Information Fusion 99 (2023) 101890

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

FL-FD: Federated learning-based fall detection with multimodal data fusion

Pian Qi, Diletta Chiaro, Francesco Piccialli [*]

University of Naples Federico II, Department of Mathematics and Applications "R. Caccioppoli", Naples, Italy

ARTICLE INFO

Keywords:
Federated learning
Data fusion
Multimodal data fusion
IoMT
Fall detection

ABSTRACT

Multimodal data fusion is a critical element of fall detection systems, as it provides more comprehensive information than single-modal data. Yet, data heterogeneity between sources has posed a challenge for the effective fusion of such data. This paper proposes a novel multimodal data fusion method under a federated learning (FL) framework that addresses the privacy concerns of users while exploiting the complementarity of such data. Specifically, we fuse time-series data from wearable sensors and visual data from cameras at the input level, where the data is first transformed into images using the Gramian Angular Field (GAF) method. Moreover, each user is treated as a private client in the FL system whereby the fall detection model is trained without requiring the sharing of user data. The proposed method is evaluated using the UP-Fall dataset, where we perform different fall detection tasks: binary classification for fall and non-fall detection yields a remarkable accuracy of 99.927%, while multi-classification for different fall activity recognition attains an accurate result of 89.769%.

1. Introduction

Over the past few decades, the Internet of Medical Things (IoMTs) has evolved significantly with technologies that provide many benefits for people's health and safety. The deployment of sensor devices in IoMTs has become commonplace, with devices such as wearable devices, cameras, and various clinical instruments providing a diverse and vast amount of data for medical diagnosis [1]. IoMTs bring convenient medical services and enable remote monitoring of patients' conditions through real-time data collection from sensor devices. On the other hand, machine learning (ML) is a powerful tool for data analysis thanks to its ability to capture hidden relationships between data. Nowadays, ML is increasingly integrated into IoMTs, bringing a reliable means of support for disease prevention and diagnosis, and greatly contributing to the development of healthcare [2].

Among the many studies on condition detection related to IoMTs, fall detection is a task of great importance [3]. In particular, for special populations such as the elderly, children, and pregnant women, falls can often lead to serious consequences, including disability and even death. According to the World Health Organization (WHO), falls are the second leading cause of death from unintentional injuries worldwide [4]. It is widely recognized that in addition to exercising caution to prevent falls, timely detection and alerting when a fall occurs is particularly crucial. To this end, a variety of sensor devices, such as wearable devices, smart bracelets, and indoor cameras, have been deployed in IoMTs, producing real-time data on human activities. By analyzing these raw sensor data with ML, a reliable and convenient fall detection system can be developed, offering enhanced security and peace of mind to individuals.

While a number of fall detection systems are available, many of them rely solely on single-modal data [5]. However, the limited amount of information contained within single-modal data impacts the precision of the detection systems when compared to those based on multimodal data. Multimodal data-based systems have shown improved performance on account of researchers fusing data from sources, such as accelerometers and gyroscopes [6], which has helped to provoke thought towards the implementation of data fusion. Despite being different in nature, time-series data and visual data from cameras and wearable devices have also been fused together by some researchers [7]. This provides complementary information that can be leveraged to further enhance the accuracy of fall detection systems. Studies have shown that fusing multimodal data as input can considerably improve fall detection accuracy. It is worth noting that when fusing multimodal data for fall detection, the first task is to consider how the data is fused. In general, multimodal data fusion can be divided into three types: input-level fusion (data-level fusion), feature-level fusion, and decision-level fusion [8], see Fig. 1 for illustration. Regarding the feature-level fusion, in Martínez et al. [9], manual features – such as mean or standard deviation – were extracted from the original data and merged together for use in ML-based fall detection methods. Those methods include Random Forests (RF) [10], Support Vector Machines (SVM) [11], Multi-layer Perceptron (MLP) [12], and

[*] Corresponding author.
E-mail address: francesco.piccialli@unina.it (F. Piccialli).

https://doi.org/10.1016/j.inffus.2023.101890
Received 2 May 2023; Received in revised form 6 June 2023; Accepted 12 June 2023
Available online 23 June 2023

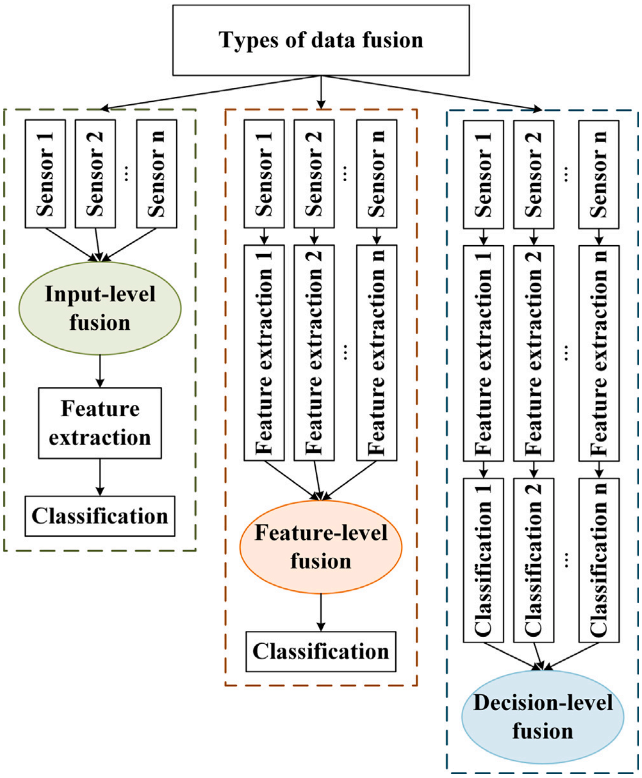


Fig. 1. Types of data fusion in classification tasks.

k-Nearest Neighbors (k-NN) [13]. This feature fusion method has some limitations because the extraction of manual features may lead to the loss of important information in the data. Islam et al. [14] used two different neural network (NN) branches to extract features of the two modal data separately and performed feature fusion on this basis, but this method also led to the overall complexity of the NN model. In the decision-level fusion, [15] proposed a decision-level data fusion method in which the data of each modality is classified as the input of a separate sub-NN, and the final decision of which classification model wins is made by a majority voting system. However, this approach highlights the competitive nature of the models and does not take full advantage of the complementary nature of multimodal data. Among the data fusion approaches, input-level fusion of multimodal data is not as developed due to the challenge of fusing data with different morphologies. However, this fusion method has the ability to preserve more information within the data than other methods such as feature-level or decision-level fusion. As a result, there is a pressing need for the development of effective input-level fusion methods to fully capitalize on the benefits of retaining higher amounts of preserved information in heterogeneous multimodal data. Such efforts would be of significant research value.

In recent years, countries around the world are paying more and more attention to citizens' privacy and information security, and it is necessary to take responsibility for privacy protection while developing data fusion in IoMTs [16]. Federated learning (FL) has become a learning paradigm favored by researchers because of its stronger privacy protection compared to traditional ML [17]. In general, in FL, there exists a server responsible for the coordination, with multiple clients carrying local data. FL is an emerging learning paradigm in the past few years, which is radically different from traditional data-centralized ML because of its collaborative server–client training model. In traditional data-centric ML, a large amount of user data is collected for model training, which poses a significant threat to user privacy. However, in FL, data is distributed across different clients, which range from small mobile devices to large organizations. Generally, in FL, there is a server responsible for coordination and multiple clients hosting local data. The clients do not share data among themselves, but only perform local gradient updates based on their local data, and then perform model updates with the server, such that the collaborative training yields the final desired model. The learning process is as follows: first, each client downloads the latest global model for initialization and performs local training on the local data; then, the clients upload the latest local model to the server; finally, the server performs model aggregation and updates the global model. The above steps are repeated until the model reaches the required accuracy [18]. A general framework for FL is shown in Fig. 2. As in FL the models rather than raw data are shared, the privacy of the data is preserved. The application of FL to fall detection to protect user privacy is of great practical importance.

To the best of our knowledge, no previous research work has combined multimodal data fusion with user privacy protection for fall detection. Therefore, this paper proposes a method that fuses multimodal data at the input level within a FL framework. The aim is to create a fall detection system that is efficient and preserves user privacy. First, to protect users' privacy from disclosure, each user acts as a separate client in the federated system. Each client has a local data fusion module for multimodal data fusion to generate local data for training. In the data fusion module, the time series data from the wearable sensors are converted into images using the Gramian Angular Field (GAF) method; the GAF images are then fused with the visual data from the cameras. Second, each client performs local model training based on local fused data, and then the server performs model aggregation to complete the training of the global model. The local model is a simple 3-layer CNN in order not to cause a communication burden in FL. Finally, experimental evaluation on the UP Fall dataset shows that our proposed model combines fall detection capability with privacy preservation.

