computer sciences and mathematics forum

MDPI

Proceeding Paper

Heterogeneous Federated Learning Model for Recognizing Human Activity †

Nwadher S. Alblihed and Dina M. Ibrahim *

Department of Information Technology, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia; n.alblihed@qu.edu.sa

* Correspondence: d.hussein@qu.edu.sa

† Presented at the 1st International Conference on Emerging Tech & Innovation (ICETI), Buraydah, Saudi Arabia, 10 February 2026.

Abstract

A range of sensors are used by human activity recognition (HAR) to identify the activities that people complete each day. The recognition of human activities has benefited greatly from machine learning (ML), as it has made many human activities more easily recorded. Unfortunately, a centralized approach is used in many HAR applications, which might compromise user privacy. One must use deep learning (DL) using different algorithms and models to analyze the data generated from ML. Another kind of ML is distributed ML, called federated learning (FL), which tries to distribute ML models across edge devices. Thus, this study presents an FL model to support HAR by building a generic model and using user-based training data without data sharing. Through developing heterogeneous local models, each client takes the most suitable DL model to the client. This study uses three different DL models to develop the local model: Convolutional Neural Network (CNN), Residual Network (ResNet), and Long Short-term Memory (LSTM). Moreover, different numbers of clients are experimented with: two, five, and ten clients. The UniMiB SHAR dataset is used to apply the experiments. As a result, using five clients with three mixed DL models gives the highest Accuracy of 90.8%.

Keywords: federatedlearning; human activities; human activity recognition; machine learning; distributed machine learning; heterogeneous models

1. Introduction

Human activity recognition (HAR) includes various activities such as walking, running, sitting, standing, sleeping, driving, cooking, and other regular or irregular activities. According to [1], activity recognition is the process of recording and identifying activity using data that has not been processed and is obtained from a variety of sensors, cameras, wearable technologies, mobile devices, and sensors placed in surroundings.

The HAR is a more popular topic because of the availability of sensors, its low cost and energy usage, and advancements in machine learning (ML) techniques and the Internet of Things (IoT) to complement each other. By monitoring, controlling medication, reducing social isolation, and improving nursing productivity, these solutions may enhance technology's role in the quality of older people's daily lives [2]. Another essential point is the Department of Economic and Social Affairs of the United Nations' expectations regarding the worldwide population; in 2020, citizens aged 15 to 64 had a ratio of 7:1, while in the future year of 2050, this will reduce to approximately 4:1 [3].

Check for updates

Academic Editors: Ali Mustafa Qamar and Rehan Ullah Khan

Published: 16 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

ML is part of Artificial Intelligence (AI) and is considered one of the methods that may support the processing of HAR data and designing models to facilitate operations between the elderly and care centers. The central approach of ML has multiple HAR applications. Thanks to advanced technology, the concept of distributed ML, which is federated learning (FL), has been extracted. FL is used to train ML with a decentralized approach: collaborating between edge devices during the training process of local data without transferring or capturing personal data. Therefore, this study introduces FL models to decentralize the HAR operations, aiming to assist elderly people in daily life.

With regard to other studies, they have used just one distribution of clients for their federated learning process, and there needs to be a further implementation of the FL approach using different numbers of clients. Therefore, this study applied FL to diverse numbers of clients, namely two, five, and ten.

The University of Milano Bicocca Smartphone-based HAR (UniMiB SHAR) [4] dataset includes subjects of ages suitable for any study on HAR. Additionally, previous studies have used different deep learning models to apply FL approaches, but none have used the Residual Network (ResNet) model. These factors contribute to this study using three deep learning models: Convolutional Neural Network (CNN), ResNet, and Long Short-term Memory (LSTM).

From the previous findings, we can summarize the gaps:

- Most datasets contain daily activities without fall detection.
- There is a shortage of previous studies that use different numbers of clients when applying FL.
- There is a lack of an attempt to utilize heterogeneous FL.

In this paper, we use another method to apply federated learning strategies. The central concept is that each client selects different deep learning model techniques based on Accuracy results, which produces heterogeneous local model clients. To clarify the plan for our work, Figure 1 illustrates a mind map of each of the three categories of the heterogeneous experiment: two, five, and ten clients. Each client has undertaken training and results are introduced by experiments on the deep learning models.

