Cross-Modal Federated TinyML for MCU-based Internet of Medical Things

Kainat Ibrar, Pietro Fusco, Gennaro Pio Rimoli, Francesco Palmieri, Massimo Ficco

***Abstract*— Internet of Medical Things (IoMT) and Machine Learning (ML) have become increasingly popular in healthcare. Wearable tiny medical devices can collect and transmit personal health-related data. However, applying ML-driven IoT in healthcare presents several challenges, especially in remote patient monitoring: *(i)* latency and privacy issues can hinder the transmission of sensitive medical data; *(ii)* it could require the correlation of multiple and heterogeneous data collected by different medical devices; and *(iii)* the limited resources of ultra-low-power wearable devices prevent the implementation of complex ML models directly on-board. On the other hand, Federated Learning (FL) and TinyML can address these limitations by enabling collaborative model training across distributed IoMT-edge resource-constrained microcontroller unit (MCU) based devices, allowing local data processing and improving latency and energy efficiency. However, traditional FL mainly handles unimodal data, limiting its direct applicability to several real-world IoT healthcare scenarios. This work proposes a Cross-Modal Federated TinyML (TinyCFL) implementation for MCU-based medical devices, which employs an intermediate multimodal distributed data fusion approach. Data from different modalities are independently processed on different tiny devices to extract features, which are then fused for healthcare tasks that require cross-modal reasoning. The proposed approach is evaluated by using the "UP-Fall detection" dataset, which is used in balanced, unbalanced, and Participant-Wise distribution scenarios. The proposed approach has been designed to be deployed in a distributed IoT-edge scenario. A prototype of the proposed TinyCFL approach based on resource-constrained MCUs has also been implemented.**

***Index Terms*— Cross-modal feature learning, Federated Learning, TinyML, IoMT, Microcontroller Unit.**

I. Introduction

In recent years, the Internet of Medical Things (IoMT) and Machine Learning (ML) have found wide applicability in many aspects of the healthcare sector, such as disease prevention and diagnosis [1]. Wearable medical devices have revolutionized healthcare by providing continuous monitoring and personalized insights [2]. However, several challenges should be faced in applying ML-driven IoT for healthcare [3]. Privacy concerns can preclude the transmission of sensitive medical personal data to remote servers for processing [4]. Moreover, remote patient monitoring could require the simultaneous use of multiple and heterogeneous information, such as vital signals and human behaviors [5]. Such information could be produced in different format ($e.g.$, images, sounds, text) and measured by different sensors ($e.g.$, glucose meters, blood pressure cuffs, micro-cameras, microphones), which may also belong to different IoT devices deployed in the home environment or worn by the patients ($e.g.$, smartwatches, webcams, and smartphones).

Federated Learning (FL) introduced by Google in 2016 offers a paradigm to training ML models collaboratively. Unlike traditional server-side ML methods that require sending sensitive data to a remote server or cloud, FL processes raw data on individual clients, fostering a privacy-centric environment [6]. However, the conventional FL paradigm primarily addresses scenarios in which each client processes a single data type, referred to as *unimodal data*. This assumption limits its applicability in complex real-world IoT-based healthcare environments, where data often originates from multiple sources and modalities. To overcome this limitation, *Multimodal Federated Learning* (MFL) has emerged as an advanced decentralized ML paradigm. *Multimodal learning* plays a key role by fusing information from various modalities, such as text, audio, images, video, and biomedical sensor readings, to enhance ML and robustness. In MFL settings, clients equipped with multiple sensors collect different types of data modalities simultaneously. These clients collaboratively train a global model without sharing raw data, enabling improved generalization and performance in heterogeneous data environments [7].

On the other hand, the inherent limitations of ultra-low-power IoT and wearable devices in terms of memory- and computational-constrains could hinder the processing of multimodal data on a single device, as well as the on-board deployment of complex ML models [8].

Tiny Machine Learning (TinyML) has emerged as a field dedicated to the design of ML models optimized for resource-constrained devices, including microcontroller units (MCUs) based wearables and IoT devices, also with less than 256 KB of SRAM [9], [10]. In this direction, *Tiny Federated Learning* (TinyFed) allows IoT and edge-devices with limited resources to collaboratively train tiny ML models while preserving data privacy and minimizing data transfer overhead. However, in TinyFed settings, all participating clients either possess unimodal or homogeneous data modalities, resulting in a shared and homogeneous feature space [11], [12]. The diversity of feature space consequently makes conventional parameter sharing and aggregation like FedAvg, ineffective.

K. Ibrar, P. Fusco, G. P. Rimoli, F. Palmieri, and M. Ficco are with the University of Salerno, Via Giovanni Paolo II, 132 - 84084 Fisciano (SA), e-mail: {kibrar, pfusco, grimoli, fpalmieri, mficco}@unisa.it.

Therefore, the objective should be to enable collaborative learning across heterogeneous modalities under strict IoT-device constraints [13]. As a result, the integration of FL, multimodal fusion, and TinyML could represent a viable framework for designing intelligent healthcare applications by MCU-based devices. However, the resource-constrains of the involved devices could impose the processing of a single data modality for each client [14]. This setting reflects a more realistic and challenging real-world IoT deployment, characterized by tiny devices with strict memory and computational constrains.

Early data fusion represents one of the common approaches to explore the correlation of different modalities [5], [15], [16]. Raw data sources are directly merged before being fed into the global model. There are also methods that use *late data fusion* to combine the decisions or probability independently predicted from different modalities, by using techniques such as voting and averaging [17], [18]. On the other hand, both approaches require simultaneous access to different modalities or sensors for multimodal fusion or selection, and thus, they cannot be directly applied to the considered IoT contexts characterized by strict resource-constrained devices [19]. In contrast, *intermediate data fusion* is the midway between two extremes. It allows the model to learn higher-level modality-specific features locally, and then, fused in a shared latent space. It does not require simultaneous access to different modalities at client-side. Thus, it can be more suitable to be implemented in MFL TinyML-based IoT-edge distributed environments.

This work proposes an *intermediate cross-modal FL data fusion* approach for resource-constrained medical devices. The approach can be hierarchically structured and customized to be deployed in a distributed IoT-edge scenario (while preserving data privacy). Each federated tiny IoT client acquires a single data modality and produces the modality-exclusive and modality-common features, which are used to learn the client-specific and shared models in the edge node. Then, the intermediate models are aggregated on the server-side to generate the corresponding global model. To evaluate the proposed approach, a realistic fall detection system has also been implemented as a case study.

The proposed framework jointly addresses intermediate *Cross-Modal Federated Learning* and *TinyML*-oriented (TinyCFL) deployment in IoMT-edge distributed settings, where each client operates on a distinct modality under strict microcontroller resource-constraints. The main contributions of this paper are as follows:

1) An intermediate TinyCFL framework for IoMT applications is presented, explicitly designed to operate under TinyML constraints and to handle heterogeneous feature spaces across MCU-based IoT clients;
2) A light-weight hierarchical IoT-edge architecture is designed to enable on-device training and MFL collaboration across distributed resource-constrained IoMT devices, overcoming the limitations of server-centric training;
3) The proposed TinyCFL is evaluated using the "UP-Fall detection" dataset [20], which is used in the balanced, unbalanced, and Participant-Wise distribution scenarios;
4) A realistic TinyCFL-driven fall-detection system implementation using generic MCU-based devices is produced, validating the practical feasibility, deployability and effectiveness of the proposed framework in a distributed IoT-edge architecture.

The rest of the paper is structured as follows: Background and state-of-the-art addressing-related problems are presented in Sec. II. Sec. III describes the proposed FL cross-modal model and the related IoMT-edge architecture. Sec. IV presents the considered case study and the experimental results. Comparison with others FL methods is presented in Sec. V. A prototype implementation of the TinyCFL approach on MCU-based devices in presented in Sec. VI. Conclusions and future work are summarized in Sec. VII.

II. Background and related work

A. Unimodal federated learning

FedAvg represents the first FL algorithm introduced to decentralize model training and improve data privacy. However, FedAvg suffers from significant performance degradation when training data are not independent and identically distributed ($i.e.$, non-IID data). To mitigate this issue, Zhao *et al.* [21] enhanced FedAvg by enabling limited data sharing among local clients. In [22], a layer-wise FL approach for deep learning models was proposed for matching and averaging hidden elements of neural layers with similar feature extraction signatures. FedRobust tackled device-dependent data heterogeneity by assuming that local data distributions are shifted from a common ground distribution through an affine transformation [23]. To enforce consistency between local and global representations, a contrastive learning at the model level has been proposed in [24]. Moreover, to alleviate feature shifts across local clients, a local batch normalization has been applied in [25]. Finally, Bercea *et al.* [26] proposed an FL approach designed for unsupervised brain pathology segmentation, disentangling the parameter space into shape and appearance components.

However, the proposed methods mainly focus on a single modality for each involved client, which makes it hard to satisfy the multimodality and the scalability of several IoT-based health applications. For this reason, multimodal learning has recently attracted a lot of attention, as it can offer complementary information across multiple modalities and outperform unimodal approaches [27].

B. Cross-modal federated learning

Cross-modal refers to the ability to realize data structures that allow different types of input ($e.g.$, images, audio, text, video, and sensor signals) to be fused and processed in a unified way. These representations enable ML-based systems to link knowledge across different modalities, $i.e.$, different forms of data and sources through which information is perceived and transmitted. For example, it enables the connection between an image or a vital signal to the corresponding text description. The goal is to create a common model in which data from different sources can be fused to perform tasks that require

reasoning across multiple modalities [14]. Cross-modal learning core principles include:

1) *Feature extraction*: Meaningful features are extracted from each modality.
2) *Fusion*: A unified representation is produced by combining features from different modalities.
3) *Alignment*: It ensures that corresponding elements are correctly matched. For example, aligning audio segments with corresponding video frames.
4) *Translation*: It consists in translating information from one o more modalities to another ($e.g.$, generating a textual description of an image). Hence also the term *cross-modalities*.