The novelties of the proposed method in this paper are as follows. First, using FL as the relying framework, the security and privacy of the data are fully considered. Most of the previous studies about fall detection are based on traditional DL, which undoubtedly leaks user privacy. In traditional DL, users need to upload their activity data to the data center for model training. These data record users' facial and physical characteristics, and the consequences of leakage can be imagined. Second, this paper achieves the fusion of time series data and visual data at input-level with the aim of maximizing the retention of data information. Previous studies on multimodal data fusion are often based on feature- or decision-level fusion, leading to a lack of raw information. This paper is the first to introduce input-level data fusion into the field of fall detection.

The main contributions of this paper are as follows:

(1) A method for fusing input-level data is proposed, which combines 1D time-series data with 2D visual data to achieve information complementarity.
(2) A user privacy-preserving FL framework is proposed to ensure data security.
(3) The effectiveness of the proposed method for fall detection is illustrated by experimental results on the UP Fall dataset, demonstrating that multimodal data fusion enhances fall detection performance when compared to single-modal data.

The rest of the paper is structured as follows. Section 2 presents related works, including data fusion and FL in fall detection. Section 3 introduces the proposed framework, explaining in detail the data fusion approach and the FL setup. Section 4 shows the evaluation experiments, including the dataset and experimental setup, and the analysis and discussion of the experimental results. A summary of the full paper is presented in Section 5.

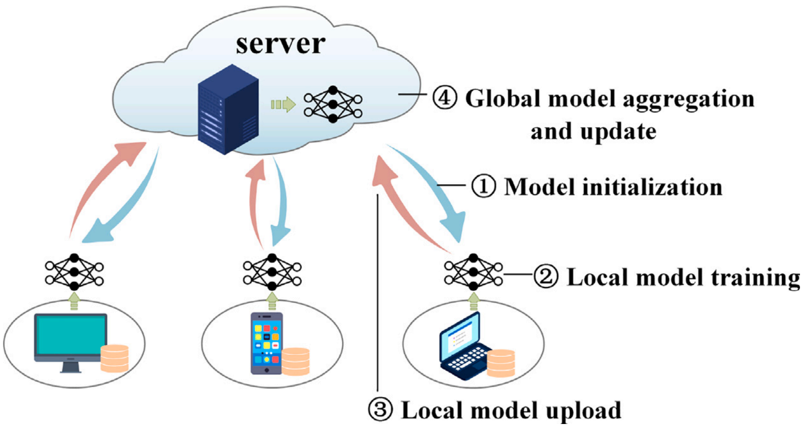


Fig. 2. A general framework for FL.

2. Related works

This section presents recent research on fall detection from two perspectives. Firstly, various approaches based on data fusion in fall detection are discussed. Secondly, the increasing use of FL in fall detection is examined. These works are discussed in Section 2.1 and Section 2.2, respectively.

2.1. Data fusion in fall detection

Focusing on the research topic of fall detection, several researchers have tried to improve the accuracy of detection through data fusion schemes. We review and discuss these data fusion proposals in three aspects: input level, feature level, and decision level.

In previous research work on fall detection, input-level fusion was mostly based on single-modal data rather than multimodal, because the heterogeneity of the data posed some challenges for fusion. In [19], Auvinet et al. fused 2D contour projections from multiple cameras to reconstruct the 3D volume of the human body. Based on this fusion method, a threshold detection method is used to determine whether a fall has occurred. This method can simulate real scenes better compared to 2D image data, but the data is only visual. It does not make use of other modal data, which may lead to the problem that the reconstructed 3D model is not fine enough. In [20], Xie et al. used an input-level fusion method to classify the UP Fall dataset for fall events. This was done by first extracting the human skeletal sequences from the RGB images and then fusing five points of the face into one key point. This method has the advantage of simplicity but does not explore the fusion of multimodal data.

Many research works fuse features extracted from the data, which can be further classified into manual feature-based and NN-based features. In [21], Cai et al. combined acceleration from the pose sensor, with the human skeleton sequence extracted from the video. After fusing the extracted features such as standard deviation, and maximum minimum value, the GBDT algorithm was used for fall action classification. The performance of the scheme is verified in comparison with algorithms such as SVM and NN. Similarly, in the articles [22,23], manual features were selected for fusion, but the subsequent classification task was performed by a bidirectional long short-term memory (Bi-LSTM) or convolutional neural network (CNN) network. In [24], the authors used infrared sensors to collect different fall movements of the human body. After fusing the variation values of features such as the center of mass, velocity, and body area, SVM is used for fall classification. All the above research works involve the fusion of manual features, but the selection of features is relatively dependent on the researcher's experience. The features extracted based on NNs are less experience-dependent than manual features. In [25], the authors proposed a fall detection method based on radar signals. The method fuses features extracted from three signal maps by three NNs separately to recognize multiple fall actions in real scenes. The proposal exploits the privacy-preserving nature of radar signals, but a single temporal signal may not carry as much information as multimodal data. Amsaprabhaa et al. [26] used two NNs, spatiotemporal graph convolutional network (STGCN) and 1D-CNN, to generate two sets of features for human skeleton sequences. After cascade fusion of the features, fall prediction was performed. The two different NNs can extract diverse features, but the method is still based on only a single visual data, which may lead to misclassifying a fall-like action as a fall.

Decision-level data fusion approaches appear in several research efforts. In [27], Yi et al. designed a practical fall detection application that monitors heart beat rate, acceleration, and body temperature to synthetically assess falls and initiate alerts via a smartphone program. The application only differentiates the fall direction and may not be able to identify the severity of different fall postures. De et al. [28] proposed a two-channel decision-level data fusion method for fall detection on the UR Fall dataset. In this method, for the data from the camera, one channel performs threshold-based classification of the aspect ratio of the human silhouette, and the other channel performs classification of key frames using the k-NN algorithm. The results of the two channels are then evaluated together to determine whether a fall event has occurred. This method analyzes only visual data and does not involve multimodal data. It is worth noting that decision-level data fusion essentially analyzes data from different single-modal separately and then discriminates between the classification results, which does not take advantage of the complementary information of different modal data.

As seen from the above-related research works, there is still room for exploring the input-level fusion of multimodal data in fall detection. Moreover, compared with feature-level and decision-level fusion, input-level fusion can fully retain the information carried by different modal data and help to enhance the accuracy of fall event detection.

2.2. FL in fall detection

In recent years, FL has become an irreplaceable paradigm and increasingly active because of its unique privacy-preserving properties. With the increasing awareness of data protection, there have been several research works on fall detection based on FL [29,30].

In [31], the authors designed a privacy-preserving enhanced FL framework for fall detection experiments on acceleration in the UP Fall dataset. The carefully designed cryptosystem in this framework has good privacy-preserving features and does not lead to excessive communication consumption. However, to prevent the negative impact of non-independent identical distribution (non-iid) of client data, each client must also upload a portion of data during the local model upload phase. This practice actually defeats the original purpose of FL, which

Table 1
Literature summary table.
The following table reports the main findings of the reviewed literature relating data fusion and FL in fall detection.

Papers	Multimodal	Data fusion	FL	Contributions	Limitations
[19,20]	✗	✓	✗	Input-level data fusion provides convenience subsequent fall detection tasks.	The data used for fusion have the same modality and the systems lack users' privacy protection.
[21,22,22–24]	✓	✓	✗	Feature-level data fusion helps for improving the accuracy of fall detection.	The features are manually extracted, and how to choose depends on the experience of the researcher. Lack of users privacy protection.
[25,26]	✗	✓	✗	Feature-level data fusion helps for improving the accuracy of fall detection.	The features used for fusion come from the same modality and the systems lack users' privacy protection.
[28]	✓	✓	✗	Data Fusion at decision Level.	Lack of users' privacy protection.
[31–35]	✗	✗	✓	The importance of protecting user privacy is considered, and FL is utilized to achieve the purpose.	The data used are single modality and could not provide complementary information.
[36]	✓	✗	✓	Representation of multimodal data is realized in FL, and user privacy is protected.	Data fusion was not achieved and could not provide complementary information.
Ours	✓	✓	✓	Multimodal data fusion at the input level provides information complementation, and the FL-based system effectively protects users' privacy.	

is not to share data among clients. In particular, the non-iid of data is an ongoing problem with FL, due to the fact that different clients usually have different data distributions, which is a major difference between FL and traditional data-centralized ML. A FL framework based on channel state information is proposed in [32], and the experimental results are comparable to the centralized model. However, in that experiment, the authors did not consider the non-iid setting of the data. In [33], the authors used manual features of acceleration data as input to the local model to detect falls. This FL framework incorporates an extreme learning machine for improving the performance of the global model. The articles [34,35] both proposed a FL framework for home health care, and indicated that the proposed framework achieved good results in fall detection. The above-mentioned research works show that FL is practical and privacy-preserving for fall detection, but they are all based on data from a single modal. In [36], the authors propose a FL approach to handle multimodal data for fall detection without sharing the original data or the representation space of the data. In this method, some clients have single-modal data, while some clients have two-modal data. In the local training phase, two autoencoders need to be trained for each client, and the local model without the corresponding modal data is frozen. By aggregating the two autoencoders for each client, the encoded representations of the different modal data are learned. This approach provides an idea of how to handle multimodal data in FL, but the drawback is that in each round of model aggregation, the client needs to upload two models, which introduces additional communication overhead.

