical area, we strive to gain a comprehensive understanding of optimization dynamics and devise effective strategies to enhance convergence efficiency and optimize overall algorithm performance.

A. Quadratic Auxiliary Function

We design a perfect symmetric quadratic function, i.e., the condition number of this designed quadratic function's Hession is 1. The learning rate for updating the shared NN model α_k is used to control the curvature in every dimension where the larger the α_k, the wider the associated quadratic becomes. We define the quadratic function as

$$\begin{aligned}\varphi_{\alpha_k}(\boldsymbol{w}_{k+1}) &= \frac{1}{2\alpha_k}\left\|\boldsymbol{w}_{k+1}-\boldsymbol{w}_k\right\|^2\\ &+ \boldsymbol{g}(\boldsymbol{w}_k)^T(\boldsymbol{w}_{k+1}-\boldsymbol{w}_k) + f(\boldsymbol{w}_k).\end{aligned} \tag{38}$$

A large α_k that the quadratic approximation $\varphi_{\alpha_k}(\boldsymbol{w}_k)$ will be flat enough which lies completely above the $f(\boldsymbol{w})$ everywhere except at its point of tangency with $f(\boldsymbol{w})$. In this case, the minimum of $\varphi_{\alpha_k}(\boldsymbol{w}_k)$ definitely lies above $f(\boldsymbol{w})$ and the negative gradient direction leads to a smaller evaluation of $f(\boldsymbol{w})$.

We set the upper bound of the curvature of the global objective function $f(\boldsymbol{w})$ as L. Large L means much curved function. To guarantee the designed quadratic auxiliary function $\varphi(\boldsymbol{w})$ is completely above $f(\boldsymbol{w})$, the curvature of $\varphi(\boldsymbol{w})$ is defined by L. The curvature information of a function lies in its second derivatives. In order to determine its maximum curvature we must determine the largest possible eigenvalue (in magnitude) of its Hessian matrix, i.e.,

$$L = \max_{\boldsymbol{w}}\left\|\nabla^2 f(\boldsymbol{w})\right\|_2 \Rightarrow \max_{\boldsymbol{w}}\left\|\nabla^2\varphi(\boldsymbol{w})\right\|_2 = \frac{1}{\alpha_k}. \tag{39}$$

Once determined the curvature upper bound L, which means we determined the minimal learning rate α_k as we set $\alpha_k = \frac{1}{L}$. The global model updating becomes to

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \frac{1}{L}\boldsymbol{g}(\boldsymbol{w}_k). \tag{40}$$

We get $\varphi_{\frac{1}{L}}(\boldsymbol{w}_{k+1})$ as

$$\begin{aligned}\varphi_{\frac{1}{L}}(\boldsymbol{w}_{k+1}) = \frac{L}{2}\left\|\boldsymbol{w}_{k+1} - \boldsymbol{w}_k\right\|^2 \\ + \boldsymbol{g}(\boldsymbol{w}_k)^T(\boldsymbol{w}_{k+1} - \boldsymbol{w}_k) + f(\boldsymbol{w}_k).\end{aligned} \tag{41}$$

Because

$$L \cdot \boldsymbol{I} - \triangledown^2 f(\boldsymbol{w}) \succeq \boldsymbol{0}, \tag{42}$$

thus, for any $\boldsymbol{w}$ we have

$$\boldsymbol{w}^{\boldsymbol{T}}(L \cdot \boldsymbol{I} - \boldsymbol{\nabla^2} f(\boldsymbol{w}))\boldsymbol{w} \geq 0 \tag{43}$$

which leads to

$$\varphi_{\frac{1}{L}}(\boldsymbol{w}_{k+1}) \geq f(\boldsymbol{w}_{k+1}), \tag{44}$$

which guarantees $f(\boldsymbol{w})$ to always descend. In practice, we should use $\alpha_k = \frac{1}{L}$ as a benchmark to search for larger convergence-forcing fixed step length values. Plugging (40) into (44), we get

$$\begin{aligned}f(\boldsymbol{w}_{k+1}) &\leq \frac{1}{2L}\boldsymbol{g}(\boldsymbol{w}_k)^T\boldsymbol{g}(\boldsymbol{w}_k) - \frac{1}{L}\boldsymbol{g}(\boldsymbol{w}_k)^T\boldsymbol{g}(\boldsymbol{w}_k) + f(\boldsymbol{w}_k) \\ &= f(\boldsymbol{w}_k) - \frac{1}{2L}\boldsymbol{g}(\boldsymbol{w}_k)^T\boldsymbol{g}(\boldsymbol{w}_k).\end{aligned} \tag{45}$$

Since $\boldsymbol{g}(\boldsymbol{w}_k)^T\boldsymbol{g}(\boldsymbol{w}_k) \geq 0$, the objective evaluation decreases at each updating. To prove that the magnitude of the gradient will become sufficiently small which means that it converges to a stationary point, we accumulate an infinite sequence of the magnitude of gradients, and then to proof this accumulation is a finite number which gives us the idea that later components in this infinite sequence should be sufficiently small. Following this idea, we subtract $f(\boldsymbol{w}_k)$ from both sides of (45) and accumulate the results from $k = 0$ to infinite as

$$\sum_{k=0}^{\infty}(f(\boldsymbol{w}_{k+1}) - f(\boldsymbol{w}_k)) \leq -\frac{1}{2L}\sum_{k=0}^{\infty}\boldsymbol{g}(\boldsymbol{w}_k)^T\boldsymbol{g}(\boldsymbol{w}_k). \tag{46}$$