The multimodal fusion activity can conceptually occur at three different levels of a decision process: early fusion (combining raw data), intermediate fusion (combining the features extracted by each modality), and late fusion (combining the outputs of individual models) [16]:

- *Early fusion*: Multiple raw data sources are directly merged before being fed into a model. Steenkiste *et al.* [28] proposed backward shortcut connections for combining data from the heart rate, oxygen saturation and thoracic and abdominal respiratory belts into a single deep learning model to detect sleep apnea.
- *Intermediate fusion*: Data from different modalities is processed independently to extract meaningful features, which are then fused for task processing [17]. For example, Miao *et al.* [18] used an approach to estimate blood pressure exploiting features extracted separately from single-channel ECG and double-channel pressure-pulse wave (PPW) signals. A multi-instance regression algorithm was implemented to fuse such features.
- *Late fusion*: Independent models are trained on different data sources, and their predictions are aggregated using techniques such as voting, averaging, or weighted combinations [29]. For example, Wei *et al.* [30] used Support Vector Machine (SVM) classifiers for independent physiological signals, including ECG, GSR, RA, and EEG, and introduced a weighted decision-level fusion of the SVM classifiers for emotion recognition.

Moreover, at the data level, three different client modalities can be considered [31]:

- *Single modality*: Each client collects data of one modality. It uses this data to train its model. The server performs local model aggregation between the different modalities. Yang *et al.* [19] proposed a cross-modal federated human activity recognition. A feature-decomposed activity recognition network is used to aggregate local models trained on clients with different modalities.
- *Multiple modalities*: In this mode, each client collects data in the same multiple modalities and uses such multimodal data to train its model and perform model aggregation through the server. FedCMR [32] proposed federated cross-modal retrieval, in which each client holds both text and image data. First, the cross-modal retrieval model is trained by each client on local data and used to learn the common space of multiple modalities. Then, the common subspace of each client is aggregated and used to update the local model.
- *Heterogeneous modality*: It can be considered a compromise between the two previous methods. Each client can learn the feature representation of a single or multiple modes. Then, using the server aggregation model, they can realize the information interaction between the same modality or different modalities under different clients. Yu *et al.* [33] presented a framework that enables training models from IoT clients with heterogeneous modalities, data statistical heterogeneity, and tasks heterogeneity. They proposed a global-local cross-modal ensemble strategy to aggregate client representations and inter-modal and intra-modal contrasts to mitigate local model drift caused by the heterogeneous factors.

Recently, the application of cross-modal FL approaches in human activity recognition has attracted enormous interest in the scientific community. Qi *et al.* [5] have proposed MFL for fall detection. An early multimodal fusion approach is applied. Time series data from wearable sensors were coded into images using the Gramian Angular Field and then fused with the visual images collected by cameras. Each client performs local model training using local data, and then the server performs multimodal data fusion. The federated approach is based on the classical FedAvg algorithm. A similar approach is proposed by Zhao *et al.* [34]. In particular, FedIoT was a heterogeneous multimodal semi-supervised FedAvg-based FL framework for fall detection.

It trained local autoencoders to extract correlated representations from different modalities and a single modality by using different IoT clients. Moreover, for each client, two autoencoders are used for local training, $i.e.$, although the local autoencoder without the corresponding modal data is frozen, two models have to be uploaded by the client during each round of model aggregation. Each autoencoder is locally trained to extract correlated representations from different local data modalities. A multimodal FedAvg algorithm is used to aggregate the produced representations. FedMSplit [35] also faced federated training assuming heterogeneous setups of sensors in the clients. To support the modality incongruity, they split local models into several blocks. Each block type provides a specific view of client relationships. A dynamic and multi-view graph structure is used to represent the underlying correlations between clients and used to promote local model relations. A recent work proposed a modality-collaborative activity recognition network to learn a global activity classifier shared across all clients [36]. To generate modality-specific and modality-agnostic features, they learned an egocentric encoder and an altruistic encoder considering the constraint of a separation loss and an adversarial modality discriminator collaboratively learned. Moreover, they proposed an angular margin adjustment scheme to improve the modality discriminator on modality-imbalanced data. However, although several MFL approaches have been proposed in the literature, multimodal TinyML on resource-constrained MCUs is an open challenge.

C. Multimodal data fusion in TinyML

TinyML is a recent paradigm studied to offer solutions based on ML at the edge level. TinyML algorithms have been designed specifically to run on computational- and memory-constrained devices. The key advantages are related to the possibility of processing data directly at the edge without transmitting it to the cloud, thus improving privacy and scalability and reducing latency [37]. ML for tiny devices is developed using server-side, client-side, and hybrid approaches, depending on where the training process is performed:

- *Server-side*: Generally, the adopted ML model is trained, optimized, and compressed offline, $i.e.$, in remote servers or cloud, and then, it is deployed on the IoT devices. It allows training models with a large dataset and achieving high accuracy [38]–[40]. Several modern deep training frameworks adopt offline training, such as TensorFlow Lite, PyTorch Mobile, X-CUBE-AI from STM, and others. However, it is not suitable for a federated approach.
- *Client-side*: On-board training is an emerging approach to support continuous learning by training the model with on-field data, reduce concept drift of data, and data privacy violations [41], [42]. Profentzas *et al.* [43] presented MiniLearn, an open-source, on-device learning system for low-power IoT devices. It enables the re-training of deep neural networks (DNNs) on-device, allowing devices to re-train the pre-trained and quantized models using data collected after their deployment. Similarly, Wulfert *et al.* [44] presented an open-source library implementation for on-board training in IoT devices.
- *Hybrid approach*: Transfer learning (TL) is another technique used when the entire model does not need to be fully trained on the device, or it does not fit the resource capability. The neural network is pre-trained offline and then deployed onto the device, which refines the output. In this direction, Ren *et al.* [41] presented the first work that brings incremental online learning. However, the authors refine only the final layers in RAM and update their weights based on specific metrics. Cai *et al.* [45] propose a different approach. They analyzed the memory footprint of back-propagation, which is the stage of the training process where weights are updated based on certain metrics. The study reveals that the activation of intermediate layers was only necessary when weights were updated. Therefore, the idea was to freeze the weights and update only the biases of each layer to reduce the memory required for training. Ravaglia *et al.* [46] utilized a technique that combines TL and parallel ultra-low-power processing. This technique involves taking a few old data points from the original training set and encoding them into a low-dimensional 8-bit quantized latent space. The encoded data is then combined with new data for incremental learning tasks. Lin *et al.* [47] demonstrated the feasibility of on-board training of neural networks with less than 256-KB SRAM. However, they performed pre-training of the models and post-training quantization before running fine-tuning on MCUs.

Several open-source implementations proposed TinyML solutions for FL and TL for resource-constrained MCUs [11], [48], [49]. However, they focused on a single modality for each involved client. In this direction, optimised neural networks for TinyML applications that leverage multimodal data fusion have been presented in [50]. Nooruddin *et al.* [51] proposed a fusion architecture for human activity recognition, capturing various data modalities by resource-constrained wearable and ambient sensors, including accelerometer, gyroscope, magnetometer, and microphone. Kayarvizhy *et al.* [52] used a combination of DNN and long short-term memory models to develop a solution for posture detection on wearables. The model is trained on the server and then deployed on wearables by using TensorFlow Lite. Kalenberg *et at.*, [53] presented a multimodal sensor fusion navigation method for nano-UAVs, which uses a low-power grayscale camera in combination with a multipixel time-of-flight sensor. TinyM2Net-V2-a is a low-power architecture for multimodal DNNs to detect Covid-19 from multimodal cough, speech, and breathing audio recordings, using cyclic sparsification and hybrid quantization of weights/activations [54]. A dedicated FPGA hardware accelerator was designed. Gibbs *et al.* [55] deployed two DNNs onto an Arduino Nano 33 Sense MCU for context-aware stress recognition using features collected by the PPG sensor, including the raw EDA values, HR amplitude, and BPM and HRV features. However, the model was too large to be trained on the MCU, thus it was trained on the server-side.

Although multimodal data fusion has attracted increasing attention within the TinyML community, particularly in applications such as human activity recognition, its adoption on severely resource-constrained MCUs remains limited. Existing approaches primarily relied on server-side training followed by model deployment, which prevents direct applicability to FL settings where local training is essential [52], [55]. Other solutions achieved multimodal processing by leveraging specialized hardware accelerators, such as FPGAs or on-chip digital signal processors, thereby departing from the constraints of general-purpose MCUs typically targeted by TinyML [54], [55]. Moreover, current TinyML-oriented federated frameworks assume homogeneous data modalities across participating clients [11], [48], [49]. As a result, the joint problem of cross-modal client heterogeneity, on-device training, and federated collaboration under TinyML constraints remains largely unexplored. Within this context, our work represents the first unified framework that enables cross-modal FL in a distributed IoT-edge architecture, leveraging on-device training on resource-limited MCUs, which we denote as TinyCFL.

III. The FL cross-modal MCU-oriented model

The intermediate data fusion approach applied to address the presented cross-modal problem in the IoMT domain draws conceptual inspiration from [19]. However, the problem setting and objectives differ substantially. Specifically, Yang *et al.* [19] focused on the algorithmic design of multi-modal federated learning under the assumption of relatively capable devices, without accounting for the stringent memory, computation, and energy constraints imposed by MCUs. In contrast, our work targets a fundamentally different setting by enabling cross-modal federated learning under explicit TinyML constraints,

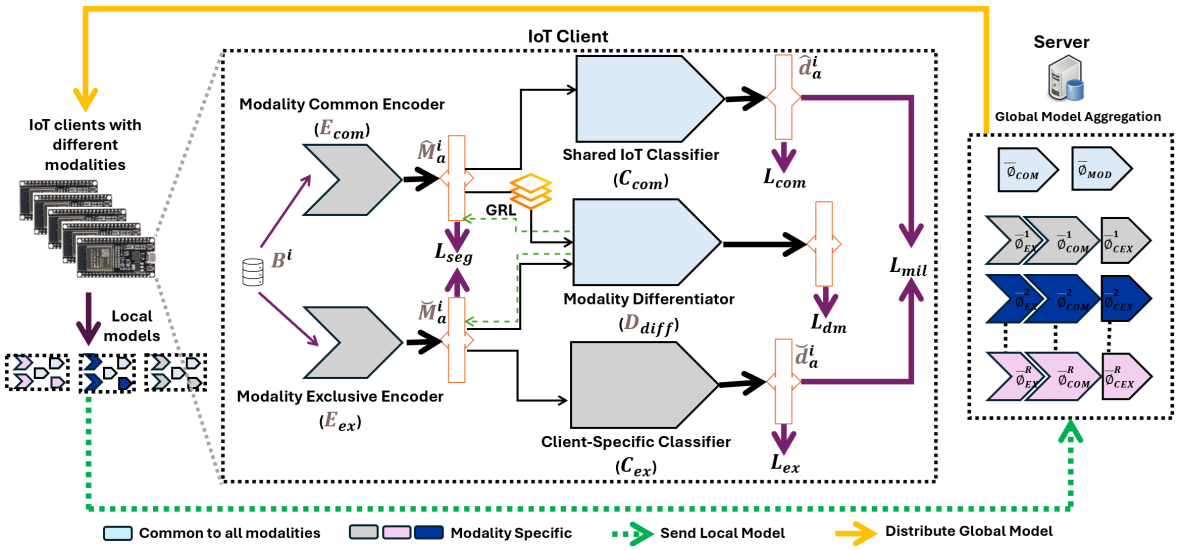

Fig. 1: *The federated cross-modal model.*

where models must be trained and updated directly on resource-constrained MCUs. Accordingly, we prioritize lightweight architectures and practical deployability in IoT-edge scenarios. These distinctions position our work as a system-level extension toward practical cross-modal federated learning in TinyML environments.