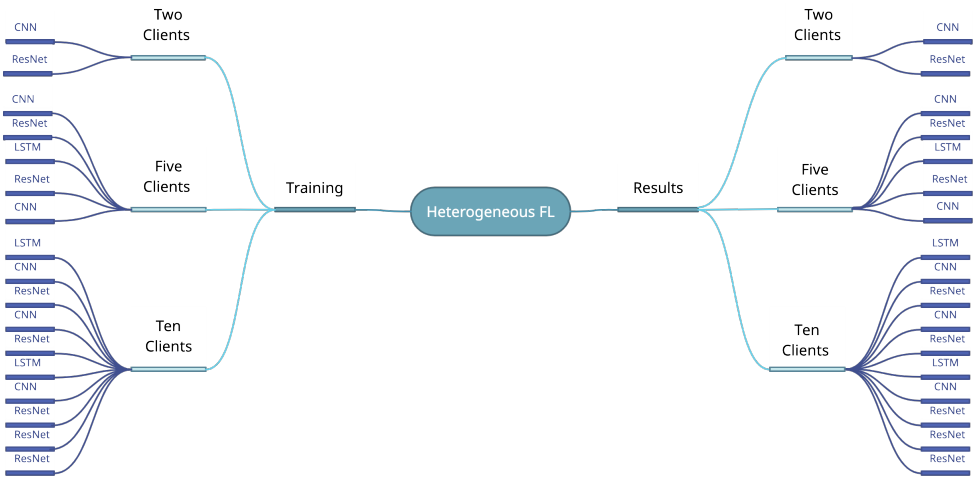


Figure 1. A mind map illustrating the distribution of experiments in this study.

In addition to the current section, the remainder of this study's sections are divided into: (1) Literature Review, (2) Methodology, (3) Implementation Experiments, (4) Discussion, and the last section (5) Conclusions.

https://doi.org/10.3390/cmsf2026013010

2. Literature Review

2.1. Human Activity Recognition Studies

The use of different machine learning algorithms in the implementation of HAR has increased, including those based on deep learning [5,6]. Deep learning is used to improve the Accuracy of HAR and remove the need for feature extraction work to be performed by humans [7–9]. Various studies have benefited from the use of deep learning in HAR applications. These studies have combined a Deep Boltzmann Machine and Multimodals (DBMs) to improve the performance of a HAR task [7,9]. Additionally, to handle background noise in the environment, a Deep Ear [10] model has been proposed to manage the audio sensing functions carried out using DBMs.

Moreover, HAR can be used in different scenarios and in different areas, especially in patients' lives, for instance, in assisted living [11], ambient intelligence [12,13], and urban mobility management [14]. Specifically, to assess the quality of elderly patients' environments, ambient sensors can perform behavioral analyses to discover their living patterns [15]. Thus, as it offers management decisions for the efficient healthcare of the elderly, technology that recognizes HAR data is called AAL [16]. For example, a model for smart home activity recognition based on AAL and using four different machine learning methods was proposed in [17]. Ghadi et al. developed an AAL model using two different deep learning classifiers [18].

The key aspects in this area are applying centralized methods, collecting data, and training the model on a central server, which raise privacy issues and require the sharing of user data. Regardless, the models developed using deep learning have improved performances; however, the training process may be insufficient on central devices [19].

2.2. Federated Learning Studies

The distribution of ML via FL began in 2016, where FL was used as an alternative to traditional deep learning techniques [20,21]. FL works through several clients collaborating to learn and train the global model on a cloud server. The server sends the global model to all selected clients, and each client's model is trained using their local data to update the global model without sharing the clients' data [21,22]. When taking an average of the clients' local models to train a global model, the method is called FedAvg, as used in [20,23]. Thus, several studies have benefited from FL, and its use has been demonstrated in the fields of healthcare [24], monitoring [25], and finance [26].

Google was the first to build a large-scale FL platform, allowing tens of millions of users to collaborate in training a next-word prediction model without revealing their data [23]. Moreover, Wu et al. [27] offered a personalized FL model for deployment in smart IoT applications. A distributed FL system was constructed by Khan et al. to build an integer linear optimization [28]. A scheme introduced by Oh et al. [29] was a federated-learning-based communication system efficient at preserving privacy using the MNIST dataset. To solve the issue of restricted datasets, Wang et al. [30] suggested federated transfer learning, which merges FL with transfer learning. Ultimately, the benefits of this approach include its capacity to function on isolated data, its emphasis on data privacy, and its ability to learn customized models [31].