And we know that

$$\sum_{k=0}^{K}(f(\boldsymbol{w}_{k+1}) - f(\boldsymbol{w}_k)) \geq f(\boldsymbol{w}^*) - f(\boldsymbol{w}_0), \tag{47}$$

where $\boldsymbol{w}^*$ is the optimal solution to minimize $f(\boldsymbol{w})$. Obviously, $f(\boldsymbol{w}^*) - f(\boldsymbol{w}_0) \neq -\infty$, that means

$$\sum_{k=0}^{\infty}\boldsymbol{g}(\boldsymbol{w}_k)^T\boldsymbol{g}(\boldsymbol{w}_k) < +\infty \Rightarrow \lim_{k\to\infty}\boldsymbol{g}(\boldsymbol{w}_k)^T\boldsymbol{g}(\boldsymbol{w}_k) = 0. \tag{48}$$

The gradient will finally vanish, and this conclusion can be achieved by using any smaller than $\alpha_k \leq \frac{1}{L}$.

B. Convergence Speed Analysis

The one iteration reduction for the shared NN model is bounded by

$$\begin{aligned}||\boldsymbol{w}_{k+1} - \boldsymbol{w}^*||^2 &= ||\boldsymbol{w}_k - \alpha_k\boldsymbol{g}(\boldsymbol{w}_k) - \boldsymbol{w}^*||^2 \\ &= ||\boldsymbol{w}^* - \boldsymbol{w}_k||^2 - 2\alpha_k\boldsymbol{g}(\boldsymbol{w}_k)^T(\boldsymbol{w}_k - \boldsymbol{w}^*) \\ &\quad + \alpha_k^2||\boldsymbol{g}(\boldsymbol{w_k})||^{\boldsymbol{2}}.\end{aligned} \tag{49}$$

We need to set lower bound for the inner product $\boldsymbol{g}(\boldsymbol{w}_k)^T(\boldsymbol{w}_k - \boldsymbol{w}^*)$, since when the search direction tends to become asymptotically orthogonal to the gradient direction, the updating will get stuck. Analysis can be shown as follows.

According to the quadratic upper bound and the linear lower bound of the global objective function, we obtain the inequality as

$$\begin{aligned}&f(\boldsymbol{w}_{k+1}) - f(\boldsymbol{w}_k) = f(\boldsymbol{w}_{k+1}) - f(\boldsymbol{z}) + f(\boldsymbol{z}) - f(\boldsymbol{w}_k) \\ &\leq \boldsymbol{g}(\boldsymbol{w}_{k+1})^T(\boldsymbol{w}_{k+1} - \boldsymbol{z}) + \boldsymbol{g}(\boldsymbol{w}_k)^T(\boldsymbol{z} - \boldsymbol{w}_k) + \frac{L}{2}||\boldsymbol{z} - \boldsymbol{w}_k||^2 \\ &= \boldsymbol{g}(\boldsymbol{w}_{k+1})^T(\boldsymbol{w}_{k+1} - \boldsymbol{w}_k) + (\boldsymbol{g}(\boldsymbol{w}_{k+1}) - \boldsymbol{g}(\boldsymbol{w}_k))^T(\boldsymbol{w}_k - \boldsymbol{z}) \\ &+ \frac{L}{2}||\boldsymbol{z} - \boldsymbol{w}_k||^2.\end{aligned} \tag{50}$$

We define

$$\boldsymbol{z} = \boldsymbol{w}_k - \frac{1}{L}(\boldsymbol{g}(\boldsymbol{w}_k) - \boldsymbol{g}(\boldsymbol{w}_{k+1})), \tag{51}$$

then, we obtain

$$\begin{aligned}&(\boldsymbol{g}(\boldsymbol{w}_{k+1}) - \boldsymbol{g}(\boldsymbol{w}_k))^T(\boldsymbol{w}_k - \boldsymbol{z}) \\ &= -\frac{1}{L}||(\boldsymbol{g}(\boldsymbol{w}_{k+1}) - \boldsymbol{g}(\boldsymbol{w}_k))||^2.\end{aligned} \tag{52}$$

And since

$$\frac{L}{2}||\boldsymbol{z} - \boldsymbol{w}_k||^2 = \frac{1}{2L}||(\boldsymbol{g}(\boldsymbol{w}_{k+1}) - \boldsymbol{g}(\boldsymbol{w}_k))||^2, \tag{53}$$

we obtain

$$\begin{aligned}&\boldsymbol{g}(\boldsymbol{w}_k)^T(\boldsymbol{w}_{k+1} - \boldsymbol{w}_k) \leq f(\boldsymbol{w}_{k+1}) - f(\boldsymbol{w}_k) \\ &\leq \boldsymbol{g}(\boldsymbol{w}_{k+1})^T(\boldsymbol{w}_{k+1} - \boldsymbol{w}_k) - \frac{1}{2L}||(\boldsymbol{g}(\boldsymbol{w}_{k+1}) - \boldsymbol{g}(\boldsymbol{w}_k))||^2\end{aligned} \tag{54}$$

which leads to

$$\begin{aligned}&(\boldsymbol{g}(\boldsymbol{w}_{k+1}) - \boldsymbol{g}(\boldsymbol{w}_k))^T(\boldsymbol{w}_{k+1} - \boldsymbol{w}_k) \\ &\geq \frac{1}{2L}||(\boldsymbol{g}(\boldsymbol{w}_{k+1}) - \boldsymbol{g}(\boldsymbol{w}_k))||^2.\end{aligned} \tag{55}$$