Table 1 summarizes the contributions and limitations of the related works. It is clear that there are still unresolved challenges and areas that need to be explored in current research. Therefore, this paper proposes a multimodal data fusion approach based on FL for fall detection and activity recognition. The fusion of multimodal data can improve the accuracy of classification models, while the FL-based framework ensures the privacy of user data.

3. Proposed framework

This section presents the proposed framework, and Section 3.1 is a general introduction of the framework. Section 3.2 describes the proposed approach for data fusion, which illustrates how multimodal data fusion is performed at the input level. Section 3.3 describes the deployment of FL, which improves the security of the proposed framework.

3.1. Overview

This paper proposes an FL framework for fall detection using multimodal data while preserving user privacy, as shown in Fig. 3. To make the framework adaptable to the non-iid case, each user represents a client in the FL system, making it resilient to the addition or removal of clients. In this framework, each client first downloads the latest global model from the server for initialization. The client has access to multimodal data collected by IoMTs devices, including time-series data and visual data. Additionally, each client has a data fusion module for fusing multimodal data at the input level. After training on the local data using the downloaded models, the clients upload the model parameters to the server. Finally, the server aggregates the received local models and updates the global model.

3.2. Multimodal data fusion

Data fusion at the input level aims at minimizing the loss of data information. In this paper, we focus on one-dimensional time-series data recorded by wearable sensors and two-dimensional visual data recorded by cameras. Generally, it is difficult to directly fuse these two kinds of data in terms of shape because of their heterogeneity. Therefore, we use the Gramian Angular field (GAF) [37] method to encode the 1D time series data into 2D images; then the encoded images are stacked with the visual data on the channel to achieve fusion. The detailed process steps are as follows:

A. *Encoding of time series data*
There are various methods to encode 1D time series data into 2D images, such as Markov Transform Field (MTF) [38], Recurrence Plot (RP) [39]. However, in the MTF method, hyperparameters (fractional bins k) need to be set, and different values lead to different imaging effects. Similarly, in the RF method, the imaging effect has a strong empirical dependence on the selection of the hyperparameter (critical distance ϵ). In this paper, the GAF method is chosen for encoding time series data as it avoids the issue of selecting hyperparameters. In GAF, time series data is represented in a polar coordinate system rather than a traditional Cartesian coordinate system, preserving the absolute temporal relationships between data points. For the GAF method, the detailed procedure is as follows:

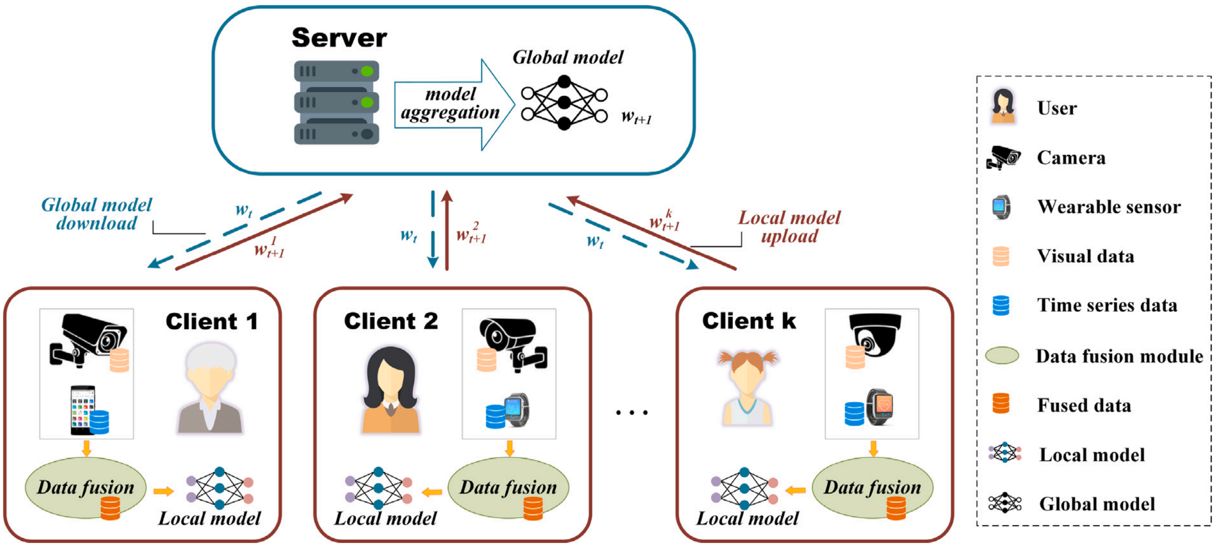


Fig. 3. The proposed framework is based on FL. In each communication round t, the client k downloads the latest model w_t from the server. The client then trains locally using the fused multimodal data and sends the updated model w^k_{t+1} to the server. The server performs model aggregation by combining the local models received from all clients and updates the global model w_{t+1}.

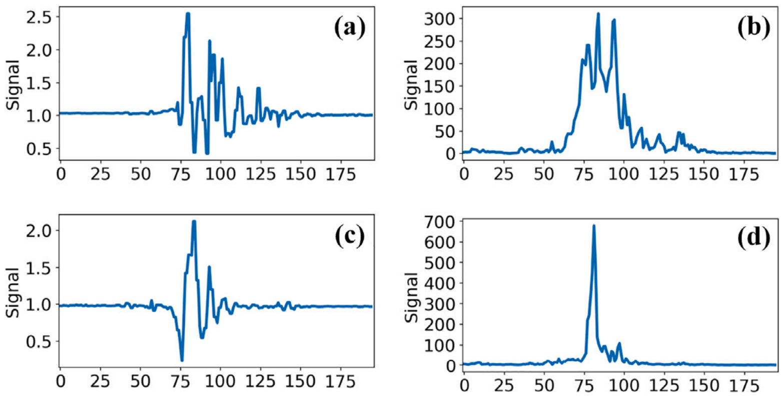


Fig. 4. Signals from wearable sensors in UP Fall dataset, (a) Ankle accelerometer, (b) Ankle angular velocity, (c) Belt accelerometer, (d) Belt angular velocity.

Given a raw time series $X_T = \{x_1, x_2, \dots, x_n\}$, it first needs to be normalized to the interval $[-1, 1]$, and obtain:

$$\tilde{X}_T = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n\}$$

In the second step, the scaled time series $\tilde{X}_T$ is mapped to polar coordinates the value $\tilde{x}_i$ is encoded as angular cosine, and the timestamp t_i is encoded as radius.

$$\begin{cases} \phi = \arccos\left(\tilde{x}_i\right), -1 \leq \tilde{x}_i \leq 1, \tilde{x}_i \in \tilde{X} \\ r = \frac{t_i}{N}, t_i \in \mathbb{N} \end{cases}$$

where N is a constant factor that regularizes the span of the polar coordinate system [40], here refers to the number of time points. In the third step, the temporal correlation between different time points is expressed as a GAF using trigonometric sum functions. The GAF is defined as follows.

$$GAF = \begin{bmatrix} \cos\left(\phi_1 + \phi_1\right) & \cdots & \cos\left(\phi_1 + \phi_n\right) \\ \cos\left(\phi_2 + \phi_1\right) & \cdots & \cos\left(\phi_2 + \phi_n\right) \\ \vdots & \ddots & \vdots \\ \cos\left(\phi_n + \phi_1\right) & \cdots & \cos\left(\phi_n + \phi_n\right) \end{bmatrix}$$

Here, the GAF is a matrix of size $n \times n$. that is, the length of the time series determines the size of the encoded image X_G. Fig. 4 shows signals from wearable sensors in UP Fall dataset [9], and Fig. 5 is the transformation into GAF images.

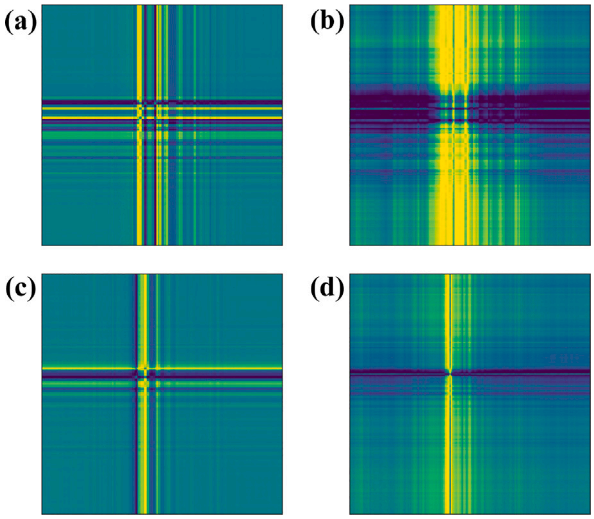


Fig. 5. GAF images of different signals in UP Fall dataset, (a) Ankle accelerometer, (b) Ankle angular velocity, (c) Belt accelerometer, (d) Belt angular velocity.