2.3. Human Activity Recognition with Federated Learning Studies

Sozinov et al. [32] evaluated an FL performance model whose central model was trained using HAR. The experiment illustrated an 89% Accuracy using FL and 93% using a centralized method. Likewise, they assessed a CNN model trained using centralized approaches and three different FL algorithms, whereas Sannara et al. evaluated an FL model with semi-supervised learning [33].

FL was applied to person-movement identification by the authors of [34] in order to overcome smart healthcare's shortcomings. Similar to this, mobile activity monitoring was used in the study [35]. Two more research works have focused on health monitoring: Tran et al. used FL [36] to implement HAR. In contrast, a FedHome model that performs in-home health monitoring was developed by Qiong et al. [37].

The clustering method was used by Presotto et al. [38] to handle non-IID problems. A distributed HAR used the FL approach, which carried out a Federated Aggregate (FA) through implementing clients' devices as the basis of the infrastructure [39]. Federated Inverse Averaging (FedInAvg) for HAR was developed by [40]. A Graph Neural Network used FL for a HAR (GraFeHTy) algorithm to eliminate unlabeled or noisy data [41].

A wearable device with an FL system (FL4W) was suggested in [42] for classifying human activities. Its methodology had four steps. First, it registered a wearable device to enter it into collaborative learning. Second, the global model was broadcast to all clients' wearable devices. Third, the local model was trained on a physical workout. Finally, the global model was aggregated using the FedAvg algorithm. The authors used the Daily and Sports Activities (DSA) dataset, which contains data collected from eight subjects and ten daily sports activities using a motion sensor. As a result, their model achieved a 95% accurate result, a satisfactory result, but not truly sufficient given the limited number of subjects in the dataset.

Chenglin et al. developed a federated repreparation learning framework that was meta-trained on embedding networks using a federated approach. Their work aimed to generate a personalized HAR model for every user based on each user's activities (Meta-HAR). To predict user activities, the signal from the embedding network's repreparation participated in developing a personalized classification network. Thus, they trained a single model on a range of related learning tasks with a limited number of samples. Afterward, this model could handle each user's other tasks as separate tasks, solving the activity recognition problem. The model was trained on two different datasets. The first was Heterogeneous Human Activity Recognition (HHAR), which contained nine subjects completing six activities: standing, sitting, walking, going up stairs, going down stairs, and biking. The other dataset was USC-HAD, which included 14 subjects completing the same activities as the HHAR dataset but, instead of biking, it used running.

On the other hand, the FedHAR model suggested by Bettini et al. was a combination of FL and semi-supervised approaches to HAR. For illustrative purposes, active learning with label propagation was used to categorize the sensor's unlabeled data. The FedHAR model used transfer learning techniques to personalize the global model for each user. It used two public datasets. The first one, MobiAct, contained 60 subjects and covered sitting, standing, walking, jogging, and jumping activities. The second one was the WISDM dataset, which contained 36 subjects completing five activities which were the same as the activities in MobiAct but, instead of jumping, it used taking stairs. The FedHAR model was evaluated with and without active learning and label propagation. As a result, we can observe that the recognition rate of human activities was improved by a small number of labels being assigned to both datasets by users. Compared with the FedAvg and FedHealth methods of FL, which are based on completely labeled data, the FedHAR model was better than FedAvg for just the WISDM dataset, performing with an F1-score of 87.5%, whereas FedHealth outperformed it by 3% on both datasets [43]. The work was distinguished and presented many techniques; however, the study results were not analyzed in sufficient depth.

3. Methodology

This study is based on a methodology designed to eliminate the implementation shortcomings of the FL approach to HAR and achieve the objectives. It includes six

phases: (1) global model construction, (2) data preparation, (3) building local models, (4) aggregating and averaging local models, (5) updating and distributing global models, and (6) communication results.

The methodology features heterogeneous local models for each client; each local model has a distinct color from the rest, as demonstrated in Figure 2. In this study, each client is trained and tested on different models.