We define $\psi(\boldsymbol{w}) = f(\boldsymbol{w}) - \frac{\tau}{2}||\boldsymbol{w}||^2$. Then, we have

$$\begin{aligned}&(\boldsymbol{\nabla}\psi(\boldsymbol{w}^*) - \boldsymbol{\nabla}\psi(\boldsymbol{w}_k)^T(\boldsymbol{w}^* - \boldsymbol{w}_k) \\ &\geq \frac{1}{L - \tau}||\boldsymbol{\nabla}\psi(\boldsymbol{w}^*) - \boldsymbol{\nabla}\psi(\boldsymbol{w}_k)||^2,\end{aligned} \tag{56}$$

which leads to

$$\begin{aligned}&(\boldsymbol{g}(\boldsymbol{w}^*) - \boldsymbol{g}(\boldsymbol{w}_k))^T(\boldsymbol{w}^* - \boldsymbol{w}_k) - \tau||\boldsymbol{w}^* - \boldsymbol{w}_k||^2 \\ &\geq \frac{1}{L - \tau}||\boldsymbol{g}(\boldsymbol{w}^*) - \boldsymbol{g}(\boldsymbol{w}_k) - \tau(\boldsymbol{w}^* - \boldsymbol{w}_k)||^2.\end{aligned} \tag{57}$$

We continue to simplify (57) as

$$\begin{aligned}&(\boldsymbol{g}(\boldsymbol{w}^*) - \boldsymbol{g}(\boldsymbol{w}_k))^T(\boldsymbol{w}^* - \boldsymbol{w}_k) \\ &\geq \frac{\tau L}{L + \tau}||\boldsymbol{w}^* - \boldsymbol{w}_k||^2 + \frac{1}{L + \tau}||\boldsymbol{g}(\boldsymbol{w}^*) - \boldsymbol{g}(\boldsymbol{w}_k)||^2.\end{aligned} \tag{58}$$

Since $\boldsymbol{g}(\boldsymbol{w}^*) = \mathbf{0}$, we obtain

$$\begin{aligned}\boldsymbol{g}(\boldsymbol{w}_k)^T(\boldsymbol{w}_k - \boldsymbol{w}^*) \geq \frac{\tau L}{L+\tau}||\boldsymbol{w}_k - \boldsymbol{w}^*||^2 \\ + \frac{1}{L+\tau}||\boldsymbol{g}(\boldsymbol{w}_k)||^2.\end{aligned} \tag{59}$$

Thus, one step reduction can be bounded by

$$\begin{aligned}||\boldsymbol{w}_{k+1} - \boldsymbol{w}^*||^2 \leq \frac{L+\tau-2\alpha_k\tau L}{L+\tau}||\boldsymbol{w}_k - \boldsymbol{w}^*||^2 \\ + (\alpha_k^2 - \frac{2\alpha_k}{L+\tau})||\boldsymbol{g}(\boldsymbol{w}_k)||^2.\end{aligned} \tag{60}$$

According to

$$||\boldsymbol{g}(\boldsymbol{w}_k)||^2 = ||\boldsymbol{g}(\boldsymbol{w}_k) - \boldsymbol{g}(\boldsymbol{w}^*)||^2 \leq ||\boldsymbol{w}_k - \boldsymbol{w}^*||^2, \tag{61}$$

we rewrite (60) as

$$\begin{aligned}||\boldsymbol{w}_{k+1} - \boldsymbol{w}^*||^2 \leq \frac{L+\tau-2\alpha_k\tau L}{L+\tau}||\boldsymbol{w}_k - \boldsymbol{w}^*||^2 \\ + (\alpha_k^2 - \frac{2\alpha_k}{L+\tau})L^2||\boldsymbol{w}_k - \boldsymbol{w}^*||^2.\end{aligned} \tag{62}$$

And according to (62) and the , we set $\alpha_k = \frac{2}{L+\tau}$ to obtain

$$\begin{aligned}||\boldsymbol{w}_{k+1} - \boldsymbol{w}^*||^2 \leq (\frac{L-\tau}{L+\tau})^2||\boldsymbol{w}_k - \boldsymbol{w}^*||^2 \\ = (\frac{\kappa-1}{\kappa+1})^2||\boldsymbol{w}_k - \boldsymbol{w}^*||^2.\end{aligned} \tag{63}$$