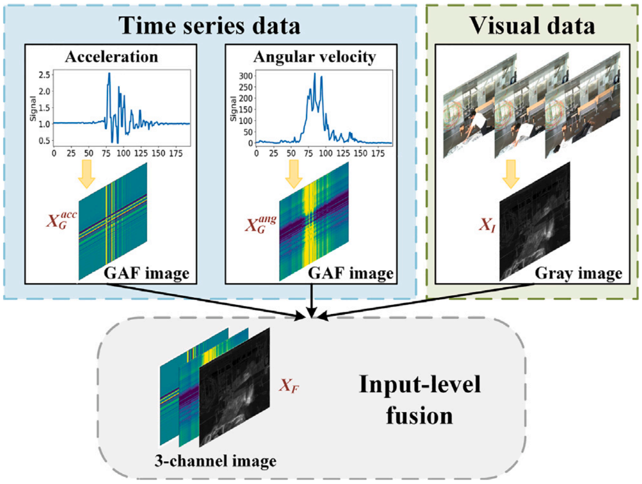


Fig. 6. The flowchart of input-level fusion.

B. *Processing of visual data*

For visual data from the camera, a fall activity is composed of multiple frames of RGB When using visual data from the camera, a fall activity typically consists of multiple frames of RGB images. Using a single frame to represent a fall event can lead to confusion in data labeling. For instance, when a person is transitioning from standing to falling, they may be standing in the first few frames and lying down in the last few frames. Both standing and lying down positions in this activity represent the same fall posture, which can cause subsequent DL models to learn confusing information. To overcome this issue, we employ a visual image stacking approach.

Specifically, given a segment of the raw video with multiple frames $X_V = \{x_1, x_2, \dots, x_n\}$, take the neighboring frames x_i and x_{i+1}, subtract between them to get Δx_i, which represents the change between adjacent frames. Then, all Δx_i are summed to concentrate the information change of one activity into one image. Finally, take the gray image of the image, which is the value of the first channel. In this way, the information about a falling activity is concentrated in a single gray image X_I.

C. *Input-level data fusion*

Input-level data fusion means that the data is fused before it is entered into the DL model. It can avoid the loss of information and also take advantage of the complementarity of multimodal data. The GAF image X_G can be obtained based on the time series data encoding method described above; and the visual data are processed to obtain the gray image X_I. Based on this, the fusion of input levels can be performed.

Specifically, for the wearable sensors, we take the acceleration signal and angular velocity signal to get the acceleration GAF image X_G^{acc} and the angular velocity GAF image X_G^{ang}, respectively. For the visual sensors, we take one of the cameras. On the channels, the three X_G^{acc}, X_G^{ang}, and X_I are stacked to get the 3-channel fusion data X_F. Fig. 6 shows the flowchart of input-level fusion.

$$X_F = [X_G^{acc}, X_G^{ang}, X_I]$$

3.3. Federated learning deployment

In this paper, in order to protect the data privacy of the participating users in the fall detection system, we adopt the paradigm of FL. The specific learning algorithm is FedAvg proposed by McMahan et al. [18]. It is worth stating that the proposed framework is also applicable to other FL algorithms, and we choose the most classical FedAvg for simplicity.

Algorithm 1 shows the flow of FedAvg. The clients are required to contain their locally fused data X_F and their corresponding labels Y. The hyperparameters to be defined are, the number of communicating clients K, the local batch size B, the number of epochs E for the local, and the learning rate η. Specifically, the clients are initialized by a model w_0 at first. In each communication round t, the server randomly selects K clients to be communicated to. This is because it is important to consider that it is not possible for all clients to participate in each round, and some clients will lose their connection due to poor communication. The selected clients perform local training and update the model w_t. Specifically during local training, the local data D is split into batches of size B. The local model performs gradient descent for each batch to perform the update. The updated model is returned by the client to the server. Then, the server averages the received models to update the global model w_{t+1}.

Algorithm 1: FedAvg algorithm.

```
Require: $D = (X_F, Y)$: labeled local dataset; $K$: number of choosed
    clients; $B$: local batch size; $\eta$: learning rate.
initialize $w_0$ at $t = 0$
for each round $t$ do
$S_t \leftarrow$ randomly selected $K$ clients
    for all client $k \in S_t$ do
        split $D$ into batches of size $B$
        for each batch $b \in B$ do
            $w_{t+1}^k \leftarrow w_t - \eta \nabla \ell(w_t; b)$
        end for
    upload $w_{t+1}^k$ to server
    end for
Server executes: $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{1}{S_t} w_{t+1}^k$
end for
```

4. Evaluation experiment

In this section, we show the details of the evaluation experiments. Section 4.1 describes the UP Fall dataset used for the experiments. In Section 4.2, we illustrate the setup of this experiment. Section 4.3 contains details about the FL algorithm and local training model. Sections 4.4 and 4.5 present the results of fall detection and fall activity recognition, with detailed discussions.

4.1. Dataset

In this paper, the UP Fall dataset [9] is used for experimental evaluation. The public dataset contains sensor data from different modalities of the same activity, from wearable sensors, infrared sensors, and cameras, respectively. There are 6 wearable sensors placed on different parts of the body, including the head, ankle, right pocket, belt, neck, and wrist. Sensors located in the head record brain EEG signals, while sensors in other locations record acceleration, angular velocity, and luminosity. 6 pairs of infrared sensors were placed around the experimental site, to record the connection and interruption of the signal generated during the activities. As well, 2 cameras were arranged in front and side of the experimental field to record visual data of the activities. Fig. 7 shows the distribution of sensors. The dataset provider indicated that all signals were aligned on the timestamp at the camera sampling rate, which was 18 Hz.

In the fall experiment, 17 healthy males and females performed 11 different activities, each repeated 3 times (trials). In those activities, Five different postural falls included: (1) falling forward using hands, (2) falling forward using knees, (3) falling backward, (4) falling sideways, and (5) falling sitting in an empty chair. Six non-falling activities included: (6) walking, (7) standing, (8) sitting, (9) picking up an object, (10) jumping, and (11) laying. Fig. 8 shows examples of each activity, and Table 2 lists the duration of each activity in the UP Fall dataset.

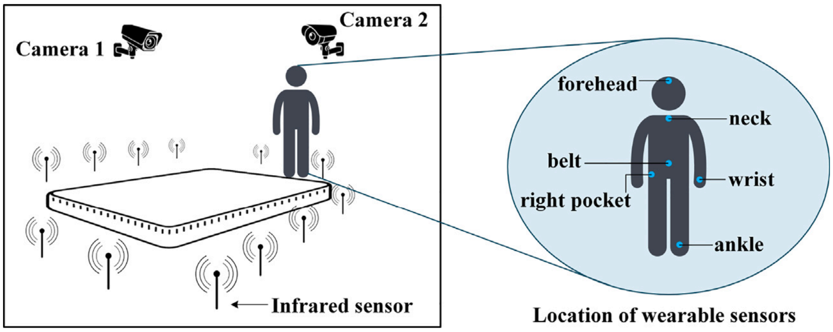


Fig. 7. Distribution of sensors. 2 cameras are located in front and side of the monitoring site; 6 pairs of infrared sensors are arranged around the site; 6 wearable sensors are located in different parts of the human body.