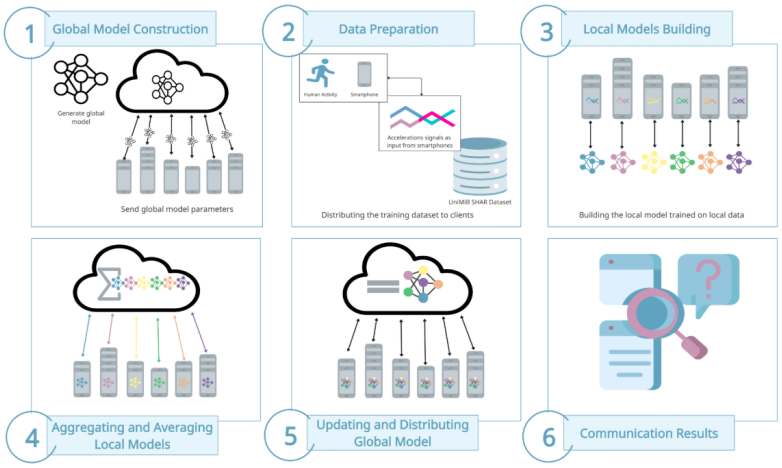


Figure 2. The six phases of the heterogeneous methodology.

Communication Results

A large percentage of studies on FL use two well-known performance metrics, Accuracy and the Loss function, to evaluate the proposed model's performance. Thus, we will use these two metrics to evaluate our proposed model. The method for calculating Accuracy is given in Equation (1), while the Loss function is derived from an equation in FedAvg, as shown in Equations (2) and (3), where T denotes True, F indicates False, P means Positive, and N means Negative.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

$$\max_{w} F(w), \text{where } F(w) = \sum_{k=1}^{m} \frac{n_k}{n} F_k(w) \tag{2}$$

where the Loss function is

$$F_k(w) = \frac{1}{n_k} \sum_{i \in D_k} f_i(w) \tag{3}$$

4. Implementation Experiments

4.1. The Dataset Chosen

The University of Milano Bicocca Smartphone-based HAR (UniMiB SHAR) [4] is the dataset selected for this study, and it will be used to train the FL model. The Samsung Galaxy Nexus I9250 smartphone's 3D accelerometer is used to collect 11,700 acceleration samples for the dataset. Thirty subjects are sampled; six are male and 24 are female, with ages ranging from eighteen to sixty.

4.2. The Models Chosen

4.2.1. CNN

The CNN model, which is trained using two dimensions, has three layers. We employed the 2D convolution layer. BatchNorm2d was used to achieve the normalization

technique between the layers. Furthermore, ReLU, a non-linear activation function commonly used in multilayer neural networks, was applied to the CNN model.

4.2.2. ResNet

The basic block has two layers of 3x3 convolution: the first layer has Conv2d, BatchNorm2d, and ReLU, while the second layer only has Conv2d and BatchNorm2d.

4.2.3. LSTM

Due to the UniMib SHAR dataset, the data form was an array. We used the 512-neuron LSTM model to train the data.

4.3. Training and Evaluation with Two Heterogeneous Clients

We concluded that the mixing process introduced a higher average, and the results were based on Accuracy. As the Accuracy of the models mattered more in the study than the Loss function, we took the best results based on the Accuracy outcomes of each experiment and then mixed the local models of the two clients. After many experiments were conducted on the mixing processes, the better combination of the two clients is described in the below sections.

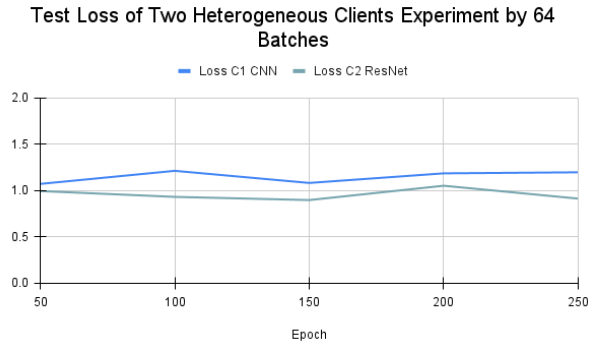
4.4. Results for Training and Evaluation with Two Heterogeneous Clients

The central theme of our work was heterogeneous. Therefore, the experiment that achieved the best results for each client was accepted individually. As ResNet attained better results than the others in this experiment, the CNN model for the first client achieved results that were more useful than those for the second client. Thus, the local model that was accepted for the first client was CNN, while the model accepted for the second client was ResNet. Table 1 presents the outcomes for the first client using CNN and those of the second client using ResNet.