Fig. 1 gives an overview of the employed approach. It is divided into five consecutive key phases. The first four are performed at the client-side and the last at the server side, including: (1) the acquisition of multi-modality data, (2) the extraction of basic input features, (3) the production of modality-exclusive and modality-common features by using both an exclusive encoder and a common encoder for each client, (4) the desired latent representations (*i.e.*, modality-exclusive and modality-common) are used to learn the client-specific and shared classifiers, respectively. A modality differentiator is also added to facilitate the parameter learning of the modality-specific and modality-common encoders. In the final phase (5), the modality-based aggregation of local models is performed on the server-side to generate the corresponding global models. A detailed description of the proposed methodology is presented in the following section.

A. Client-side cross-model

As illustrated in Fig. 1, at the client-side, the cross-modal model includes five components for each IoMT client: the *modality-common encoder* (E_{com}), the *modality-exclusive encoder* (E_{ex}), the *shared IoT classifier* (C_{com}), the *modality differentiator* (D_{diff}), the *client-specific classifier* (C_{ex}). The functionalities of these components are described in the following paragraphs.

a) Modality-common and -exclusive encoders: In the considered model, the *modality-common* and *-exclusive* encoders are designed as 2-layer perceptrons in each IoMT client. These encoders are employed to capture distinct perspectives of the input data. The E_{com} encoder is responsible for generating a modality-common feature space for all involved IMoT devices. Instead, the E_{ex} encoder encodes data representations that emphasize modality-specific features. Each encoder is implemented by two layers: an input layer and a hidden layer linked to the output layer. The *Rectified Linear Unit* function (ReLU) is applied as an activation function. Mathematically, the connection between the input layer and the hidden layer is represented in Eq. (1):

$$\sigma(x) = ReLU(W_1 \cdot x + b_1). \tag{1}$$

Subsequently, the transformation of the hidden layer results in the encoder's output, as expressed in Eq. (2):

$$y = W_2 \cdot \sigma(x) + b_2, \tag{2}$$

where $W_* \in \mathbb{R}^{d \times d}$ represents the weight matrix with $d = 64$, $b \in \mathbb{R}^d$ is the bias vector and $x \in \mathbb{R}^d$ denotes the input vector, which may correspond to data from either the modality-common or -exclusive perspective.

b) Modality differentiator: The modality differentiator D_{diff} is employed to distinguish between modality-common and -exclusive features, which are generated by the encoders E_{com} and E_{ex}, respectively. To ensure consistency across all IoMT clients, D_{diff} is shared and designed to guide the encoders E_{com} and E_{ex} in embedding input instances into the desired feature subspace by adversarial guidance. The "Gradient Reversal Layer (GRL)" is inclined between the modality differentiator and the modality-common encoder to implement adversarial guidance. The modality classifier is utilized to implement D_{diff}, mapping the input features $y \in R^d$ from either the E_{com} or E_{ex} encoder to modality labels, as described in Eq. (3). To transform inputs into a 32-dimensional space, A single-layer ReLU-activated perceptron $g : \mathbb{R}^d \rightarrow \mathbb{R}^{\tilde{d}}$ is used, followed by a Softmax classification layer:

$$D_{diff}(y; \boldsymbol{\theta}_{diff}) = \text{Softmax}(W \cdot g(y)), \tag{3}$$

where the transformed output feature space dimensionality is $\tilde{d} = 32$. The transformed feature space is mapped in the output space of modality labels by The weight matrix $W \in \mathbb{R}^{\tilde{d} \times M}$.

c) IoMT-based detector: It comprises two classifiers. The C_{ex} classifier represented in Fig. 1 is client-specific. It is uniquely assigned to each client based on modality, as well as trained on parameters generated by the E_{ex} encoder. The C_{com} classifier in Fig. 1 serves as a shared classifier that is utilized across the involved clients and is trained on features generated by the E_{com} encoder. The two classifiers are designed as two-layer perceptrons with ReLU activations and a hidden layer of dimension 64.

1) Losses calculation:

a) Segregation loss: To ensure that the encoders E_{ex} and E_{com} generate distinct feature representations, segregation loss is employed, inspired by "domain separation networks" [56]. This loss function enforces orthogonality among the encoders' outputs, as formulated in Eq. 4:

$$L_{\text{seg}}(B^i, c_i) = \frac{1}{m_i^2} \left\| \hat{M}_i^\top \breve{M}_i \right\|_F^2, \tag{4}$$

where $\hat{M}_i \in \mathbb{R}^{d \times m_i}$ and $\breve{M}^i \in \mathbb{R}^{d \times m_i}$ denote matrices whose columns correspond to the output features of the encoders E_{com} and E_{ex}, respectively. The Frobenius norm is represented by $\| \cdot \|_F$.

b) Modality differentiator loss: This loss function is designed to ensure clustering of features of the same modality, as well as maintain separation among different modalities. This functionality is performed through the incorporation of additive angular loss [57]. The mathematical formulation of this loss is provided in Eq. (5):

$$L_{\text{am}}(\gamma, c) = -\log \frac{e^{\nu \cos(\psi_c + \xi)}}{e^{\nu \cos(\psi_c + \xi)} + \sum_{\substack{l=1 \\ l \neq c}}^{L} e^{\nu \cos(\psi_l)}}, \tag{5}$$

with $\psi_c = \arccos(W_c^\top \tilde{\gamma}), \quad \psi_l = \arccos(W_l^\top \tilde{\gamma})$. The variable c represents the ground-truth label of the modality and indicates the c-th and l-th column of the weight matrix W. Similarly, ξ and ν are defined as the margin and scaling factors, set to 0.5 and 45, respectively. The angular margin loss is computed separately for features produced by E_{ex} and E_{com}, using the notations L_{amx} and L_{amc}, respectively. Furthermore, the dispersion regularizer, inspired by [58], is employed to constrain D_{diff} to maintain cross-modality variations within an FL model. The modality differentiator loss, denoted as L_{dp}, is formulated as Eq. (6):

$$L_{\text{dp}} = \sum_{l=1}^{L} \sum_{l'=1, l' \neq l}^{L} \max\left\{0, \rho - \left(1 - W_l^\top W_{l'}\right)\right\}, \tag{6}$$

where ρ is a margin factor equal to 1.5. Finally, the modality differentiator overall loss is computed as $L_{dm} = L_{dp} + L_{amx} + L_{amc}$.

c) Modality-aware global-local adjustment: The shared IoMT classifier, C_{com}, maintains stable parameter learning even with modality-imbalanced data, as it is trained by using all clients. However, the client-specific classifier, C_{ex}, is more susceptible to instability due to modality imbalances. Therefore, a modality-aware global-local adjustment scheme has been used to enhance local client's discriminative capability by exploiting C_{com}. The *Kullback-Leibler divergence* [59] is used to align class-level relationships between C_{com} and C_{ex}, as formulated in Eq. (7):

$$L_{\text{mil}}(\mathcal{B}^i, c_i) = -\sum_{a,b=1}^{N} \omega(W_{\text{ex}}^{c_i,a}, W_{\text{ex}}^{c_i,b}) \log\left(\frac{\gamma(W_{\text{com}}^a, W_{\text{com}}^b)}{\gamma(W_{\text{ex}}^{c_i,a}, W_{\text{ex}}^{c_i,b})}\right). \tag{7}$$

W_{com}^a and $W_{\text{ex}}^{c_i,a}$ represent the final layer parameters of C_{com} and C_{ex} for the a-th class, where c_i denotes the client's modality label. The function ω quantifies class similarity as a normalized cosine distance within the $[0, 1]$ range.

d) Fall detection loss: The classification loss for C_{com} and C_{ex} is computed using the *cross-entropy loss*, defined in Eq. (8) and (9) as follows:

$$L_{\text{com}}(\mathcal{B}^i, z_i) = -\frac{1}{m_i} \sum_{a=1}^{m_i} d_a^i \log \hat{d}_a^i, \tag{8}$$

$$L_{\text{ex}}(\mathcal{B}^i, z_i) = -\frac{1}{m_i} \sum_{a=1}^{m_i} d_a^i \log \breve{d}_a^i, \tag{9}$$

where $\hat{d}_a^i$ and $\breve{d}_a^i$ are the predicted probabilities from C_{com} and C_{ex}, respectively, and d_a^i denotes the ground truth label.