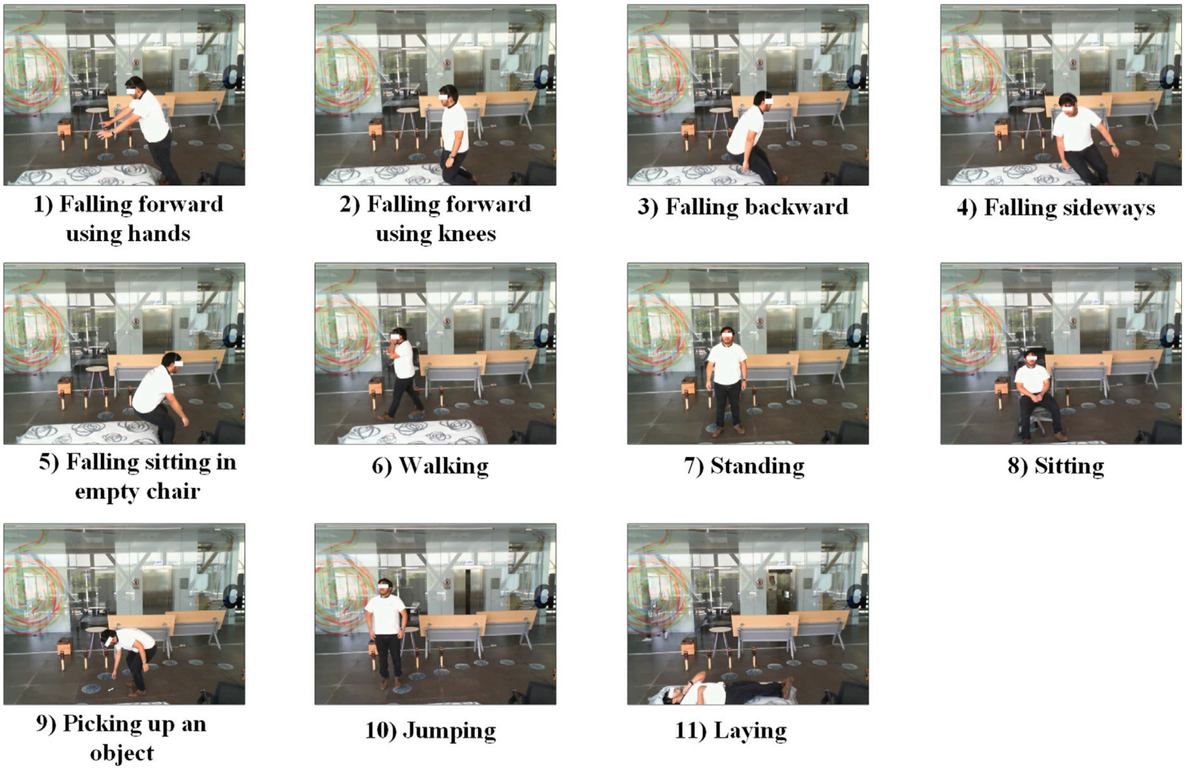


Fig. 8. Examples of each activity in UP Fall dataset [9].

Table 2
Each activity and its description in the UP Fall dataset.

Activity ID	Description	Duration (s)
1	Falling forward using hands	10
2	Falling forward using knees	10
3	Falling backward	10
4	Falling sideways	10
5	Falling sitting in empty chair	10
6	Walking	60
7	Standing	60
8	Sitting	60
9	Picking up an object	10
10	Jumping	30
11	Laying	60

4.2. Experimental setup

Although this dataset contains data from multiple modalities, it is not always realistic to use too many devices just to detect falls. In real life, people tend to carry wearable devices such as smartphones or smartwatches. More commonly, surveillance cameras are installed indoors rather than deploying a large number of infrared sensors. Therefore, our experiments exclude infrared sensor signals and use data from wearable sensors (without head sensors) and cameras. All activities are intercepted to the same according to the shortest activity time. In addition, excluding subjects No. 5 and No. 9, who were missing experimental data, the data of the remaining 15 subjects were excluded. Correspondingly, there are 15 clients in the FL system, and one user represents one client. In each round of model aggregation, 12 clients are randomly selected.

As in the experimental setup of articles [14,41], the data from trials 1 and 2 of each subject were used as the training set and the data from trial 3 were used as the test set. Each sample used for training is a 3-channel image data that is fused into two modalities: time series and visual. The first and second channels are the acceleration and angular velocity recorded by the same sensor, respectively, and the third channel is the visual data from the camera. The specific fusion method is described in Section 3.2.

Two different experimental scenarios are set up in this paper:

(1) *Fall detection*
Five different postures of falls are classified as falls and six daily activities are classified as non-falls. These two classes are marked as 0 and 1, respectively.

(2) *Fall activity recognition*
For 11 different activities, each one is classified as a separate class, labeled from 0 to 10.

To investigate the impact of multimodal data fusion on improving fall detection performance, single-modal data is also provided as input for comparative experiments. Specifically,

(a) TS + C1: Fusion data of time series and Camera 1;
(b) TS + C1: Fusion data of time series and Camera 2;
(c) TS: Time series only;
(d) C1: Camera 1 only;
(e) C2: Camera 2 only;

Where, in (a) and (b), the time series data are fused with Camera 1 and Camera 2, respectively, according to the method proposed in this paper. In (c), the acceleration and angular velocity of the sensor are converted to 2-channel GAF images, and then as input of the local model. For (d) and (e) single frames of RGB from Camera 1 and Camera 2 are taken as inputs, respectively.

4.3. Experimental models

In this section, we describe the specific federal learning algorithms used in the experiments, as well as the training models used for fall detection and fall activity recognition, respectively.

4.3.1. FL algorithm

The Federated learning algorithm used in this paper is FedAvg [18], which is the most basic and practical FL method. The key setup of this FL experiment, each user represents a client who manages its own data independently and does not share it with each other. Each client only performs model transfer with the server. After the clients perform data fusion locally, they train the local model with the fused data. In each round of communication, the server randomly selects 12 clients out of 15 clients for the current round of training. Each of the selected clients downloads the latest current model of the server for initialization and performs gradient updates based on the locally fused data. At the end of the local training round, the clients upload the locally updated gradients to the server separately, and the server averages them. The averaged gradients will be used as the gradients of the global model. After a certain number of communication rounds, the final global model is determined. In the experiment, the number of communication rounds is set based on the difficulty of the classification task, where the number of communication rounds is 100 for fall detection and 200 for fall activity recognition.

4.3.2. Model for local training

One of the important components of federated learning is the local model used for local training. After fusing the multimodal data at the input level according to Section 3.2, a 3-channel image data $X \in R^{C\times H\times W}$ will be generated. The fused data are fed into a CNN-based deep learning model, and after feature extraction, they are fed into the classification for fall prediction.

The experiment in this paper is based on Pytorch. After the data fusion module, the input data is resized to 3 × 70 × 70. The locally trained model for fall detection is shown in Fig. 9. Specifically, a 3-layer CNN module is used for feature extraction. Each convolutional layer has a kernel size of 3 × 3 and a stride of 1. As the convolution depth increases, the number of kernels is 3, 6, and 12, and the activation functions are *Softsign*. After each convolution operation, the maximum pooling operation is performed, and the kernel size of all three pooling layers is 2 × 2 with a stride of 2. To prevent overfitting of the model, *Droupout* and *Batchnormalize* operations are performed after each pooling operation. Finally, the extracted features are fed into the classification layer after flatten, and the cross-entropy loss function is used to calculate the loss of the local model. In addition, the optimization function is stochastic gradient descent (SGD) with a learning rate of *lr* is 0.001. In each round, each local model is trained with an epoch of 3 and batch size of 32.

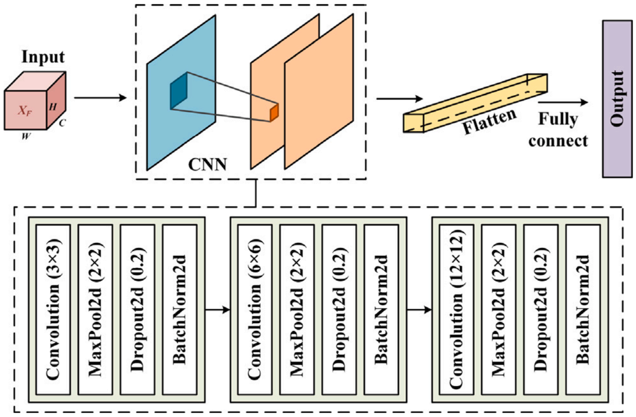


Fig. 9. The model for local training.

Table 3
Accuracy, Precision, Recall, and F1-score for fall detection.

	Accuracy	Precision	Recall	F1
TS + C1	99.867 ± 0.049	99.840 ± 0.070	99.867 ± 0.071	99.854 ± 0.054
TS + C2	99.927 ± 0.041	99.841 ± 0.089	100.00 ± 0.000	99.920 ± 0.045
TS	95.880 ± 0.138	93.938 ± 0.313	97.279 ± 0.293	95.573 ± 0.147
C1	96.147 ± 0.225	94.496 ± 0.369	97.251 ± 0.297	95.847 ± 0.242
C2	97.561 ± 0.197	96.999 ± 0.385	97.706 ± 0.281	97.346 ± 0.210

4.4. Experimental results

This section shows the results of the two experimental scenarios under FL settings in detail, along with a detailed discussion of the results. The classification experiments' metrics are Accuracy, Precision, Recall, and F1 score. Each experiment was performed 10 times and the average was taken as the final result.