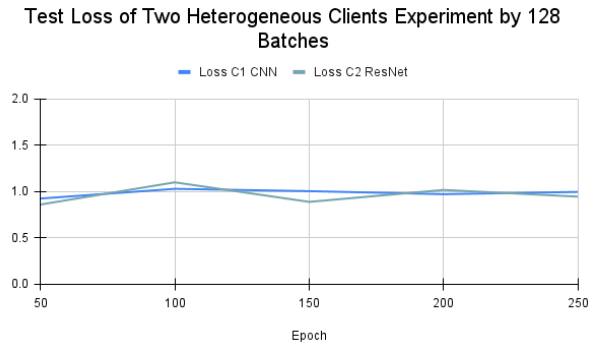
Table 1. The results of the experiment with two heterogeneous clients.

		First Client CNN		**Second Client ResNet**	
Epoch	**Batch**	**Loss**	**Acc**	**Loss**	**Acc**
50	64	1.0726	0.8826	0.9951	0.8855
	128	0.9252	0.8772	0.8604	0.8775
	256	0.9148	0.8747	0.8608	0.8651
	512	0.8232	0.8806	0.7594	0.8716
100	64	1.2136	0.8861	0.9326	0.8875
	128	1.03	0.8777	1.1003	0.882
	256	0.9088	0.8777	0.8803	0.8726
	512	0.8603	0.8752	0.7739	0.8711
150	64	1.0829	0.8816	0.8981	0.8914
	128	1.0052	0.8752	0.8894	0.8726
	256	0.9253	0.8777	0.8614	0.8731
	512	0.8231	0.8707	0.8186	0.8691
200	64	1.1865	0.8767	1.0528	0.8775
	128	0.9723	0.8737	1.0183	0.882
	256	0.9518	0.8762	0.8475	0.8632
	512	0.8502	0.8752	0.7931	0.8681
250	64	1.1977	0.8856	0.9138	0.8845
	128	0.9969	0.8831	0.9459	0.8731
	256	1.0332	0.8777	0.9012	0.8716
	512	0.8278	0.8787	0.8435	0.8726

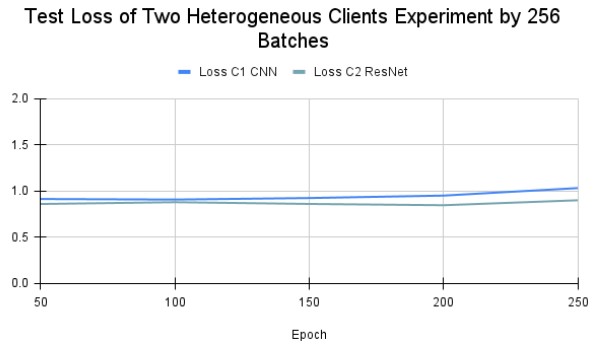
In another way, Figure 3 reveals the Loss function for the heterogeneous experiment with two clients. The Accuracy of the experiment is provided in Figure 4.