By unrolling the recursion in (63), we obtain that

$$||\boldsymbol{w}_{k+1} - \boldsymbol{w}^*||^2 \leq (\frac{\kappa-1}{\kappa+1})^{2k}||\boldsymbol{w}_0 - \boldsymbol{w}^*||^2, \tag{64}$$

and combining with the quadratic upper bound

$$\begin{aligned}f(\boldsymbol{w}_{k+1}) \leq f(\boldsymbol{w}^*) + \boldsymbol{g}(\boldsymbol{w}^*)^T(\boldsymbol{w}_{k+1} - \boldsymbol{w}^*) \\ + \frac{L}{2}||\boldsymbol{w}_{k+1} - \boldsymbol{w}^*||^2 \\ = f(\boldsymbol{w}^*) + \frac{L}{2}||\boldsymbol{w}_{k+1} - \boldsymbol{w}^*||^2,\end{aligned} \tag{65}$$

we obtain

$$\begin{aligned}f(\boldsymbol{w}_{k+1}) - f(\boldsymbol{w}^*) \leq \frac{L}{2}||\boldsymbol{w}_{k+1} - \boldsymbol{w}^*||^2 \\ \leq \frac{L}{2}e^{-\frac{4k}{\kappa+1}}||\boldsymbol{w}_0 - \boldsymbol{w}^*||^2.\end{aligned} \tag{66}$$

Based on (66), we conclude that iterate $\boldsymbol{w}_{k+1}$ achieves approximation accuracy $|f(\boldsymbol{w}_{k+1}) - f(\boldsymbol{w}^*)| \leq \epsilon$ as long as k satisfies

$$k \geq \frac{\kappa+1}{4}\log\frac{L||\boldsymbol{w}_0 - \boldsymbol{w}^*||^2}{2\epsilon}. \tag{67}$$

VI. Performance Evaluation

In this section, we conduct extensive simulations to evaluate the performance of the proposed scheme. We use a diverse set of synthetic local datasets that emulate real-world healthcare IoT data characteristics. These simulations provide practical insights into the effectiveness of our method. We employ centralized learning with overall features (CELF) as a benchmark. It is important to note that CELF is considered an ideal solution but may not be practical in real-world scenarios since it processes all features together. We compare our proposed Collaborative Learning with Adaptive Latent Feature (COLAF) against this benchmark to assess its promising capabilities. Furthermore, we compare COLAF with individual learning based on local features (ILLF) to highlight the significant improvements achieved by our approach. By presenting the simulation results and benchmark comparisons, we demonstrate the practical benefits and effectiveness of our proposed method for collaborative learning with heterogeneous healthcare IoT devices.

A. Local Data Set Synthesis

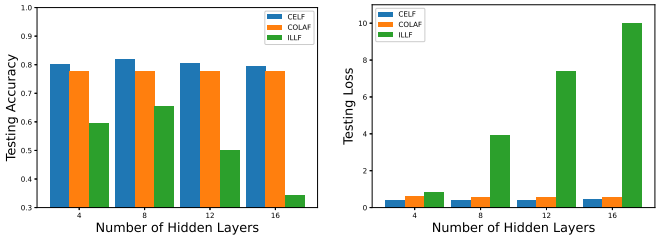
We begin by describing the original dataset, which consists of samples with six features: monitoring time (TIME), sugar level (SL), EEG monitoring rate (EEG), blood pressure (BP), heart beat rate (HR), and blood circulation (CIRCULATION). After removing outliers, we are left with a total of 14203 samples. Among these samples, approximately 29.1% are associated with the activity of Falling. To facilitate our evaluation, we divide the dataset into a training set containing 9516 samples and a testing set containing 4687 samples. In our scenario, IoT devices in institutions collect samples with features TIME, EEG, and CIRCULATION. The remaining features are captured by personal IoT devices. Specifically, we consider three types of personal IoT devices, each measuring a different physical sign: SL, BP, and HR. The 9516 training samples are divided into multiple local datasets, with each type of IoT device having its own set of local data. The test dataset for each individual IoT device comprises 4687 samples with their associated features. Given the varying sizes of the local training datasets and our goal of comparing training approaches with the same number of epochs, we define the batch size rate as the percentage of one batch in the entire training dataset. This ensures that the number of updates each epoch remains consistent across different local datasets. Controlling the number of updates is crucial as it greatly affects the training results.

B. Performance with Various Neural Network Architectures

To evaluate the effectiveness of our proposed method, we performed experiments using different neural network architectures with varying numbers of hidden layers and neurons per layer. In this experiment, we set the total number of epochs to 1000 and the batch size rate to 1. For our proposed training method to generate latent features, the local model parameters of each individual IoT device underwent 200 training epochs with a batch size rate of 0.1. The dimension of the latent space is defined as 6, and there are 20 IoT devices participating in the collaborative training for each type. These experiments aimed to assess the performance of our approach across a range of the shared neural network configurations, providing valuable insights into its effectiveness and robustness.

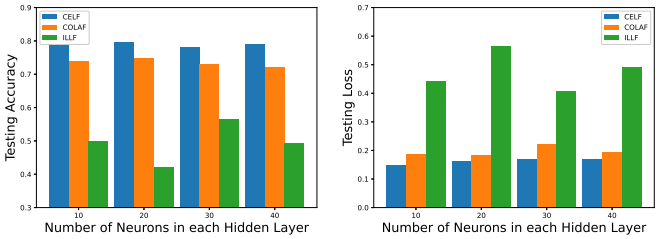
CELF serves as an ideal benchmark is evaluated by the original test data set and is not intended for practical implementation in healthcare IoT applications. For both COLAF and ILLF, we simulate heterogeneous local features by partitioning the original test dataset samples, enabling us to generate local testing datasets that reflect the realistic behavior of healthcare IoT devices. The performance of COLAF and ILLF is then evaluated using the corresponding local features from the test

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3307675



(a) Accuracy (b) Loss

Fig. 2. Different number of hidden layers.