B. Server-side global model aggregation

a) Federated learning: The proposed approach is deployed within a FL framework. In this setup, each client independently trains a local model on its private data for L epochs. After training, clients share their model parameters with a central server, aggregating them to construct an updated global model. The refined global model is then redistributed to the clients. This iterative process continues for G rounds until the model converges to an optimal solution.

b) Global model production: As previously explained, the approach comprises both private modules ($i.e.$, E_{ex}, E_{com}, and C_{com}) and shared modules ($i.e.$, D_{diff} and C_{com}). A modality-aware aggregation method is employed on the server side, enabling the aggregation of local parameters from involved IoMT clients to produce the shared modules. In contrast, a modality-based aggregation is applied to the local parameters of clients to produce private modules. In Eq. (10), the implementation of the FedAvgaggregation scheme of the common classifier (C_{com}) and the modality differentiator (D_{diff}) is presented as follows:

$$\overline{\phi}_{\text{COM}} = \sum_{i=1}^{K} \frac{n_i}{n} \phi_{\text{COM}}^i, \text{ and } \overline{\phi}_{\text{MOD}} = \sum_{i=1}^{K} \frac{n_i}{n} \phi_{\text{MOD}}^i. \tag{10}$$

As shown in Eq. (11), only the local parameters shared by the clients with the same data modality are used to process the modality-common (E_{com}) encoder, modality-exclusive (E_{ex}) and the and client exclusive (C_{com}) classifier.

$$\overline{\mathbf{\Pi}}_{\text{MS}}^n = \sum_{i \in \Omega_n} \frac{m_i}{\sum_{i' \in \Omega_n} m_{i'}} \mathbf{\Pi}_{\text{MS}}^i, \quad n = 1, \ldots, N, \tag{11}$$

where Ω_n is the set of clients that share the same data modality. The loss L_T included in the local adversarial update is defined as $L_{\text{t}} = L_{\text{ex}} + L_{\text{com}} + L_{\text{mil}}$ and $L_T = L_{\text{t}} + \gamma_1 L_{\text{seg}} + \gamma_2 (L_{\text{amx}} - L_{\text{amc}})$. The balance weights γ_1 and γ_2 are set to 0.6 and 0.4, respectively.

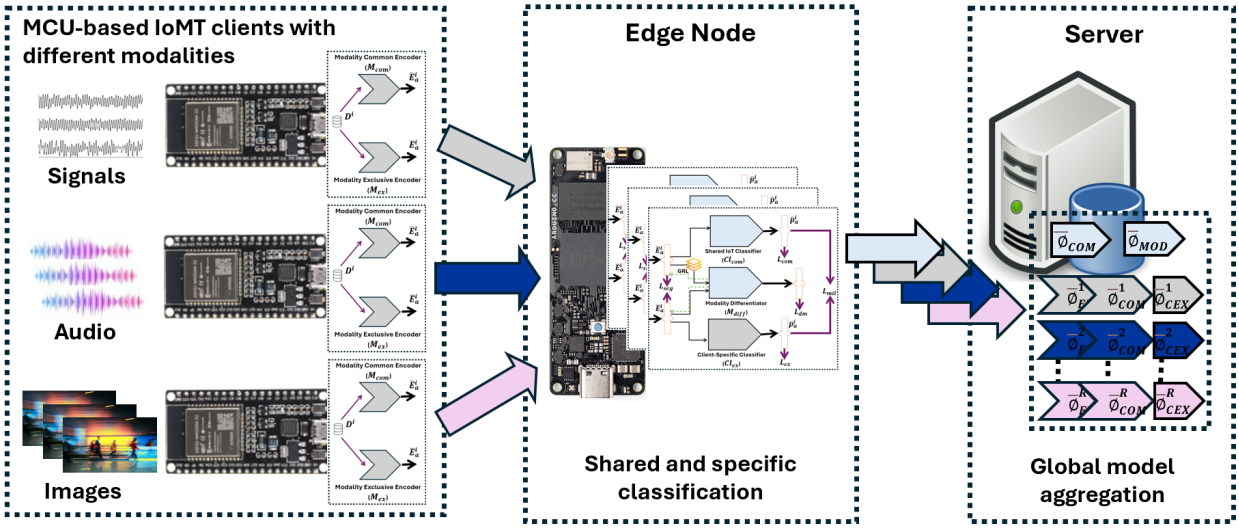

Fig. 2: *The distributed architecture of the FL cross-modal in the IoMT-edge scenario.*

C. IoMT-edge distributed architecture

The implementation of the presented cross-modal model in resource-constrained MCU-based devices poses several challenges. Specifically, their limited memory capacity represents a significant constraint on the processing of the encoders and classifiers necessary for model deployment.

Therefore, the model has been designed to be deployed according to the architecture illustrated in Fig. 2, where the client-side components have been distributed hierarchically between the IoMT clients and the edge nodes (while preserving data privacy). In particular, the initial layer of the presented architecture, encompassing both the modality-exclusive and modality-common encoders, is implemented on the IoMT clients, preserving the data privacy. The remaining components are deployed on the more powerful edge nodes. Each IoMT client gathers single-modal data and extracts both modality-exclusive and modality-common features. These extracted features are then transmitted to the edge node, via various wireless technologies, such as Bluetooth and WiFi. The edge node processes these received features to learn a client-specific classifier, a shared classifier, and a modality differentiator for each connected client. Furthermore, the edge node facilitates connection to high-bandwidth networks, enabling communication with the server for modality-based aggregation of local models, ultimately leading to the generation of the corresponding federated global models.

IV. Experimental Setup and Results Analysis

A. Procurement of multimodal data

A cross-model FL-based fall detection system has been considered a case study to evaluate the proposed approach. To induce cross-model heterogeneity and optimize classification accuracy, we incorporate techniques established in the literature [60], [61]. In particular, we utilized the ”UP-Fall detection” dataset [20], a publicly accessible resource encompassing 11 different activities, each repeated across three trials. The dataset includes recordings of six basic daily actions, such as sitting, walking, jumping, picking up an object, standing, and lying down, alongside five types of fall scenarios: forward falls using hands or knees, backward falls, sideward falls, and falls while attempting to sit on an empty chair. Data were gathered using a multimodal setup from 17 healthy young participants. This dataset integrates sensor and image modalities, utilizing wearable and ambient sensors for sensorial data in conjunction with visual data acquired via frontal and lateral cameras, supporting a multi-perspective analysis. This multi-modal configuration of the dataset makes it an ideal choice for cross-modal reasoning tasks, as it provides synchronized data of same physical activities captured from multiple perspectives through sensorial and visual sources. For the sensorial modality, we utilized both temporal and frequency domain features obtained from all devices. In contrast, for the imaging modality, we derived visual representations capturing the pixel-wise displacement across successive frames, calculated using an optical flow method (as detailed in Sec. IV-A.1).

1) Basic input features extraction:

a) Sensor devices: Following the acquisition of multimodal data, the subsequent phase involves feature extraction, according to the methodology presented by Martínez-Villaseñor et al. [20]. In particular, the authors generated three separate feature sets based on different window sizes: (a) one-second, (b) two-second, and (c) three-second intervals, each incorporating a 50% overlap. Within the scope of wearable sensors and ambient devices, twelve temporal and six frequency domain features were extracted across each window. Consequently, each feature file comprised window samples, beginning with a timestamp column, followed by 756 columns representing 18 features derived from 42 distinct sensor signals, along with 3 supplementary columns indicating the activity type, subject identifier, and trial number. In this study, we evaluated the fall detection framework utilizing the feature set constructed with: a one-second window size and a 50% overlap.

b) Vision devices: In contrast to the averaging strategy employed in [20], our image feature extraction method preserves temporal granularity by retaining individual feature vectors for each window per camera. Moreover, [20] enables the estimation of apparent object displacements across image sequences

typically associated with brightness variations and used to infer pixel correspondences between successive frames. We modified the final aggregation step to better capture temporal dynamics. For this dataset, the "Horn and Schunck optical flow" algorithm was applied [62]. Feature extraction was conducted as follows. For both cameras, image features within a one-second window were obtained, representing the relative horizontal (x) and vertical (y) displacements. These components were recorded as numerical matrices with dimensions matching the original images. To facilitate interpretability, the two matrices were combined to compute the magnitude of displacement, resulting in matrix M, defined as: $M_{i,j} = \sqrt{x_{i,j}^2 + y_{i,j}^2}$. Moreover, to reduce computational complexity, matrix M was downsampled from 640×480 to 20×20 pixels and subsequently reshaped into a 400-dimensional row vector for each camera. Unlike the methodology in [20], we did not average these vectors over the window. Instead, one distinct 400-element vector was maintained per window per camera, preserving finer temporal resolution. In total, 800 features were extracted from the two cameras and 756 features from wearable and ambient sensors.

2) Experimental results: The considered dataset is structured in CSV files, which contain sensor-derived data and ZIP files (that hold camera-captured images). Data is organized into 17 primary folders, each corresponding to an individual participant. Within each participant's folder are 11 subfolders representing the various activities. Each of these activity folders contains three additional subfolders, each corresponding to a distinct trial. Within these, a single CSV file stores the pre-processed sensor data for the specific trial, and two ZIP files contain the visual recordings from the two cameras, one per device. The finalized integrated dataset comprises 296,364 instances of unprocessed sensor and images data, acquired at an approximate frequency of 18.4 Hz and amounting to roughly 812 GB of digital storage.

In addition to modality heterogeneity, the datasets also exhibit client-level variability in sample size and label distribution, further reinforcing the non-IID nature of the experimental setup.

Since this study focuses on the task of fall detection, we categorize the activities into two classes: *Fall* and *Not-Fall*, assigning them labels 1 and 0, respectively. The multimodal dataset is distributed across four clients within a FL environment, with each client exclusively holding data from a specific modality. For each modality, we randomly select two clients and evaluate the proposed framework under three real-world scenarios, considering both balanced and imbalanced data distributions. The proposed model is tested using PyTorch, with training conducted via the SGD optimizer. The optimizer is configured with a weight decay of 1×10^{-5} and a momentum of 0.9. Clients processing sensorial activity data are trained using a learning rate of 0.0005, while those handling imaging data adopt a rate of 0.125. Each client undergoes training for two local epochs per communication round using a batch size of 32, across a total of 150 FL rounds. Data on each client is randomly divided in a 75:25 ratio for training and testing respectively.

a) Scenario 1 - Balanced dataset distribution: This experimental setup for the balanced dataset distribution is designed in a manner in which the dataset is uniformly distributed among four clients, with each client receiving 4,590 instances comprising 2,295 samples per class. Consequently, a cross-modal IoMT-based classification task is formulated with four clients, two classes, and two modalities, totaling 18,360 instances. To verify the performance of our approach, we employ widely used evaluation metrics for classification tasks. Tab. I presents detailed results from our cross-modal IoMT-based fall detection system. Metrics, including accuracy, sensitivity, specificity, precision, F1-score, and AUC, are calculated for each client per round and subsequently averaged per modality. This process is conducted across 150 FL rounds, with final results representing the mean of each metric. Our findings confirm the method's efficacy: sensor modality clients reach 98.95% accuracy and 99.92% AUC, whereas imaging modality clients achieve 96.73% accuracy and 99.25% AUC at round 150. Fig. 3 and Fig. 4 present the key performance trends of the sensor and image modalities in an FL framework throughout 150 training rounds. The evaluation includes four critical metrics: Accuracy, F1-score, AUC, and average shared loss, each plotted as a function of the FL rounds.