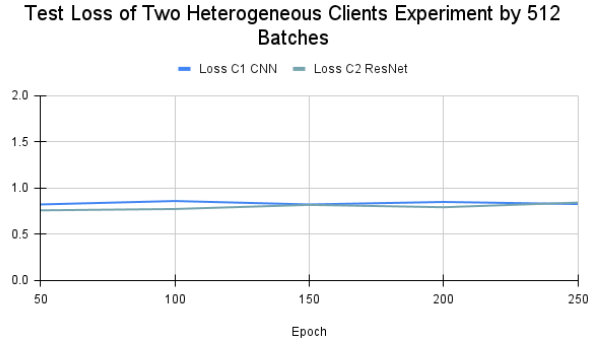
(**a**) 64 Batches



(**b**) 128 Batches

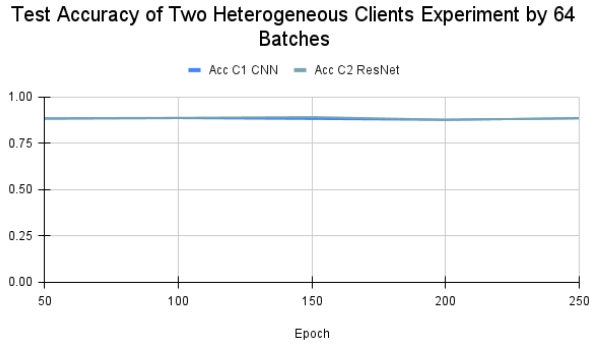


(**c**) 256 Batches

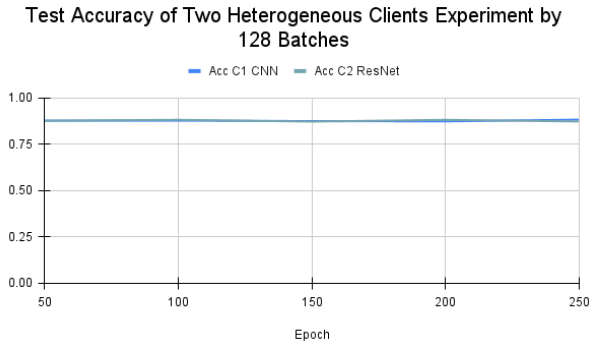


(**d**) 512 Batches

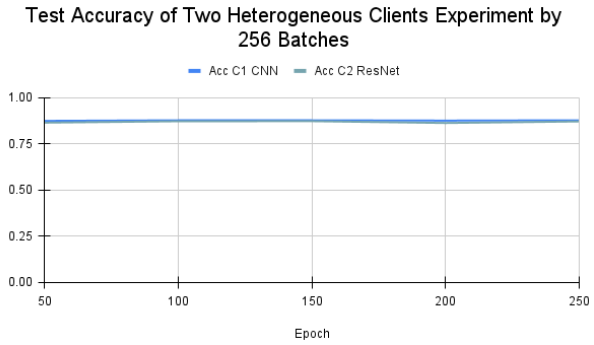
Figure 3. Loss function of the experiment with two heterogeneous clients with a fixed batch size and different epochs.



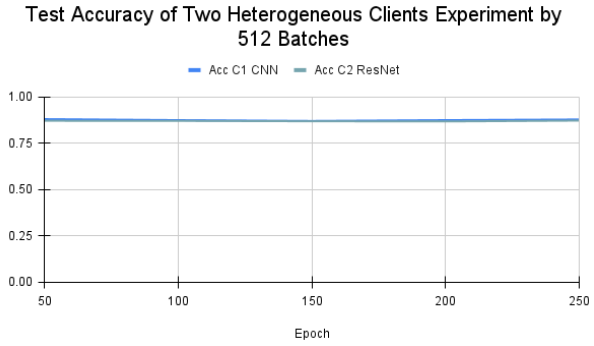
(**a**) 64 Batches



(**b**) 128 Batches



(**c**) 256 Batches



(**d**) 512 Batches

Figure 4. Accuracy of the experiment with two heterogeneous clients with a fixed batch size and different epochs.

Averaging and Updating

To obtain the average values for the heterogeneous experiment, the results were divided by 2. The average Loss was 0.7912, but the average Accuracy was 0.8251. Table 2

presents the significant outcomes and the average results. The process of updating the global model was based on the averages of Loss and Accuracy.

Table 2. The main results of the experiment with two heterogeneous clients.

	Low Loss	High Acc	Avg Loss	Avg Acc
First Client CNN	0.8231	0.8861	0.7912	0.8251
Second Client ResNet	0.7594	0.8914		

4.5. Training and Evaluation with Five Heterogeneous Clients

The models developed using two clients' experiences were used to experiment with five heterogeneous clients. We selected these models based on the best Accuracy for each client. For example, the LSTM achieved the best outcomes with the third client, the ResNet with the second and fourth clients, and the CNN with the first and fifth clients.

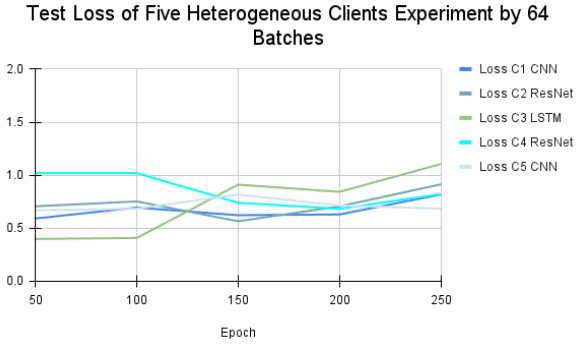
4.6. Results for Training and Evaluation with Five Heterogeneous Clients

The different local models used in this experiment were CNN for the first client, ResNet for the second client, LSTM for the third client, ResNet for the fourth client, and CNN for the fifth client. Table 3 presents the results for each model separately.

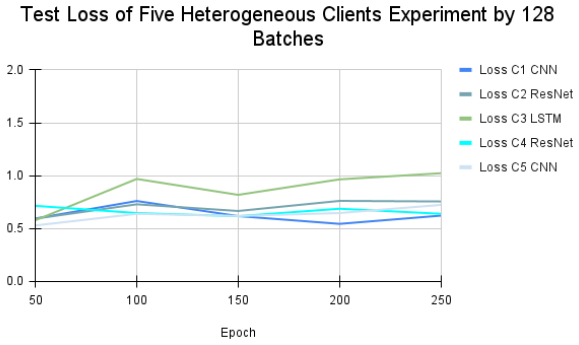
Table 3. The results of the experiment with five heterogeneous clients.