(a) Accuracy (b) Loss

Fig. 3. Different number of neurons on each hidden layer

dataset of each IoT device. Fig. 2 illustrates the averaged testing accuracy and testing loss results of COLAF and ILLF across all participating IoT devices, providing a more practical and meaningful evaluation of their performance in healthcare IoT applications.

As depicted in Fig. 2, we evaluated the impact of varying the number of hidden layers in the shared neural network model. The range of hidden layers tested was from 4 to 16, with each layer containing 10 neurons. The results consistently demonstrated that as the number of hidden layers increased, our proposed COLAF approach achieved performance levels that close to the benchmark set by CELF, the global training scheme. In contrast, the benchmark ILLF, which relies solely on individual IoT devices, performed significantly worse due to limited access to comprehensive data. Even assuming individual IoT devices have sufficient computing capabilities for training large neural network models, COLAF still outperformed ILLF. These findings emphasize the effectiveness of collaborative learning in enhancing the overall performance of healthcare IoT devices, showcasing its potential for driving advancements in intelligent healthcare applications.

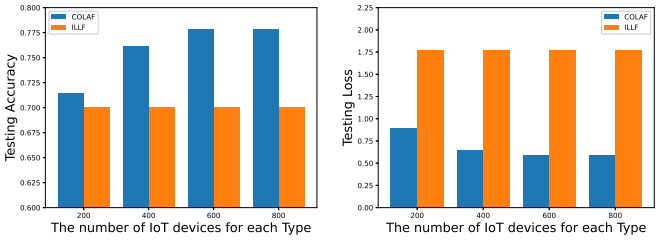
Then, we investigate the impact of varying the number of neurons in each hidden layer of the shared neural network architecture. We consider a range of values from 10 to 40 neurons in each hidden layer, while maintaining a total of 8 hidden layers. The results, depicted in Fig. 3, clearly demonstrate the superiority of the proposed COLAF over the ILLF approach. The performance of COLAF approaches that of the CELF, indicating its ability to achieve comparable performance while overcoming the limitations of individual training. Notably, the optimal performance of COLAF is achieved with 20 neurons in each hidden layer, as shown in Fig. 3(a). As the complexity of the shared neural network model increases beyond this point, the performance of COLAF slightly deteriorates due to the limitations of the latent features' representation ability.

In conclusion, the proposed COLAF demonstrates its capability to effectively train relatively complex neural network models by leveraging the latent local features from healthcare IoT devices. It achieves performance levels comparable to CELF, which is not practical for real-world scenarios, while surpassing the performance of ILLF trained by individual IoT devices.

C. Number of IoT Devices Participating in Collaborative Learning

To investigate the impact of the number of IoT devices participating in the collaborative learning process, we conducted experiments with varying numbers of IoT devices while keeping the number of samples held by each device fixed. We performed 1000 training epochs with a batch rate of 1 for each configuration. We define the aligned latent feature dimension as 6, the parameters of the local encoders were trained by 200 epochs. The shared neural network architecture consisted of 8 hidden layers, each with 10 neurons, which was consistent across all following experiments.

The experimental results, depicted in Fig. 4, reveal that increasing the number of IoT devices participating in the collaborative learning process significantly improves its performance when the number of devices is relatively small as shown in Fig. 4 with 200 to 600 IoT devices. However, as the number of devices collaborating reaches 600, the performance improvement becomes marginal. Continuing to add more IoT devices beyond this point offers only slight enhancements to COLAF's performance as shown with 800 participated IoT devices. In summary, the experiments highlight the diminishing returns of including additional IoT devices in the collaborative learning process once a sufficient number of devices are already collaborating. This finding underscores the importance of optimizing the IoT device participation to strike a balance between performance gains and resource utilization to combining more IoT devices.



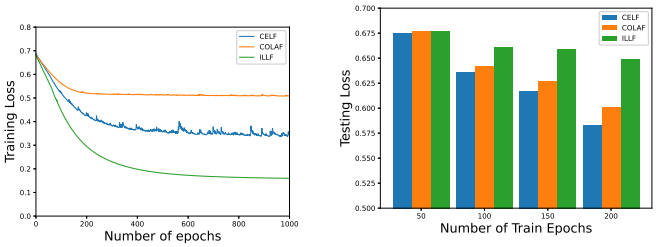
(a) Accuracy (b) Testing Loss

Fig. 4. Different number of collaborated IoT devices

Figure 5 presents the convergence speed of the proposed COLAF, compared with CELF and ILLF, when 200 IoT devices of each type participate in the collaborative training. The local model of each IoT device develops the latent space in 6 dimensions and the local parameters are trained over 200 epochs with a batch rate of 0.1.

In Fig. 5(a), it can be observed that the ILLF demonstrates a significantly faster convergence speed compared to both COLAF and CELF during the training procedure. Moreover, ILLF consistently outperforms the other two training schemes in terms of training loss. This rapid convergence of ILLF is particularly evident when the number of samples available to each IoT device is limited. However, while ILLF achieves

superior training loss, its testing performance is significantly poorer than the other two schemes, as shown in Fig. 5(b). This is due to the limited training samples available to individual IoT devices, which prevents them from acquiring a comprehensive understanding of the samples in the test dataset. As a result, the locally trained model becomes highly overfit to the limited local samples, leading to poor testing performance. It is worth noting that the faster convergence of training loss in ILLF for smaller training datasets does not necessarily indicate better model performance in practical applications. To achieve reliable and robust model performance, it is crucial to consider a broader scope beyond the limitations of limited local datasets which leading to the motivation of the proposed COLAF.