4.4.1. Fall detection

For the experimental scenario of fall detection, federated learning aims to jointly train a binary classification model to identify falls and non-falls. The final global model is obtained after 100 rounds of communication between the clients and the server. The test set is used to evaluate the performance of the global model, and Fig. 10 shows the accuracy curves of the global model under 5 different input modes. It can be seen that for the fused data TS + C2, the global model achieves the best accuracy after around 10 communication rounds. When the fused data TS + C1 is input, the best accuracy occurs after about 40 communication rounds. When a single-modal (TS, C1, and C2) is used as input, the best accuracy is lower than the first two. That is, Fewer communication rounds are required to achieve the best accuracy when fused data is used as input. More, by observing the accuracy curve, the curve of fused data (orange and blue line) behaves more flatly and converges faster as the number of communication rounds increases.

The classification results for fall detection are shown in Table 3, in which each value consists of the mean ± standard error. It is obvious that when fused data is used as input, the test accuracy is much higher than that of a single modality. This is consistent with what the accuracy curve presents. The second term is precision, which indicates the correct rate in the samples predicted as falls. the precision of both

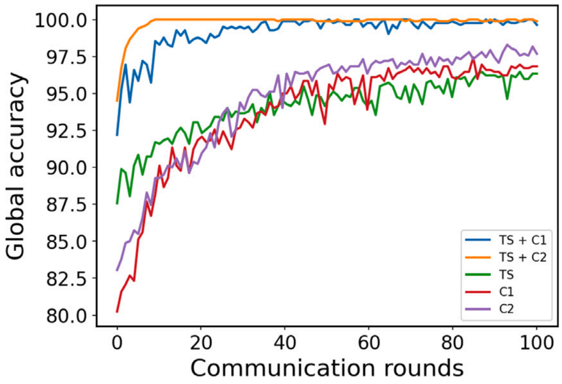


Fig. 10. The accuracy curves of global model for fall detection.

TS + C1 and TS + C2 reach 99.84%. And the precision of TS, C1, and C2 are 93.938%, 94.496%, and 96.999%, respectively. Recall refers to the proportion of samples that are correctly predicted as falls among all the fall samples. Recall plays a very important role as a metric in fall detection. The correct prediction of falls and timely warning can minimize potential life-threatening events. By the table, the recall of TS + C1 is 99.867%, and TS + C2 reaches 100%, and the recall of the remaining three single modalities is all around 97%. Obviously, the former two (TS + C1 and TS + C2) have better performance than the latter three (TS, C1, and C2). Finally, F1 is a combination of accuracy and recall, which indicates the robustness of the model. The F1 of TS + C1 is 99.854% and TS + C2 is 99.920%, while the F1 of TS, C1, and C2 is 95.573%, 95.847%, and 97.346%, respectively. In the four scores, the fusion data has higher expressiveness than the single modality data. This shows that the amount of information carried by the fused data is higher than that of the single modality data. In the training of the local model, taking fusion data as input can enhance the fall detection ability. Additionally, comparisons and analyzes are performed for each mode. The Accuracy, Precision, Recall and F1 of TS + C2 are all slightly higher than TS + C1, which shows that when the wearable sensor data is combined with camera 2 (front view), it brings the best classification results. Similarly, when the data from camera 2 is used as input, its classification results are higher than those of the other two single modalities. This leads to the conclusion that the data from camera 2 plays a more significant role in the experiment.

Correspondingly, Fig. 11 shows the confusion matrix for the best results in fall detection. It can be seen when using fused data for fall detection, almost no classification errors occur in Fig. 11(a) TS + C1 and Fig. 11(b) TS + C2. For single-modal data, the number of misclassifications in Fig. 11(e), (d), and (c) increases sequentially. For TS, C1, and C2, the number of fall samples predicted as non-falls was 10, 7, and 3, respectively. Although classification errors seem to account for a low proportion of the total samples, for real-life fall detection, even an occasional prediction error may have unimaginable consequences. This again shows the importance of fused data for fall detection.

As a conclusion, the experimental results show that the fused data outperformed the single-modal in the binary classification task of fall detection. Specifically, TS has lower accuracy than C1 and C2 when fall detection is performed with a single-modal. This is because the time series data is transformed into image data and then fed into a CNN-based classification model. It is reasonable that the result performance is poorer than the original RGB images. However, when the time series data were fused with Camera 1 or Camera 2 at the input level, the obtained classification accuracy was improved. This also verifies that the data from different modalities have complementary information. When data fusion is performed at the input level, it can help to improve the performance of the classification model.

Table 4
Accuracy, Precision, Recall, and F1-score for fall activity recognition.

	Accuracy	Precision	Recall	F1
TS + C1	89.769 ± 0.162	90.094 ± 0.155	89.769 ± 0.162	89.748 ± 0.162
TS + C2	84.097 ± 0.234	84.546 ± 0.253	84.097 ± 0.234	83.999 ± 0.249
TS	53.085 ± 0.255	52.938 ± 0.450	53.085 ± 0.255	52.286 ± 0.360
C1	70.439 ± 0.325	70.607 ± 0.318	70.439 ± 0.325	70.175 ± 0.329
C2	77.732 ± 0.415	77.777 ± 0.410	77.732 ± 0.415	77.416 ± 0.428

4.4.2. Fall activity recognition

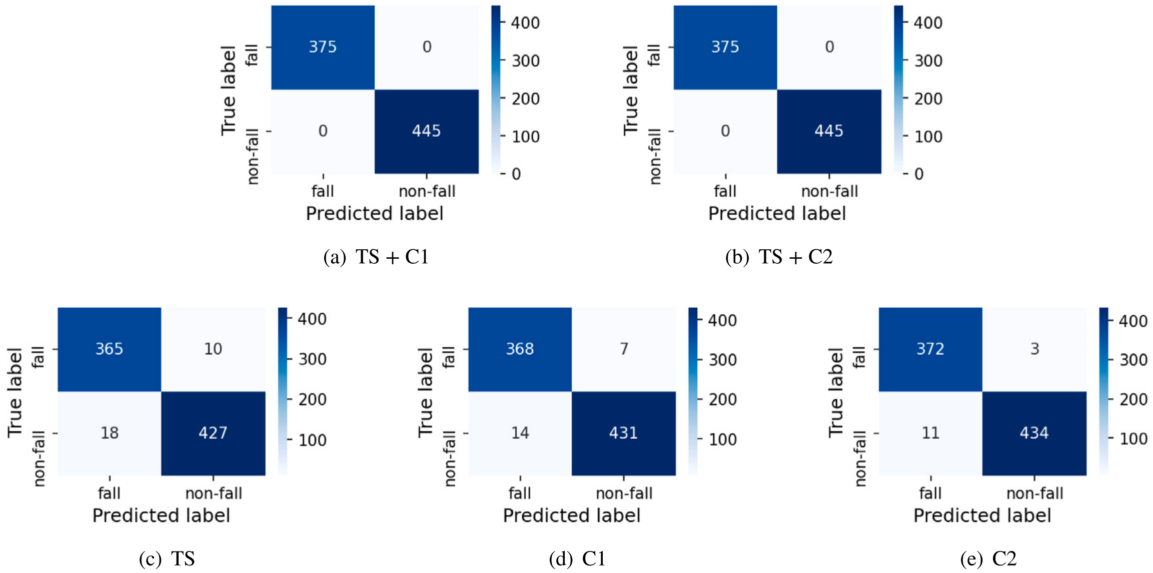
To further evaluate the performance of the proposed FL framework, the experiment was extended to a multi-classification task. In this experiment, each activity of the UP Fall dataset as a class, including the fall activities with different postures and daily activities. Labels A1–A11 are refer to these activities, where, A1: falling forward using hands, A2: falling forward using knees, A3: falling backward, A4: falling sideways, A5: falling sitting in an empty chair, A6: walking, A7: standing, A8: sitting, A9: picking up an object, A10: jumping, A11: laying.

Fig. 12 depicts the accuracy curves of the FL global model as the number of communication rounds increases. Similar to the trend observed in binary classification, the accuracy achieved by fusing data as input is superior to that of using a single-modal as input. Furthermore, the accuracy of TS + C2 is marginally better than that of TS + C1 in the early stages of the model's run. However, after approximately 35 communication rounds, the latter outperforms the former. After about 75 communication rounds, both TS + C1 and TS + C2 (blue and orange lines) reach a steady state, while the steady state of TS, C1, and C2 occurs after about 130 communication rounds. In the multi-classification task, the accuracy difference is more obvious, from high to low are TS + C1, TS + C2, C2, C1, and TS.

Table 4 presents the results for the multi-classification task, each value is the mean ± standard error of 10-time experiments. For accuracy, TS + C1 and TS + C2 are 89.769% and 84.097%, respectively, while TS, C1 and C2 are 53.085%, 70.439% and 77.732%, respectively. It is consistent with the performance of the accuracy curves. Precision plays a more important role in fall activity recognition, which indicates the ability of the model to accurately identify each type of activity. It can be seen from the table that the precision of TS + C1 and TS + C2 are 90.094% and 84.546%, respectively, while the precision of C1, C2, and TS did not exceed 80%. The five input modes have similar performances to precision on Recall and F1-score. The specific values are shown in the Table 4 and will not be repeated here.