		First Client CNN		Second Client ResNet		Third Client LSTM		Fourth Client ResNet		Fifth Client CNN	
Epoch	Batch	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc
50	64	0.5915	0.9045	0.7066	0.902	0.3988	0.8759	1.0199	0.8945	0.6689	0.897
	128	0.5954	0.8995	0.5939	0.9107	0.5786	0.7605	0.7152	0.9082	0.5296	0.9144
	256	0.4113	0.9144	0.7099	0.8933	0.6278	0.7804	0.7867	0.8933	0.4844	0.8995
	512	0.4042	0.8958	0.5762	0.8921	0.7393	0.7568	0.5287	0.9119	0.4188	0.9032
100	64	0.6951	0.9194	0.7526	0.9169	0.4083	0.8734	1.02	0.9181	0.6845	0.8908
	128	0.7608	0.8908	0.7295	0.9007	0.9693	0.7866	0.6468	0.9094	0.6394	0.8983
	256	0.4991	0.8933	0.6676	0.9032	0.7277	0.773	0.3614	0.9243	0.5317	0.9181
	512	0.5202	0.8859	0.6226	0.9082	0.6688	0.763	0.6425	0.9032	0.4939	0.8945
150	64	0.6226	0.9069	0.5658	0.9181	0.9107	0.8226	0.74	0.928	0.8176	0.8983
	128	0.6182	0.8921	0.6663	0.9107	0.8189	0.7543	0.6174	0.9082	0.6194	0.897
	256	0.6031	0.9057	0.674	0.8846	0.6856	0.7878	0.6783	0.9094	0.7159	0.9256
	512	0.4258	0.902	0.6102	0.8821	0.6713	0.7643	0.5972	0.897	0.5012	0.9057
200	64	0.6301	0.9107	0.7061	0.9032	0.8432	0.8337	0.6832	0.9069	0.7141	0.8896
	128	0.5447	0.9169	0.7619	0.9194	0.9654	0.7891	0.6877	0.9119	0.647	0.902
	256	0.5901	0.8834	0.4372	0.9243	0.6579	0.7854	0.635	0.9156	0.6354	0.897
	512	0.4338	0.9069	0.5076	0.9082	0.6565	0.773	0.5455	0.9032	0.5362	0.8871
250	64	0.8184	0.9082	0.9142	0.9119	1.1054	0.8089	0.822	0.9144	0.6855	0.902
	128	0.6227	0.9181	0.7565	0.8995	1.0234	0.7742	0.6399	0.9243	0.7239	0.8883
	256	0.6899	0.8846	0.387	0.9256	0.5324	0.7916	0.4894	0.9107	0.4884	0.9045
	512	0.4354	0.9007	0.4314	0.9181	0.7064	0.7618	0.4139	0.9256	0.5129	0.8945

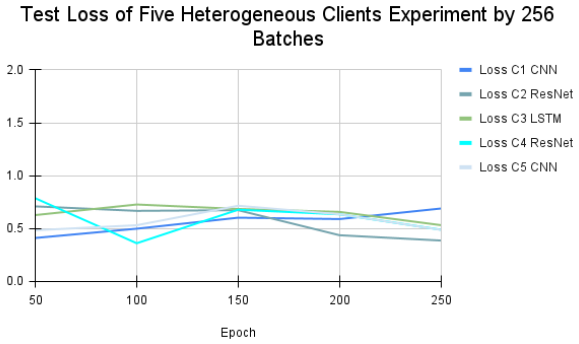
To better visualize the results, Figure 5 shows the line charts for the Loss functions of the five local models of different clients constructed using different deep learning model techniques. Likewise, Figure 6 presents the line charts for the Accuracy measure.