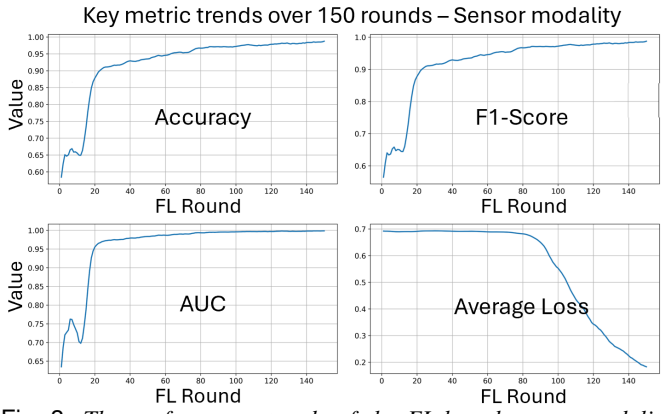

Fig. 3*: The performance trends of the FL-based sensor modality during 150 training rounds for balanced data distribution.*

The visualized trends in Fig. 3 across 150 FL rounds demonstrate a consistent and substantial improvement in model performance on sensor modality data. The accuracy metric demonstrates a steep ascent within the initial 20 rounds, surpassing 0.9 and progressively stabilizing around 0.98, indicating rapid convergence and strong generalization capacity. Similarly, the F1 score follows a comparable trajectory, exceeding 0.9 early in training and maintaining stability thereafter, reflecting the model's ability to effectively balance false positives and false negatives, an especially critical factor in sensor-based monitoring where misclassifications can compromise system reliability. The AUC curve reinforces these observations, showing a swift progression towards 1.0 within the early rounds and maintaining a consistently high value, thereby affirming the model's excellent discrimination power in distinguishing event classes. In parallel, the average shared loss remains relatively constant in the earlier rounds, but undergoes a marked decline after round 90, eventually dropping below 0.2. This downward trend signifies enhanced optimization across decentralized

TABLE I*: Performance analysis of cross-modal IoMT-based fall detection, averaged across clients and modalities.*

Metrics Frequency	Modality	Accuracy	Sensitivity	Specificity	Precision	F1-Score	AUC
Scenario 1: Balanced Dataset Distribution							
At Round 150	Sensor	98.95%	98.95%	98.83%	98.96%	98.95%	99.92%
	Image	96.73%	96.73%	94.63%	96.86%	96.73%	99.25%
Mean Metrics after 150 rounds	Sensor	92.88%	92.88%	93.00%	93.04%	92.81%	96.54%
	Image	88.83%	88.83%	85.00%	89.86%	88.01%	93.67%
Scenario 2: Imbalanced Dataset Distribution							
At Round 150	Sensor	99.27%	99.27%	100.0%	99.28%	99.26%	100.0%
	Image	85.63%	85.63%	100.0%	73.35%	79.01%	90.23%
Mean Metrics after 150 rounds	Sensor	93.20%	93.20%	100.0%	88.68%	90.56%	92.63%
	Image	87.07%	87.07%	99.17%	81.62%	82.50%	86.72%
Scenario 3: Primary Participant-Wise Dataset Distribution							
At Round 150	Sensor	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
	Image	95.01%	95.01%	99.02%	95.05%	94.54%	98.49%
Mean Metrics after 150 rounds	Sensor	93.42%	93.42%	100.0%	89.46%	90.98%	93.60%
	Image	87.55%	87.55%	98.73%	80.39%	82.99%	70.62%

clients and aligns with the observed improvements in the other performance metrics. Fig. 4 illustrates the performance

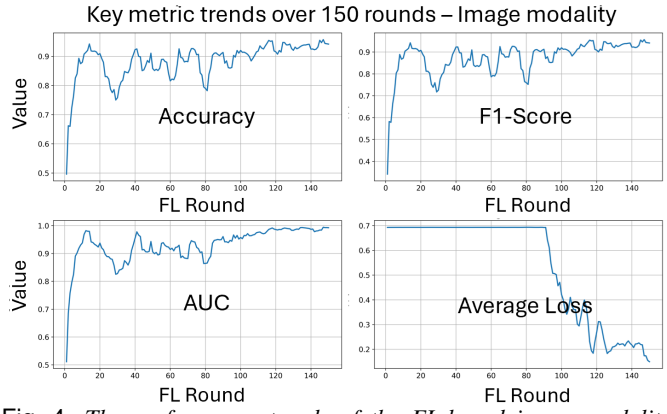

Fig. 4*: The performance trends of the FL-based image modality during 150 training rounds for a balanced data distribution.*

trajectory of an image-based modality within an FL framework across 150 training rounds, assessed through four principal metrics: Accuracy, F1-score, AUC, and average shared loss. The accuracy metric rapidly rises in the early training stages, surpassing 0.9 within the first 20 rounds, and eventually stabilizes despite minor fluctuations. This indicates effective convergence and robust generalization capabilities. The F1-score mirrors this trend, reaching a stable value above 0.9 around round 80, signifying the model's ability to balance precision and recall, an essential requirement in fall detection systems where both false positives and missed events carry significant consequences. The AUC curve further supports these findings, showing a sharp increase toward 1.0 early in training, highlighting the model's strong discriminative capacity in differentiating between fall and non-fall events. Although minor variations are observed, the AUC remains consistently high throughout. Concurrently, the average shared loss exhibits a steady decline, with a notable drop post-round 90, ultimately falling below 0.2. This reduction corresponds with enhanced performance in the other metrics, suggesting efficient error minimization across federated nodes.

Collectively, these trends highlight the effectiveness of leveraging FL for sensor and image-based data modalities within IoMT systems, demonstrating its ability to deliver precise outcomes under decentralized and potentially heterogeneous conditions. Although the imaging modality underperforms relative to sensor data, the observed improvements across rounds indicate the framework's capacity to glean useful information even from less-informative sources.

b) Scenario 2 - Imbalanced dataset distribution: The second experiment investigates the framework's performance with unbalanced data, characterized by disproportionate allocations of modality-specific samples. Clients handling sensory modality data are given 4,590 instances, while clients 3 and 4 handling imaging modality data are given reduced allocations of 3,000 and 2,000 instances, respectively. Tab. I provides a comprehensive summary of performance metrics from the cross-modal IoMT-based fall detection framework. This procedure spans all 150 FL rounds, with the final reported values reflecting the mean performance after training for all 150 rounds. Despite slightly lower performance compared to the balanced dataset, the proposed methodology demonstrates robust effectiveness under imbalanced conditions, with sensor clients achieving 93.20% accuracy and 92.63% AUC, and image clients reaching 87.07% accuracy and 86.72% AUC, based on the mean values computed over 150 rounds. These results confirm the framework's resilience and adaptability to data heterogeneity across modalities.

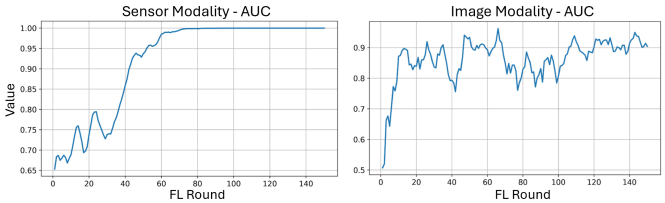

Fig. 5*: Comparative analysis of AUC trends between the sensor and image modalities over 150 rounds in Scenario 2.*

Fig. 5 presents an AUC progression over 150 FL rounds using diverse modality data. The sensor modality demonstrates a rapid and stable improvement in AUC, reaching near-perfect performance after approximately 70 rounds. In contrast, the image-based model displays significant variability throughout the training process, with noticeable dips and peaks. Despite these oscillations, it maintains a generally high performance, stabilizing around 0.90 in the later rounds. This inconsistency may stem from the inherent complexity and higher variability in image data, which can introduce noise and increase the difficulty of model convergence, especially in a FL context with heterogeneous clients. The image modality also achieves commendable results, underscoring the adaptability and robustness of the proposed approach across diverse data types.

c) Scenario 3 - Primary Participant-Wise dataset distribution: This scenario extends the previous setup by assigning participant-specific data to four clients via random selection of major folders within the consolidated dataset, ensuring distinct allocation of sensorial and imaging modality data. The differing durations of recorded activities inherently lead to imbalanced sample distributions. A total of 3,828 instances were assigned to clients dealing with sensor modality data, in contrast to the higher allocations of 17,899 and 16,690 instances received by clients 3 and 4, who managed imaging modality data. As illustrated in Tab. I, Scenario 3 examines the Primary Participant-Wise dataset distribution, where the sensor modality demonstrates exceptional and consistent performance, achieving perfect scores, i-e 100% across all evaluation metrics at round 150 and maintaining high accuracy and robustness over 150 rounds. In contrast, the image modality begins with comparatively lower performance, reflected in its reduced mean values ($e.g.$, 87.55% accuracy and 70.62% AUC). However, it demonstrates substantial improvement by round 150, attaining 95.01% accuracy and 98.49% AUC. This trend indicates that model performance for the image modality significantly improves with increased communication rounds and global model aggregation through FL.

V. Comparison results and ablation study

To evaluate the scalability and performance of the proposed TinyCFL with more than two data modalities, the `Epic-Kitchens` dataset was considered [63]. It is a large-scale multimodal benchmark for egocentric human activity recognition. Experiments are conducted on the Epic-Kitchens-100 subset, which contains nearly 90k video segments of human-object interactions from 37 participants. A subset of participants also provides audio and sensor data. In the context of this work, 97 verbal activity labels are used across four modalities: sensor data, video, optical flow, and audio. The benchmark is constructed by assigning each modality to independent clients, resulting in 16 clients, 97 activity classes, and 34,018 total instances.

We compare our method to state-of-the-art FL approaches, including: (1) *FedAvg* learns a single global model by averaging client updates [64]; (2) *pFedMe* introduces personalization by decoupling local and global optimization via the Moreau envelope [65]; (3) *LG-FedAvg* improves FedAvg through local representation learning [66]; (4) *PerAvg* based on meta-learning framework [67], learns an initial model that enables fast client adaptation [68]; (5) *FedBN* handles variations in feature distribution while maintaining local batch normalization [69]; and (6) *FedRep* separates shared representations from client-specific headers [70]. Moreover, as baseline (named *SingleMod*), we consider an approach in which each client independently trains a local model without FL, using only our approach's egocentric encoder and private classifier. Tab. II represents the average accuracy for clients sharing the same modality, assuming one-modality-one-client. Experimental results show that the TinyCFL model can successfully reduce the differences between different data modalities.