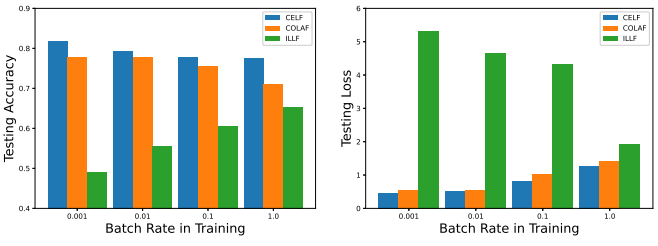
(a) Training Loss (b) Testing Loss

Fig. 5. Convergence analysis.

D. Impact of Different Batch Rates

The choice of batch size plays a crucial role in training results, as it determines the number of updates made during each epoch based on the training dataset size. Since the training datasets differ among the three schemes (ILLF, COLAF, and CELF), batch rates need to be defined to ensure an equal number of updating steps across schemes within each epoch. Fig. 6 demonstrates that the proposed COLAF consistently outperforms ILLF and achieves performance levels closer to CELF. The model in this experiment comprises 8 hidden layers with 10 neurons in each hidden layer, trained over 1000 epochs, and involves the participation of 200 IoT devices for each type in the collaborative training.

The ILLF initially exhibits poor performance with a batch rate of 0.001, primarily due to severe overfitting. As the batch rate increases from 0.001 to 1, the issue of overfitting gradually diminishes in ILLF. However, both CELF and COLAF experience a slight reduction in performance as the batch rate increases, owing to fewer updates being made. Overall, the results highlight the effectiveness of COLAF compared to ILLF, as it consistently achieves better performance and closely approaches the performance of CELF.



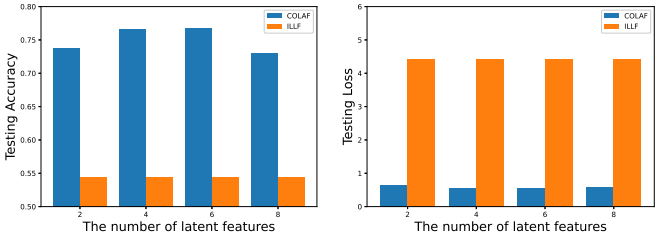
(a) Accuracy (b) Loss

Fig. 6. Impact of Different Batch Rates during Training Procedure.

E. Number of Latent Feature Space Dimensions

The dimension of the latent feature space is a critical aspect of COLAF's design. To explore the impact of the latent feature dimension, we conducted experiments with a range of latent feature dimensions, specifically from 2 to 8, as depicted in Fig. 7. The model was trained over 1000 epochs with a batch rate of 1, using samples collected from 200 IoT devices of each type. The shared NN model consisted of 4 hidden layers, each with 10 neurons. The results revealed that as the number of latent feature dimensions increased within a reasonable range, the performance of COLAF surpassed that of ILLF. Initially, as the dimensionality of the latent features increased, the performance of the collaborative trained model improved due to enhanced representation capabilities. However, beyond a certain point, when the latent feature dimensions continued to increase, the performance of the collaborative trained model began to decline, likely due to sparse latent features.

The results underscore the importance of choosing an appropriate dimension for the latent feature space in COLAF, striking a balance between representation ability and avoiding over-sparsity. The collaborative training approach demonstrated its effectiveness in improving performance compared to individual training, highlighting the potential of leveraging latent features for enhanced healthcare IoT analytics.



(a) Accuracy (b) Loss

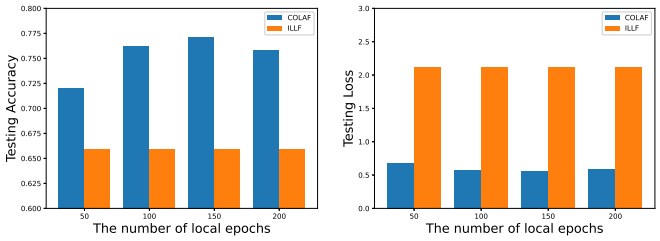
Fig. 7. Different number of dimensions for latent feature space.

F. Impact of Local Training of IoT Devices

Addressing the work load of IoT devices to train the local model is a crucial consideration in COLAF. In Fig. 8, we explore the relationship between the local work loads w.r.t. the number of local training epochs within IoT devices and the performance of COLAF. The shared NN model used for this experiment with COLAF and the individual model with ILLF consist of 4 hidden layers, with 10 neurons in each hidden layer. It is trained over 1000 epochs with a batch rate of 1. For each type of IoT device, we have 200 different devices participating in the collaborative training scheme. Each IoT device trains a local model capable of generating a latent feature space with 6 dimensions.