In addition, comparing the 4 evaluation metrics in the Figs. 10 and 12, the overall results for multi-class classification are lower than those for binary classification. This is because when the number of training samples and the model structure are consistent, the multi-classification task is obviously more difficult than the binary classification task. The experiment in this section aims to classify each of the 11 activities. Therefore, it is reasonable that the accuracy of the classification results has decreased.

Fig. 13 shows the confusion matrix for fall activity recognition. Fig. 13(a) represents TS + C1, where the recognition error rate for the first 5 fall activities was higher than that for the last 6 daily activities. For example, about 19 samples of A1 are misclassified as A3, 10 samples of A2 are misclassified as A1, and 8 samples of A4 are misclassified as A2. This is due to the similarity of these fall poses, A1 and A2 are respectively falling forward using hands and falling forward using knees, A3 and A4 are falling backward and falling sideways respectively. For activities A6 and A10, no prediction errors occurred in the experiments. In Fig. 13(b), the results for TS + C2 were slightly worse than for TS + C1. Specifically, 20 samples of A1 were misclassified as A2, 13 samples of A2 were misclassified as A1, and 14 samples of A4 were misclassified as A5. Similar to the performance of TS + C1, the model performed well in daily activities (A6–A11) and made fewer mistakes in the distinction. Furthermore, Figs. 13 (c), (d),



(a) TS + C1

(b) TS + C2

(c) TS

(d) C1

(e) C2

Fig. 11. Confusion matrices for fall detection. (a) TS + C1: Fusion data of time series and Camera 1; (b) TS + C1: Fusion data of time series and Camera 2; (c) TS: Time series only; (d) C1: Camera 1 only; (e) C2: Camera 2 only.

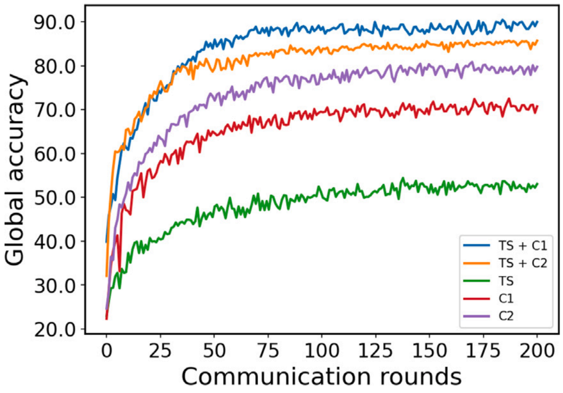


Fig. 12. The accuracy curves of global model for fall activity recognition.

and (e) show the confusion matrices of TS, C1, and C2. Intuitively, these three performed worse in the recognition of different fall activities compared to the fused data TS + C1 and TS + C2. In particular, for time series data TS, more recognition errors were observed even for daily activity recognition (A6–A11). The results of the confusion matrix further demonstrate that when fused data are used as input to the local model, the results outperform those of the single modality.

To summarize, compared to fall detection, the proposed model demonstrates a decrease in performance when recognizing each different fall posture and daily activity. Since when the local model is a simple 3-layer CNN network, the difficulty of multi-classification tasks is higher than that of binary classification tasks. However, it still outperforms single-modal data when using multimodal fusion data.

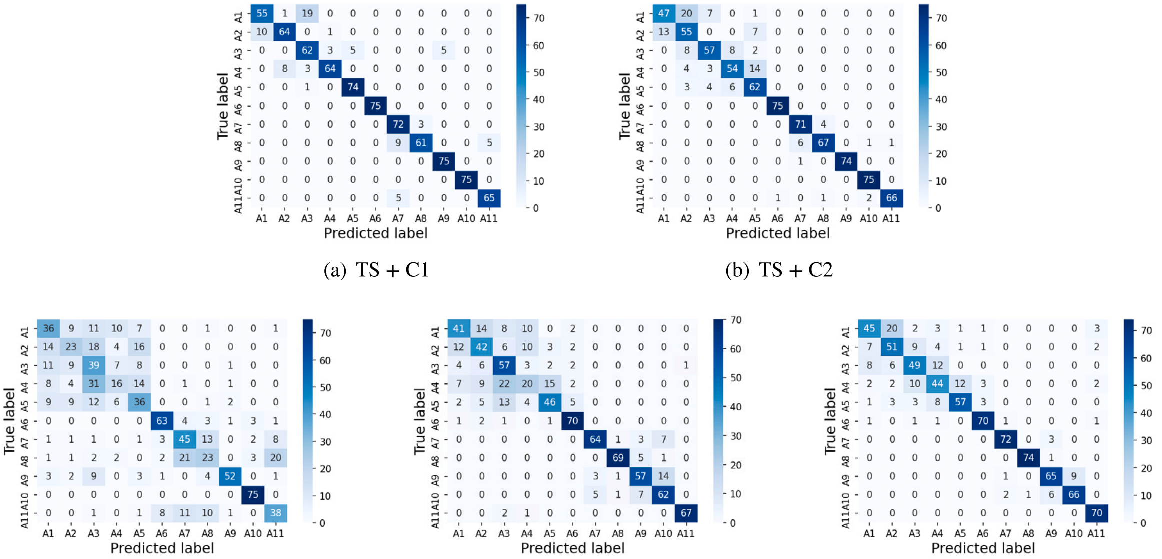
4.5. Discussion

With the two experiments in Section 4.4, it has been demonstrated that fusion of multimodal data is necessary in IoMT. For fall detection and fall activity recognition, data fusion at the input level leads to better classification results. Data fusion at the input level achieves complementary information, and the fused data contains not only visual information but also temporal information. While single-modal data, naturally carry less information than fused data. Moreover, it is inappropriate to use only time series data converted to images for fall detection and fall activity recognition, as it always leads to the lowest classification accuracy. It is worth emphasizing that in the experimental setup of this paper, the local model is based on a convolutional neural network, which is more suitable for image processing tasks rather than time series data. This is evidenced by the higher classification accuracies of C1 and C2 than TS.

In fall detection and fall activity recognition, FL has better privacy protection than data-centralized ML, thanks to its distributed training approach. During FL training, users only need to exchange models instead of uploading their own data to the cloud, which avoids privacy leakage issues. More, FL helps to solve the problem of data silos, which greatly increases the participation of edge users. However, it also poses some challenges when utilizing FL for fall detection. On one hand, frequent communication between the client and the server is required, which will lead to a communication burden once the model is too large, so FL tends to be suitable for less complex models. In this paper, the local training model used is a simple 3-layer CNN, which does not cause communication dilemma. On the other hand, the distribution of data among different clients introduces statistical heterogeneity, which may lead to a model with less accuracy than that of traditional DL models. The above are not only the challenges faced by FL in fall detection, but also the challenges faced by the FL method itself.

5. Conclusion

This paper proposes a FL based multimodal data fusion method for fall detection where users act as independent clients and do not share local data settings with the server or other clients, thus protecting the security of user information. Each client has a data fusion module that takes advantage of the complementary information from heterogeneous sensors, where time series data from wearable sensors are converted into images and then fused with visual data from cameras. A local fall detection model is trained based on the locally fused data, and multiple clients jointly train a global model through FL. The proposed approach improves the recognition accuracy of fall detection through input-level data fusion without exposing user data. The results of two fall-related classification tasks show that the proposed data fusion approach achieves a higher accuracy than that of single-modal data, demonstrating the effectiveness of data fusion for fall detection. Future work will explore the fusion of data from more modalities,



(a) TS + C1

(b) TS + C2

(c) TS

(d) C1

(e) C2

Fig. 13. Confusion matrices for fall activity recognition. (a) TS + C1: Fusion data of time series and Camera 1; (b) TS + C1: Fusion data of time series and Camera 2; (c) TS: Time series only; (d) C1: Camera 1 only; (e) C2: Camera 2 only.

such as environmental sensors, and investigate human activity recognition based on federated multi-task learning where multiple learning tasks are performed simultaneously for different purposes within a privacy-preserving framework. Other possible avenues of research include exploring the use of more advanced machine learning models for fall detection and data fusion or to investigate the transferability of the proposed FL-based multimodal fusion method to other healthcare applications, such as monitoring of chronic diseases or elderly care. Furthermore, the development of more sophisticated privacy-preserving techniques, such as secure multi-party computation and homomorphic encryption, can also be explored to further enhance the security and privacy of user data in FL-based healthcare applications. Finally, the deployment and evaluation of the proposed fall detection system in real-world scenarios can provide valuable insights into its effectiveness and practicality.