TABLE II: *Comparison with others FL methods.*

Methods	Data modality			
	Video	Flow	Audio	Sensor
SingleMod	25.42	34.06	38.08	25.60
FedAvg	25.98	31.25	36.32	24.57
pFedMe	22.50	26.97	30.63	22.92
LG-FedAvg	26.08	35.06	34.21	25.93
PerAvg	26.31	34.50	36.27	25.17
FedBN	27.35	33.13	38.21	27.94
FedRep	27.14	34.76	36.65	28.06
TinyCFL	**29.41**	**38.00**	**41.88**	**29.54**

Tab. III presents an ablation analysis of the proposed framework under the balanced dataset configurations. This study aims to highlight the contribution of each architectural module and loss component to the overall performance. Each variant selectively removes a specific component while keeping the remaining architecture and training strategy unchanged to enable an isolated assessment of its impact on cross-modal learning.

We first analyse the role of the classification components. In our variant V1, we remove the shared classifier C_{com} while keeping the remaining architecture intact including both the modality exclusive E_{ex}, and modality common E_{com} encoders. This configuration allows us to learn disentangled representations as in the full model and to isolate the contribution of the shared latent space while preserving the feature disentanglement mechanism. However, classification is performed exclusively using the private activity classifier C_{ex}, which leads to the most severe performance degradation across both sensor and image modalities. This result highlights the critical role of C_{com} in capturing modality-invariant discriminative patterns shared among heterogeneous clients. In contrast, eliminating the private classifier C_{ex} in V2 results in a smaller performance drop, indicating that modality-specific decision boundaries provide complementary information but cannot replace the shared classification space.

Next, we investigate the contribution of the encoder structure. Variant V3 removes the common encoder E_{com} together with the shared classifier C_{com}, thereby restricting learning to the modality exclusive encoder E_{ex} and private classifier C_{ex}. As a result, cross-modal knowledge sharing is disabled, and each modality is learned independently without a common latent representation. This configuration causes a substantial decline in performance, confirming that shared representations are essential for effective collaboration across heterogeneous

TABLE III*: Ablation study under the balanced dataset distribution. Results are reported as F1-score (%) averaged across clients*

Variant	Architectural components					Loss components			Data modality	
	C_{com}	C_{ex}	E_{com}	E_{ex}	D_{diff}	L_{seg}	L_{dp}	L_{mil}	**Sensor**	**Image**
V1: w/o C_{com}	×	✓	✓	✓	✓	✓	✓	×	66.40	65.20
V2: w/o C_{ex}	✓	×	✓	✓	✓	✓	✓	×	80.20	78.70
V3: w/o E_{com}	×	✓	×	✓	✓	×	✓	×	72.30	70.40
V4: w/o E_{ex}	✓	×	✓	×	✓	×	✓	×	74.00	71.90
V5: w/o D_{diff}	✓	✓	✓	✓	×	✓	×	×	70.10	68.30
V6: w/o L_{seg}	✓	✓	✓	✓	✓	×	✓	×	86.20	81.90
V7: w/o L_{dp}	✓	✓	✓	✓	✓	✓	×	×	84.10	79.50
V8: Baseline (no L_{mil})	✓	✓	✓	✓	✓	✓	✓	×	88.90	84.70
TinyCFL	✓	✓	✓	✓	✓	✓	✓	✓	**92.81**	**88.01**

modalities. Similarly, by eliminating the exclusive encoder E_{ex} along with private classifier C_{ex} as depicted in V4, we rely only on the modality common encoder E_{com} in combination with the modality differentiator D_{diff} to learn modality common features. These shared representations are then used to train the shared activity classifier C_{com}, allowing us to assess the effectiveness of purely common features without modality-exclusive modelling. This variant also degrades performance, though to a slightly lesser extent, suggesting that modality-specific encoders remain necessary to preserve information unique to each sensing modality.

The importance of explicit modality differentiation is further demonstrated by variant V5, where the modality differentiator D_{diff} is removed. This setting results in performance drop of 22.71% in sensor modality and 19.71% in image modality indicating that enforcing modality-aware discrimination is crucial for preventing feature collapse and for learning robust transferable representations in cross-modal federated settings.

We further examine the effect of individual loss components. Removing the segregation loss L_{seg} (V6) leads to a moderate performance decrease, showing its role in stabilizing feature disentanglement during training. In variant V7, when the modality discrimination loss L_{dp} is excluded, the modality differentiator loss L_{dm} is provided only through the angular-margin-based terms L_{ama} and L_{ams}. It enables us to evaluate the explicit role of L_{dp} in enforcing separation across modalities. This configuration exhibits an overall performance degradation of 8.5-8.7%, underscoring the importance of explicitly constraining modality separation in the latent space. In V8, where the model is trained without the modality-aware global-local adjustment L_{mil}, performance remains relatively high but consistently inferior to the full model, indicating that L_{mil} provides additional regularization that enhances cross-modal alignment.

VI. Implementation in a Realistic Scenario

In this section, a prototypal implementation of the proposed TinyCFL approach has been presented. It represents a realistic MCU-based fall-detection system that exploits two modalities: images and sensory data ($i.e.$, angular velocity and acceleration along three axes).

According to the distributed IoT-edge architecture presented in Sec. III-C, a different IoMT device has been implemented for each modality. Each IoMT device is composed by a dedicated smart sensor connected via Bluetooth low energy to an ESP32 S3 board (a dual-core Xtensa LX7, running at up to 240 MHz, with 512 KB of internal SRAM).

The smart sensors are dedicated to the acquisition of multi-modal raw data, as well as the extraction of basic input features to be forwarded to the ESP32. In the considered case study, the smart sensors are the ESP32-CAM and the Arduino Nano 33 BLE sense, used to collect images and movement measures, respectively. In particular, the ESP32-CAM includes an SVGA camera ($i.e.$, OV2640 2MP module) and a microSD card slot. To address memory and computaion constraints, the collected images are first saved on the microSD of ESP32-CAM and then reduced to 20×20 pixels before being transmitted to the ESP32. The Arduino Nano 33 is equipped with an ARM Cortex-M0 32-bit, a low-power chipset operating in the 2.4GHz range, and a 6-axis IMU, which makes this board suitable for fall detection alarm systems. Fig. 6 shows a prototype of the implemented sensor, which we assume is placed on the belt of the patient.

Fig. 6*: NANO 33 based motion sensor.*

Using the internal 3D accelerometer and gyroscope, the motion sensor produces the acceleration magnitude (AM), acceleration cubic-product-root magnitude (ACM), and angular velocity cubic-product-root magnitude (AVCM) measurements, as reported below:

$$AM(i) = \sqrt{\alpha_x^2(i) + \alpha_y^2(i) + \alpha_z^2(i)}, \tag{12}$$

$$ACM(i) = \sqrt[3]{|\alpha_x(i) + \alpha_y(i) + \alpha_z(i)|}, \tag{13}$$

$$AVCM(i) = \sqrt[3]{|\omega_x(i) + \omega_y(i) + \omega_z(i)|}, \tag{14}$$

where α and ω are acceleration and angular velocity, respectively.

Each ESP32 processes an exclusive encoder and a common encoder, which produce modality-exclusive and modality-common features respectively, based on the received input features. The encoders consist of two dense layers, each with approximately 4020 parameters, designed to capture two different viewpoints of the input data. Each ESP32 is connected to the edge node via WiFi.

The edge node is implemented by Arduino Portenta X8 with 2GB of DDR4. It is responsible for processing and learn the modality differentiator, client-specific, and shared classifiers. A desktop PC (Intel i5 2400 with 4 GB of RAM) at the server-side is used to perform the global model aggregation from all clients and used to generate the corresponding federated global models. The communication between IoMT devices and the edge node, as well as between the edge and the server is based on a publish-subscribe communication paradigm based on MQTT and implemented via a Mosquitto broker, which supports TLS/SSL encryption [71]. The `PubSubClient.h` and `WiFiClientSecure.h` C libraries were used to implement SSL connections on the ESP32.

Moreover, to enable efficient execution of on-device training directly on MCUs, we leveraged a C library offered by the AIfES framework [44], whereas the FL paradigm has been developed by using our previous implementation, which enhances TinyML in a distributed architecture [11].

To evaluate the presented TinyCFL implementation in the real MCU-based scenario, we adopt various metrics at client-side, including: the *current*, *power*, *execution-time*, and *number of operations* processed by the ESP32 during the training process. For each FL round, the local training was conducted over 10 epochs with a batch size of 32. The total number of multiplications executed during the training is 2.63E+10. Tab. IV presets the achieved measurements during each training epoch compared with baseline (*i.e.*, *SingleMod* in Sec. V).

TABLE IV: *Client-side performance measures during training.*

Training	*Current (mAh)*	*Power (mWh)*	*Execution time (s)*
Mean (Baseline)	56.24	280.26	2580.24
Variance (Baseline)	3.63E-02	9.24E-01	1.61E-03
Mean (TinyCFL)	85.09	425.67	4078.56
Variance (TinyCFL)	1.39E-01	3.19E+00	2.81E-01

During the measurements, the ESP32 was powered by an external supply. An Arduino UNO, fitted with an INA219 current-voltage sensor, is utilized to track ESP32 power usage. The two devices exchanged a synchronization signal: when the ESP32 initiated its training process, it prompted the UNO to start measuring and recording the power delivery routine. Each device saved data to its own SD card. Fig. 7 shows the setup used for gathering consumption data. The overall energy consumption during the training process was about 2,050 mW. However, the energy consumption related to the transmission of the model parameters (*i.e.*, Modality Common and Modality Exclusive) via WiFi to the edge node at the end of each FL round must also be added. In the scenario considered, the transmission of the model parameters requires on average 0.250 milliseconds with a power consumption of about 660 mW. Finally, the energy consumption during each training process must be further multiplied by the number of FL rounds.