The results clearly demonstrate that the proposed COLAF consistently outperforms ILLF across a range of local computing settings, varying from 50 to 200 local training epochs with COLAF. As the number of training epoch within the local IoT devices increases, there is an initial improvement in the testing accuracy and loss of COLAF. However, as the number of local training epochs reaches 200, there is a decline in performance of COLAF due to overfitting of the local model to the local training samples. These findings

highlight the trade-off between local work loads within IoT devices and performance in COLAF. While increasing the local work loads can initially improve performance, there is a limit beyond which further local training efforts leads to diminishing returns. Achieving an optimal balance is crucial to maximize the benefits of collaborative learning in healthcare IoT applications.



(a) Accuracy (b) Loss

Fig. 8. Convergence of Local Model Training.

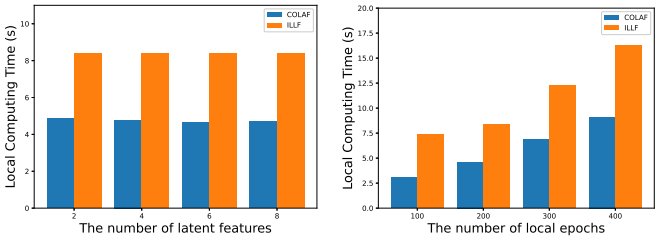
G. Computing Cost on IoT Devices

Furthermore, the proposed COLAF significantly reduces the computing cost on each IoT device compared to the ILLF, as demonstrated by the computing time required during the training phase. In Fig. 9, we present a comparison of the computing cost between COLAF and the benchmark ILLF by varying the number of latent features and local training epochs in COLAF. The evaluation of ILLF is the average computing time of conducting 200 training epochs with the NN model overall participated IoT devices. To be fair with the comparison, the evaluation of the local computing cost in each IoT device of COLAF also covers 200 local epochs with varying the latent space dimensions while keeping the training epochs constant as shown in Fig. 9(a). Then, we evaluate the average computing time of COLAF by fixing the latent feature dimension while increasing the local training epochs as shown in Fig. 9(b).

The results clearly demonstrate that COLAF significantly reduces the computing time on IoT devices compared to the ILLF. Although the computing time increases as the local model training epochs range from 100 to 400, the increment remains negligible compared to the computing cost associated with the ILLF. Fig. 8 further supports these findings, indicating that the collaborative trained model achieves superior performance by conducting no more than 200 training epochs on the local model, surpassing the performance of the ILLF. This highlights the efficiency and effectiveness of COLAF in reducing computing cost while still achieving remarkable performance compared to the ILLF.

VII. Conclusion

In this paper, we introduce a novel Collaborative Learning architecture designed to leverage the abundant healthcare data gathered by diverse IoT devices. Our architecture is capable of adapting to the diverse local feature spaces of different IoT devices and enables collaborative training of deep neural network models, making it applicable to various healthcare IoT devices. By shifting the majority of the training workload to a central server, our collaborative learning architecture alleviates the computational burden on individual IoT devices. We evaluate the performance of our proposed method through a series of experiments, demonstrating its reliability and promise. The results highlight the effectiveness of our collaborative learning framework in integrating different types of healthcare data collected from heterogeneous IoT devices. In conclusion, our collaborative learning framework offers a valuable solution for the seamless integration of diverse healthcare data from heterogeneous IoT devices. Through alignment of local feature dimensions and guaranteed data privacy, it paves the way for enhanced intelligent healthcare applications.



(a) (b)

Fig. 9. Average Computing Time on IoT Devices

How to enhance the collaboration intelligence by incorporating diverse local features through the implementation of compressive learning to process the local latent feature vectors will be an important future research issue. Our focus will be on optimizing the collaborative performance concerning the sketching of the entire set of local latent features, which holds great promise in effectively reducing the communication burden within our proposed collaborative framework.

Acknowledgments

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), Compute Canada, and British Columbia Knowledge Development Fund (BCKDF).

References

[1] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Transactions on Mobile Computing*, vol. 21, no. 8, pp. 2818–2832, 2020.

[2] F. Haider, S. De La Fuente, and S. Luz, "An assessment of paralinguistic acoustic features for detection of alzheimer's dementia in spontaneous speech," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 2, pp. 272–281, 2019.

[3] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.

[4] K. Kaur, S. Garg, G. Kaddoum, E. Bou-Harb, and K.-K. R. Choo, "A big data-enabled consolidated framework for energy efficient software defined data centers in iot setups," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2687–2697, 2019.

[5] H. Elayan, M. Aloqaily, and M. Guizani, "Sustainability of healthcare data analysis iot-based systems using deep federated learning," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7338–7346, 2021.

[6] X. Lin, R. Lu, X. Shen, Y. Nemoto, and N. Kato, "Sage: a strong privacy-preserving scheme against global eavesdropping for ehealth systems," *IEEE journal on selected areas in communications*, vol. 27, no. 4, pp. 365–378, 2009.

[7] R. Lu, X. Lin, and X. Shen, "Spoc: A secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency," *IEEE transactions on parallel and distributed systems*, vol. 24, no. 3, pp. 614–624, 2012.

[8] L. Zhao, X. Lan, L. Cai, and J. Pan, "Adaptive content placement in edge networks based on hybrid user preference learning," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[9] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[10] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[11] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2019.

[12] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2022–2035, 2020.

[13] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2134–2143, 2019.

[14] V. Kulkarni, M. Kulkarni, and A. Pant, "Survey of personalization techniques for federated learning," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. IEEE, 2020, pp. 794–797.