CRediT authorship contribution statement

Pian Qi: Conceptualization, Methodology, Data curation, Formal analysis, Writing. **Diletta Chiaro:** Conceptualization, Methodology, Writing, Formal analysis. **Francesco Piccialli:** Supervision, Investigation, Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the following projects:

- PNRR project FAIR - Future AI Research (PE00000013), Spoke 3, under the NRRP MUR program funded by the NextGenerationEU.
- PNRR Centro Nazionale HPC, Big Data e Quantum Computing, (CN_00000013)(CUP: E63C22000980007), under the NRRP MUR program funded by the NextGenerationEU.

References

[1] F. Alshehri, G. Muhammad, A comprehensive survey of the Internet of Things (IoT) and AI-based smart healthcare, IEEE Access 9 (2020) 3660–3678.

[2] M.A. Ahmad, C. Eckert, A. Teredesai, Interpretable machine learning in healthcare, in: Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, 2018, pp. 559–560.

[3] M. Mubashir, L. Shao, L. Seed, A survey on fall detection: Principles and approaches, Neurocomputing 100 (2013) 144–152.

[4] World Health Organization, et al., Step Safely: Strategies for Preventing and Managing Falls Across the Life-Course, World Health Organization, 2021.

[5] H. Ramirez, S.A. Velastin, I. Meza, E. Fabregas, D. Makris, G. Farias, Fall detection and activity recognition using human skeleton features, IEEE Access 9 (2021) 33532–33542.

[6] Q.T. Huynh, U.D. Nguyen, L.B. Irazabal, N. Ghassemian, B.Q. Tran, Optimization of an accelerometer and gyroscope-based fall detection algorithm, J. Sens. 2015 (2015).

[7] V. Mohan Gowda, M.P. Arakeri, V. Raghu Ram Prasad, et al., Multimodal classification technique for fall detection of alzheimer's patients by integration of a novel piezoelectric crystal accelerometer and aluminum gyroscope with vision data, Adv. Mater. Sci. Eng. 2022 (2022).

[8] 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, Inf. Fusion 76 (2021) 355–375.

[9] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, C. Peñafort-Asturiano, UP-fall detection dataset: A multimodal approach, Sensors 19 (9) (2019) 1988.

[10] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.

[11] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, B. Scholkopf, Support vector machines, IEEE Intell. Syst. Appl. 13 (4) (1998) 18–28.

[12] M.W. Gardner, S. Dorling, Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences, Atmos. Environ. 32 (14–15) (1998) 2627–2636.

[13] M.-L. Zhang, Z.-H. Zhou, ML-KNN: A lazy learning approach to multi-label learning, Pattern Recognit. 40 (7) (2007) 2038–2048.

[14] M.M. Islam, S. Nooruddin, F. Karray, G. Muhammad, Multi-level feature fusion for multimodal human activity recognition in Internet of Healthcare Things, Inf. Fusion (2023).

[15] L. Martínez-Villaseñor, H. Ponce, K. Perez-Daniel, Deep learning for multimodal fall detection, in: 2019 IEEE International Conference on Systems, Man and Cybernetics, SMC, IEEE, 2019, pp. 3422–3429.

[16] T. Vaiyapuri, A. Binbusayyis, V. Varadarajan, Security, privacy and trust in IoMT enabled smart healthcare system: a systematic review of current and future trends, Int. J. Adv. Comput. Sci. Appl. 12 (2) (2021).

[17] L. Li, Y. Fan, M. Tse, K.-Y. Lin, A review of applications in federated learning, Comput. Ind. Eng. 149 (2020) 106854.

[18] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.

[19] E. Auvinet, F. Multon, A. Saint-Arnaud, J. Rousseau, J. Meunier, Fall detection with multiple cameras: An occlusion-resistant method based on 3-D silhouette vertical distribution, IEEE Trans. Inf. Technol. Biomed. 15 (2) (2010) 290–300.

[20] L. Xie, Y. Yang, F. Zeyu, S.M. Naqvi, Skeleton-based fall events classification with data fusion, in: 2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI, IEEE, 2021, pp. 1–6.

[21] W.-Y. Cai, J.-H. Guo, M.-Y. Zhang, Z.-X. Ruan, X.-C. Zheng, S.-S. Lv, GBDT-based fall detection with comprehensive data from posture sensor and human skeleton extraction, J. Healthc. Eng. 2020 (2020).

[22] H. Li, A. Shrestha, H. Heidari, J. Le Kernec, F. Fioranelli, Bi-LSTM network for multimodal continuous human activity recognition and fall detection, IEEE Sens. J. 20 (3) (2019) 1191–1201.

[23] T. Xu, H. Se, J. Liu, A fusion fall detection algorithm combining threshold-based method and convolutional neural network, Microprocess. Microsyst. 82 (2021) 103828.

[24] Y. Yang, H. Yang, Z. Liu, Y. Yuan, X. Guan, Fall detection system based on infrared array sensor and multi-dimensional feature fusion, Measurement 192 (2022) 110870.

[25] Y. Yao, C. Liu, H. Zhang, B. Yan, P. Jian, P. Wang, L. Du, X. Chen, B. Han, Z. Fang, Fall detection system using millimeter-wave radar based on neural network and information fusion, IEEE Internet Things J. 9 (21) (2022) 21038–21050.

[26] M. Amsaprabhaa, et al., Multimodal spatiotemporal skeletal kinematic gait feature fusion for vision-based fall detection, Expert Syst. Appl. 212 (2023) 118681.

[27] W.-J. Yi, O. Sarkar, S. Mathavan, J. Saniie, Wearable sensor data fusion for remote health assessment and fall detection, in: IEEE International Conference on Electro/Information Technology, IEEE, 2014, pp. 303–307.

[28] A. De, A. Saha, P. Kumar, G. Pal, Fall detection method based on spatio-temporal feature fusion using combined two-channel classification, Multimedia Tools Appl. 81 (18) (2022) 26081–26100.

[29] M. Ali, F. Naeem, M. Tariq, G. Kaddoum, Federated learning for privacy preservation in smart healthcare systems: A comprehensive survey, IEEE J. Biomed. Health Inf. (2022).

[30] Y. Chen, X. Qin, J. Wang, C. Yu, W. Gao, Fedhealth: A federated transfer learning framework for wearable healthcare, IEEE Intell. Syst. 35 (4) (2020) 83–93.

[31] J. Ma, S.-A. Naas, S. Sigg, X. Lyu, Privacy-preserving federated learning based on multi-key homomorphic encryption, Int. J. Intell. Syst. 37 (9) (2022) 5880–5901.

[32] A.R. Khan, S.M. Bokhari, S. Sohaib, O. Popoola, K. Arshad, K. Assaleh, M.A. Imran, A. Zoha, Federated learning based non-invasive human activity recognition using channel state information, Available at SSRN 4395564.

[33] Z. Yu, J. Liu, M. Yang, Y. Cheng, J. Hu, X. Li, An elderly fall detection method based on federated learning and extreme learning machine (Fed-ELM), IEEE Access 10 (2022) 130816–130824.

[34] M.A. Rahman, M.S. Hossain, M.S. Islam, N.A. Alrajeh, G. Muhammad, Secure and provenance enhanced internet of health things framework: A blockchain managed federated learning approach, IEEE Access 8 (2020) 205071–205087.

[35] K. Arikumar, S.B. Prathiba, M. Alazab, T.R. Gadekallu, S. Pandya, J.M. Khan, R.S. Moorthy, FL-PMI: federated learning-based person movement identification through wearable devices in smart healthcare systems, Sensors 22 (4) (2022) 1377.

[36] Y. Zhao, P. Barnaghi, H. Haddadi, Multimodal federated learning on IoT data, in: 2022 IEEE/ACM Seventh International Conference on Internet-of-Things Design and Implementation, IoTDI, IEEE, 2022, pp. 43–54.

[37] H. Han, C. Lian, Z. Zeng, B. Xu, J. Zang, C. Xue, Multimodal multi-instance learning for long-term ECG classification, Knowl.-Based Syst. (2023) 110555.

[38] M. Wang, W. Wang, X. Zhang, H.H.-C. Iu, A new fault diagnosis of rolling bearing based on Markov transition field and CNN, Entropy 24 (6) (2022) 751.

[39] Z. Ahmad, A. Tabassum, L. Guan, N. Khan, Ecg heart-beat classification using multimodal image fusion, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2021, pp. 1330–1334.

[40] Z. Wang, T. Oates, et al., Encoding time series as images for visual inspection and classification using tiled convolutional neural networks, in: Workshops At the Twenty-Ninth AAAI Conference on Artificial Intelligence, Vol. 1, AAAI Menlo Park, CA, USA, 2015.

[41] R. Espinosa, H. Ponce, S. Gutiérrez, L. Martínez-Villaseñor, J. Brieva, E. Moya-Albor, A vision-based approach for fall detection using multiple cameras and convolutional neural networks: A case study using the UP-fall detection dataset, Comput. Biol. Med. 115 (2019) 103520.