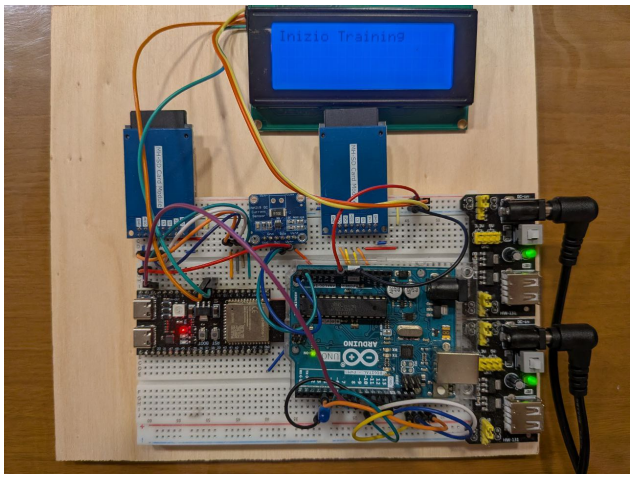

Fig. 7: *The setup used for gathering consumption data.*

Regarding the inference phase, compared to the other MFL fall-detectors for IoT presented in recent works, our solution use only two sensors (a camera and a wearable sensor). Instead, 8 clients are required in [5], including 5 wearable sensors placed on different parts of the body and 2 cameras placed in front and side of the experimental field. Depth camera videos, RGB camera videos, and sensory data (such as accelerometers, gyroscopes, and magnetic sensors) are used in [34]. 18 clients are used in [20], including 6 wearable sensors, 6 pairs of infrared sensors, and 2 cameras. However, they are tested by simulation environment and Pytorch framework. Our approach is the only one implemented on MCU-based devices.

Finally, as shown in Sec. VI, the proposed TinyCFL model is inherently designed to support scalability. It can accommodate additional clients and modalities, as it leverages modality-common representations al client-side, while preserving modality-specific features at the edge nodes. Consequently, when a new modality is introduced, even if the underlying feature space or training model differs, the system can seamlessly adapt through the modality differentiator, modality common and modality-exclusive encoder structure. Moreover, for global model construction, we introduce a modality-aware aggregation strategy in which parameters associated with shared modules are aggregated globally, while modality-specific parameters are aggregated in a modality-aware manner. This design enables effective collaboration across heterogeneous modalities and ensures robustness to cross-modal variability, thereby allowing the proposed TinyCFL framework to scale beyond UP-Fall dataset to other diverse multi-sensor healthcare datasets. On the other hand, from an architectural point of view, the Portenta X8 represents a bottleneck. Experimental tests demonstrate that it is unable to process more than four data streams (clients) simultaneously, beyond which the system crashes. A possible solution could be to use a more powerful node or redundancy in edge-nodes, and distributing the clients according to a specific schema (*e.g.*, proximity).

VII. Conclusions and Future Work

In this work, we explored the integration of multimodal data within a MCU-optimized FL framework. To overcome limitations related to resource-constrained devices usage (in terms of memory, computational capacity, and energy), we propose an intermediate cross-modal FL, which does not require simultaneous access to different modalities at the client-side and can be distributed hierarchically between the client and the edge node (while preserving data privacy).

This study confirms that FL on MCUs can be a feasible and efficient solution for privacy-preserving activity recognition when augmented by multimodal data. However, several directions can enhance this research:

- *Personalization strategies*: Implement client-level personalization approaches to mitigate the effects of user bias in Primary Participant-Wise distributions while maintaining privacy guarantees;
- *Energy optimization*: Further optimization of the model architectures and training protocols to minimize power consumption, enhancing real-time viability on MCUs;
- *Real-world deployment and testing*: Validate the system in real-life settings with live data streams to assess generalizability, user compliance, and long-term reliability;
- *Model memory optimization*: All model architectures within the ML system can be optimized using strategies such as pruning and quantization;
- *Privacy-preserving enhancements*: Integrate secure aggregation and privacy techniques to strengthen user data confidentiality without compromising performance.

This work lays the foundation for practical, multimodal, and federated TinyML solutions in IoT-edge computing applications, such as fall-detection, smart health monitoring, and activity recognition in privacy-sensitive environments.

Acknowledgment

This work is part of the research activities realized within the projects iQualitAria (CUP. B59J24000180005), Fondo di Crescita Sostenibile - Sportello PN RIC 2021/2027 “Scoperta Imprenditoriale”, as well as the projects SERICS (CUP PE00000014, under the NRRP MUR program funded by the EU - NGEUm).

References

[1] F. Alshehri, G. Muhammad, ”A Comprehensive Survey of the Internet of Things (IoT) and AI-Based Smart Healthcare,” in *IEEE Access*, vol. 9, pp. 3660-3678, 2021.

[2] I. Keshta, ”AI-driven IoT for smart health care: Security and privacy issues,” in *Informatics in Medicine Unlocked*, Vol. 30, pp. 100903, 2022.

[3] F. Cremonesi, V. Planat, V. Kalokyri, H. Kondylakis, T. Sanavia, V.M.M. Resinas, B. Singh, S. Uribe, ”The need for multimodal health data modeling: A practical approach for a federated-learning healthcare platform,” in *Journal of Biomedical Informatics*, vol. 141, pp. 104338, 2023.

[4] C. Dhasarathan, M. K. Hasan, S. Islam, S. Abdullah, S. Khapre, D. S., A. A. Alsulami, A. Alqahtani. ”User privacy prevention model using supervised federated learning-based block chain approach for internet of Medical Things,”, in *CAAI Transactions on Intelligence Technology*, pp. 1-15, 2023.

[5] P. Qi, D. Chiaro, F. Piccialli, ”FL-FD: Federated learning-based fall detection with multimodal data fusion,” in *Information Fusion*, vol. 99, pp. 101890, 2023.

[6] P. Fusco, G. P. Rimoli, M. Ficco, ”TinyML and Federated Learning for Resource-Constrained Medical Devices,” in *Artificial Intelligence Techniques for Analysing Sensitive Data in Medical Cyber-Physical Systems: System Protection and Data Analysis*, vol. 12, pp. 113-126, 2025.

[7] B. Xiong, X. Yang, and C. Xu, ”A unified framework for multi-modal federated learning,” in *Neurocomputing*, vol. 480, pp. 110–118, 2022. S. Ketu and P. K. Mishra,

[8] S.F. Ahmed, M.S.B. Alam, S. Afrin, S.J. Rafa, N. Rafa, A.H. Gandomi, ”Insights into Internet of Medical Things (IoMT): Data fusion, security issues and potential solutions,” in *Information Fusion*, vol. 102, pp. 102060, 2024.

[9] M. Bhamare, P. V. Kulkarni, R. Rane, S. Bobde, R. Patankar, ”TinyML applications and use cases for healthcare,” in *TinyML for Edge Intelligence in IoT and LPWAN Networks*, vol. 4, pp. 331-353, 2024.

[10] ”Internet of Healthcare Things: A contemporary survey Author links open overlay panel,” in *Journal of Network and Computer Applications*, vol. 192, pp. 103179, 2021.

[11] M. Ficco, A. Guerriero, E. Milite, F. Palmieri, R. Pietrantuono, S. Russo, ”Federated learning for iot devices: Enhancing tinyml with on-board training,” in *Information Fusion*, vol. 104, pp. 1-19, 2024.

[12] C. N. da Silva and C. V. S. Prazeres, ”Tiny Federated Learning for Constrained Sensors: A Systematic Literature Review”, in *IEEE Sensors Reviews*, vol. 2, pp. 17-31, 2025.

[13] K. Ibrar, F. Palmieri, P. Fusco, M. Ficco, ”Cross-Model Federated Learning-Based Network Traffic Classification,” in Proc. of the *Int. Conf. on Advanced Information Networking and Applications (AINA2025)*, vol. 3, pp.250-259, 2025.

[14] F. Zhao, C. Zhang, B. Geng, ”Deep Multimodal Data Fusion,” in *ACM Computing Surveys*, vol 56, pp. 1-36, 2024.

[15] S. Song et al., ”Multimodal multi-stream deep learning for egocentric activity recognition,” in Proc. IEEE Conf. the *Computer Vision and Pattern Recognition Workshops*, pp. 378–385, 2016.

[16] G. Muhammad, F. Alshehri, F. Karray, A. El Saddik, M. Alsulaiman, T.H. Falk, ”A comprehensive survey on multimodal medical signals fusion for smart healthcare systems”, in *Information Fusion*, vol. 76, pp. 355–375, 2021.

[17] M.M. Islam, S. Nooruddin, F. Karray, G. Muhammad, ”Multi-level feature fusion for multimodal human activity recognition in Internet of Healthcare Things,” in *Information Fusion*, vol. 94, pp. 17-31, 2023.

[18] F. Miao, Z.-D. Liu, J.-K. Liu, B. Wen, Y. Li, ”Multi-Sensor Fusion Approach for Cuff-Less Blood Pressure Measurement,” in *IEEE Journal of Biomedical and Health Informatics*, vol. 24, pp. 79-91, 2020.

[19] X. Yang, B. Xiong, Y. Huang, C. Xu, ”Cross-modal federated human activity recognition via modality-agnostic and modality-specific representation learning,” in Proc. of the *Conf. on Artificial Intelligence*, vol. 36, pp. 3063–3071, 2022.

[20] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, C. Peñafort-Asturiano, ”UP-Fall detection dataset: A multimodal approach”, in *Sensors*, vol. 19, pp. 1988, 2019.

[21] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, ”Federated learning with non-IID data,” 2018, in *arXiv:1806.00582*.

[22] H. Wang, M. Yurochkin, Y. Sun, D. S. Papailiopoulos, and Y.Khazaeni, ”Federated learning with matched averaging,” in Proc. of the *Int. Conf. Learn. Representations*, 2020, pp. 1-16.

[23] A. Reisizadeh, F. Farnia, R. Pedarsani, and A. Jadbabaie, ”Robustfederated learning: The case of affine distribution shifts,” in Proc. of the *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 21554-21565, 2020

[24] Q. Li, B. He, B. and D. Song, ”Model-Contrastive Federated Learning,” in *IEEE/CVF Conference On Computer Vision And Pattern Recognition (CVPR)*. pp. 10708-10717, 2021.

[25] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, ”FedBN: Federated learning on non-IID features via local batch normalization,” in Proc. of the *Int. Conf. Learn. Representations*, pp. 1-27, 2021.

[26] C. I. Bercea, B. Wiestler, D. Rueckert, and S. Albarqouni, ”FedDis: Disentangled federated learning for unsupervised brain pathology segmentation,” in *arXiv:2103.03705*, 2021.