[15] M. V. Perez, K. W. Mahaffey, H. Hedlin, J. S. Rumsfeld, A. Garcia, T. Ferris, V. Balasubramanian, A. M. Russo, A. Rajmane, L. Cheung *et al.*, "Large-scale assessment of a smartwatch to identify atrial fibrillation," *New England Journal of Medicine*, vol. 381, no. 20, pp. 1909–1917, 2019.

[16] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Briefings in bioinformatics*, vol. 19, no. 6, pp. 1236–1246, 2018.

[17] X. Min, B. Yu, and F. Wang, "Predictive modeling of the hospital readmission risk from patients' claims data using machine learning: a case study on copd," *Scientific reports*, vol. 9, no. 1, p. 2362, 2019.

[18] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," *arXiv preprint arXiv:1812.00564*, 2018.

[19] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *Journal of Healthcare Informatics Research*, vol. 5, pp. 1–19, 2021.

[20] S. Boughorbel, F. Jarray, N. Venugopal, S. Moosa, H. Elhadi, and M. Makhlouf, "Federated uncertainty-aware learning for distributed hospital ehr data," *arXiv preprint arXiv:1910.12191*, 2019.

[21] R. Duan, M. R. Boland, Z. Liu, Y. Liu, H. H. Chang, H. Xu, H. Chu, C. H. Schmid, C. B. Forrest, J. H. Holmes *et al.*, "Learning from electronic health records across multiple sites: A communication-efficient and privacy-preserving distributed algorithm," *Journal of the American Medical Informatics Association*, vol. 27, no. 3, pp. 376–385, 2020.

[22] Y. Kim, J. Sun, H. Yu, and X. Jiang, "Federated tensor factorization for computational phenotyping," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 887–895.

[23] D. Liu, D. Dligach, and T. Miller, "Two-stage federated phenotyping and patient representation learning," in *Proceedings of the conference. Association for Computational Linguistics. Meeting*, vol. 2019. NIH Public Access, 2019, p. 283.

[24] L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng, and D. Liu, "Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records," *Journal of biomedical informatics*, vol. 99, p. 103291, 2019.

[25] S. Silva, B. A. Gutman, E. Romero, P. M. Thompson, A. Altmann, and M. Lorenzi, "Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data," in *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*. IEEE, 2019, pp. 270–274.

[26] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *International journal of medical informatics*, vol. 112, pp. 59–67, 2018.

[27] B. Yuan, S. Ge, and W. Xing, "A federated learning framework for healthcare iot devices," *arXiv preprint arXiv:2005.05083*, 2020.

[28] O. Gupta and R. Raskar, "Distributed learning of deep neural network over multiple agents," *Journal of Network and Computer Applications*, vol. 116, pp. 1–8, 2018.

[29] 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," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[30] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[31] D. Basu, D. Data, C. Karakus, and S. Diggavi, "Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[32] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-iid federated learning," *arXiv preprint arXiv:2101.11203*, 2021.

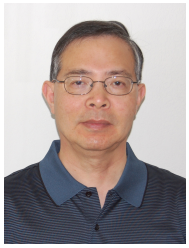
[33] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5132–5143.

[34] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[35] A. Khaled, K. Mishchenko, and P. Richtárik, "Tighter theory for local sgd on identical and heterogeneous data," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 4519–4529.

Lei Zhao (S'17) received the B.S. and M.A.Sc. degrees in computer science and technology from Xidian University, Xi'an, China, in 2015 and 2018, respectively. He is currently pursuing the PhD degree at the Department of Electrical & Computer Engineering at the University of Victoria, Victoria, B.C., Canada. His current research interests include federated learning and optimization.

Lin Cai (S'00-M'06-SM'10-F'20) received her M.A.Sc. and Ph. D. degrees (awarded Outstanding Achievement in Graduate Studies) in electrical and computer engineering from the University of Waterloo, Waterloo, Canada, in 2002 and 2005, respectively. Since 2005, she has been with the Department of Electrical & Computer Engineering at the University of Victoria, and she is currently a Professor. She is an NSERC E.W.R. Steacie Memorial Fellow, an Engineering Institute of Canada (EIC) Fellow, a Canadian Academy of Engineering (CAE) Fellow, a Royal Society of Canada (RSC) College Member, and an IEEE Fellow.



Wu-Sheng Lu Wu-Sheng Lu (F'99-LF'12) received the B.Sc. degree in Mathematics from Fudan University, Shanghai, China, in 1964, the M.S. degree in electrical engineering, and the Ph.D. degree in control science from the University of Minnesota, Minneapolis, USA, in 1983 and 1984, respectively. Since 1987, he has been with the University of Victoria, Victoria, B.C., Canada, and is now Professor Emeritus. He is the co-author with A. Antoniou of Two-Dimensional Digital Filters (Marcel Dekker, 1992) and Practical Optimization: Algorithms and Engineering Applications (2nd ed., Springer, 2021), and with E. K. P. Chong and S. H. Zak of An Introduction to Optimization (5th ed., Wiley, 2023). Dr. Lu served as editor for the Canadian Journal of Electrical and Computer Engineering and associate editor for several journals including IEEE Transactions on Circuits and Systems I, IEEE Transactions on Circuits and Systems II, International Journal of Multidimensional Systems and Signal Processing, and Journal of Circuits, Systems, and Signal Processing.