[27] D. Wang, T. Zhao, W. Yu, N. V. Chawla, and M. Jiang, ”Deep multimodal complementarity learning,” in *IEEE Tran. on Neural Networks and Learning Systems*, vol. 34, pp. 10213–10224, 2023.

[28] T. V. Steenkiste, D. Deschrijver, and T. Dhaene, ”Sensor Fusion using Backward Shortcut Connections for Sleep Apnea Detection in Multi-Modal Data,” in *Proceedings of the Machine Learning for Health NeurIPS Workshop*, vol. 116, pp. 112-125, 2020.

[29] L. M-Villaseñor, H. Ponce, K. P.-Daniel, ”Deep learning for multimodal fall detection,” in Proc. of the *IEEE Int. Conf. on Systems, Man and Cybernetics*, pp. 3422–3429, 2019.

[30] W. Wei, Q. Jia, Y. Feng, G. Chen ”Emotion Recognition Based on Weighted Fusion Strategy of Multichannel Physiological Signals”, in *Computational Intelligence and Neuroscience*, pp. 1-7, 2018.

[31] W. Huang, D. Wang, X. Ouyang, J. Wan, J. Liu, T. Li, "Multimodal federated learning: Concept, methods, applications and future directions", in *Information Fusion*, vol. 112, pp. 102576, 2024.

[32] L. Zong, Q. Xie, J. Zhou, P. Wu, Zhang, X. and B. Xu, "FedCMR: Federated Cross-Modal Retrieval," in Proc. Of the *44th Int. ACM SIGIR Conf. On Research And Development In Information Retrieval*, pp. 1672-1676, 2021.

[33] Q. Yu, Y. Liu, Y. Wang, K. Xu, and J. Liu, "Multimodal federated learning via contrastive representation ensemble," 2023, arXiv preprint arXiv:2302.08888.

[34] Y. Zhao, P. Barnaghi, and H. Haddadi, "Multimodal federated learning on iot data," in Proc. of the *IEEE/ACM 6th Int. Conf. on Internet-of-Things Design and Implementation (IoTDI)*, pp. 43–54, 2022.

[35] J. Chen and A. Zhang, "Fedmsplit: Correlationadaptive federated multi-task learning across multimodal split networks," in Proc. of the *28th ACM Conf. on Knowledge Discovery and Data Mining*, pp. 87–96, 2022.

[36] X. Yang, B. Xiong, Y. Huang, C. Xu, "Cross-Modal Federated Human Activity Recognition," in *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 46, pp. 5345-5361, 2024.

[37] T. Flores, M. Silva, P. Andrade, I. Silva, E. Sisinni, P. Ferrari, S. Rinaldi, "A tinyml soft-sensor for the internet of intelligent vehicles," in Proc. of the *IEEE Int. Workshop on Metrology for Automotive*, pp. 18–23, 2022.

[38] M. Hashir, N. Khalid, N. Mahmood, M. A. Rehman, M. Asad, M. Zubair, Y. Massoud, "A tinyml based portable, low-cost microwave head imaging system for brain stroke detection," in Proc. of the *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, pp. 1–4, 2023.

[39] X. Yu, S. Park, D. Kim, E. Kim, J. Kim, W. Kim, Y. An, S. Xiong, "A practical wearable fall detection system based on tiny convolutional neural networks," in *Biomedical Signal Processing and Control*, vol. 86, pp. 105325, 2023.

[40] B. Sudharsan, S. Salerno, R. Ranjan, "Tinyml-cam: 80 fps image recognition in 1 kb ram," in Proc. of the *28th Annual Int. Conf. on Mobile Computing And Networking*, 2022, pp. 862–864.

[41] H. Ren, D. Anicic, T. A. Runkler, "Tinyol: Tinyml with online-learning on microcontrollers," in Proc. of the *Int. Joint Conf. on Neural Networks (IJCNN)*, 2021, pp. 1–8.

[42] L. Ji, Z. Ligeng, C. Wei-Ming, W. Wei-Chen, G. Chuang, S. Han, "On-device training under 256kb memory," 2022, arXiv preprint arXiv:2206.15472.

[43] C. Profentzas, M. Almgren, O. Landsiedel, "Minilearn: On-device learning for low-power IoT devices," in Proc. of the *Int. Conf. on Embedded Wireless Systems and Networks*, 2023, pp. 1–11.

[44] L. Wulfert et al., "AIfES: A Next-Generation Edge AI Framework," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 46, pp. 4519-4533, 2024.

[45] H. Cai, C. Gan, L. Zhu, S. Han, "Tinytl: Reduce activations, not trainable parameters for efficient on-device learning," in Proc. of the *34th Int. Conf. on Neural Inf. Processing Systems*, 2020, pp. 11285-11297.

[46] L. Ravaglia, M. Rusci, D. Nadalini, A. Capotondi, F. Conti, L. Benini, "A tinyml platform for on-device continual learning with quantized latent replays," 2021, arXiv:2110.10486.

[47] L. Ji, Z. Ligeng, C. Wei-Ming, W. Wei-Chen, G. Chuang, S. Han, "On-device training under 256kb memory," 2022, arXiv preprint arXiv:2206.15472.

[48] H. Huang, W. Zhuang, C. Chen and L. Lyu, "FedMef: Towards Memory-Efficient Federated Dynamic Pruning," in Proc. of the *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2024, pp. 27538-27547.

[49] H. Huang, L. Zhang, C. Sun, R. Fang, X. Yuan and D. Wu, "Distributed Pruning Towards Tiny Neural Networks in Federated Learning," in Proc. of the *IEEE 43rd Int. Conf. on Distributed Computing Systems (ICDCS)*, 2023, pp. 190-201.

[50] S. Marilina, "Optimisation of neural networks for applications based on multimodal data fusion running on MCUs and MPUs," in Proc. of the *Int. Conf. on Mobile and Ubiquitous Multimedia*, 2024, pp. 552-554.

[51] S. Nooruddin, M. M. Islam, F. Karray, G, Muhammad, "A multi-resolution fusion approach for human activity recognition from video data in tiny edge devices," in *Information Fusion*, vol. 100, pp. 101953, 2023.

[52] N. Kayarvizhy, B. M. Gera, B. Sharma, S. Dakshinamurthy, "TinyML based Deep Learning Model for Activity Detection," in *Int. Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, pp. 149-158, 2023.

[53] K. Kalenberg, H. Müller, T. Polonelli, A. Schiaffino, V. Niculescu, C. Cioflan, "Stargate: Multimodal Sensor Fusion for Autonomous Navigation on Miniaturized UAVs," in *IEEE Internet of Things Journal*, vol. 11, pp. 21373-21390, 2024.

[54] H.-A. Rashid, U. Kallakuri, T. Mohsenin, "TinyM2Net-V2: A Compact Low-power Software Hardware Architecture for Multimodal Deep Neural Networks," in *ACM Transactions on Embedded Computing Systems*, vol. 23, pp. 1-23, 2024.

[55] M. Gibbs, K. Woodward, E. Kanjo, "Combining Multiple tinyML Models for Multimodal Context-Aware Stress Recognition on Constrained Microcontrollers," in *in IEEE Micro*, vol. 44, pp. 67-75, 2024.

[56] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in Proc. of the *Int. Conf. of Neural Inf. Process. Syst.*, 2016, pp. 343-351.

[57] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *IEEE Trans. On Pattern Analysis And Machine Intelligence*, vol. 44, pp. 5962-5979, 2022.

[58] F. Yu, A. Rawat, A. Menon, S. Kumar, "Federated Learning with Only Positive Labels", (2020), https://arxiv.org/abs/2004.10342

[59] S. Ji, Z. Zhang, S. Ying, L. Wang, X. Zhao, Y. Gao, "Kullback–Leibler divergence metric learning," in *IEEE Tran. On Cybernetics*, vol. 52, pp. 2047-2058, 2020.

[60] Y. M. Galvão, J. Ferreira, V. Albuquerque, A. Vinícius, J. B. Fernandes, "A multimodal approach using deep learning for fall detection", in *Expert Systems with Applications*, vol. 168, no. 114226, 2021.

[61] Wen-Nung Lie, Fang-Yu Hsu, Y. Hsu, "Fall-down event detection for elderly based on motion history images and deep learning", in Proc. of the *Int. Workshop on Advanced Image Technology (IWAIT)*, vol. 11049, pp. 787-792, 2019.

[62] B. KP Horn, B. G Schunck, "Determining optical flow", in *Artificial intelligence*, vol. 17, pp. 185-203, 1981.

[63] Aldamen, D., Moltisanti, D., Kazakos, E., Doughty, H., Munro, J., Price, W., Wray, M., Perrett, T. & Ma, J. EPIC-KITCHENS-100. (University of Bristol,2020), https://data.bris.ac.uk/data/dataset/2g1n6qdydwa9u22shpxqzp0t8m/

[64] McMahan, B., Moore, E., Ramage, D., Hampson, S. & Arcas, B. Communication-efficient learning of deep networks from decentralized data. *Artificial Intelligence And Statistics*. pp. 1273-1282 (2017)

[65] T Dinh, C., Tran, N. & Nguyen, J.Personalized federated learning with moreau envelopes. *Advances In Neural Information Processing Systems*, vol. 33 pp. 21394-21405, 2020.

[66] Liang, P., Liu, T., Ziyin, L., Allen, N., Auerbach, R., Brent, D., Salakhutdinov, R. & Morency, L. Think locally, act globally: Federated learning with local and global representations. *ArXiv Preprint ArXiv:2001.01523*. (2020)

[67] Finn, C., Abbeel, P. & Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *Proceedings Of The 34th International Conference On Machine Learning*, vol. 70, pp. 1126-1135 (2017), https://proceedings.mlr.press/v70/finn17a.html

[68] Fallah, A., Mokhtari, A. & Ozdaglar, A. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances In Neural Information Processing Systems*. **33** pp. 3557-3568 (2020)

[69] Li, X., Jiang, M., Zhang, X., Kamp, M. & Dou, Q. Fedbn: Federated learning on non-iid features via local batch normalization. *ArXiv Preprint ArXiv:2102.07623*. (2021)

[70] Collins, L., Hassani, H., Mokhtari, A. & Shakkottai, S. Exploiting shared representations for personalized federated learning. *International Conference On Machine Learning*. pp. 2089-2099 (2021)

[71] Eclipse Mosquitto - An open source MQTT broker, available at: https://mosquitto.org/ [Last access May 2025]