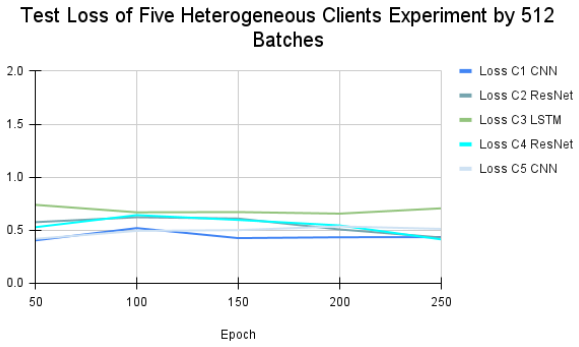
(**a**) 64 Batches



(**b**) 128 Batches

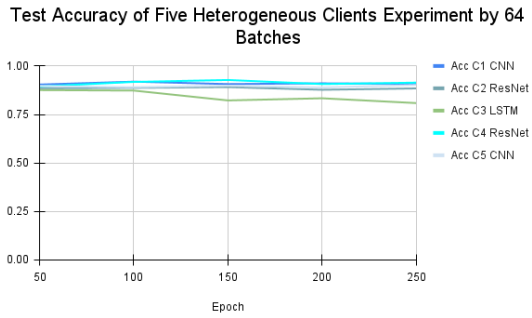


(**c**) 256 Batches

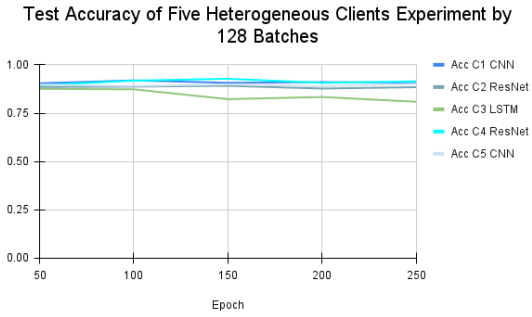


(**d**) 512 Batches

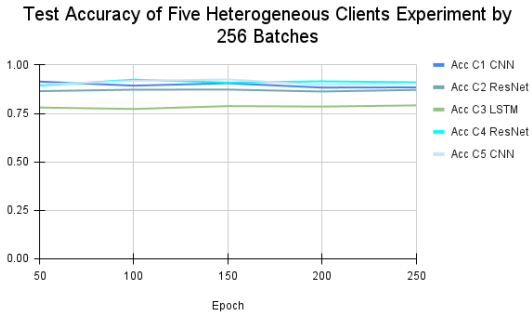
Figure 5. Loss function of the experiment with five heterogeneous clients with a fixed batch size and different epochs.



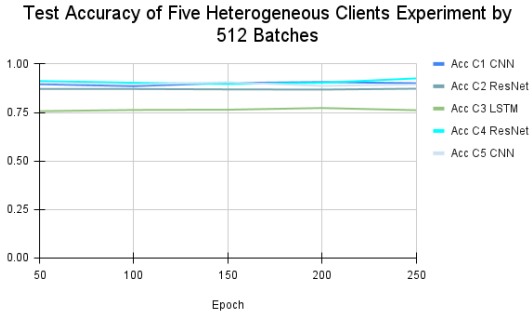
(**a**) 64 Batches



(**b**) 128 Batches



(**c**) 256 Batches



(**d**) 512 Batches

Figure 6. Accuracy of the experiment with five heterogeneous clients with a fixed batch size and different epochs.

Averaging and Updating

The heterogeneous experiment with five clients implemented more than two deep learning models; thus, summing and division in this experiment were totally different from those in the experiment using a single-type deep learning model. Table 4 presents the average Loss and Accuracy. The average Loss was 0.3940, whereas the average Accuracy was 0.9080. These results were used to update the next global model.

Table 4. The main results of the experiment with five heterogeneous clients.

	Low Loss	**High Acc**	**Avg Loss**	Avg Acc
First Client CNN	0.4042	0.9194		
Second Client ResNet	0.387	0.8914		
Third Client LSTM	0.3988	0.8759	0.3940	0.9080
Fourth Client ResNet	0.3614	0.928		
Fifth Client CNN	0.4188	0.9256		

4.7. Training and Evaluation with Ten Heterogeneous Clients

The method for selecting the model techniques was the same as that described in the previous sections. We conducted experiments with 10 heterogeneous clients, where the distribution of the local 10 client models was as follows:

- LSTM for the first client.
- CNN for the second client.
- ResNet for the third client.
- CNN for the fourth client.
- ResNet for the fifth client.
- LSTM for the sixth client.
- CNN for the seventh client.
- ResNet for the eighth client.
- ResNet for the ninth client.
- ResNet for the tenth client.

To summarize, based on the previous sections, we had three local client models using CNN, five client models applying the ResNet model, and two client models using the LSTM model.

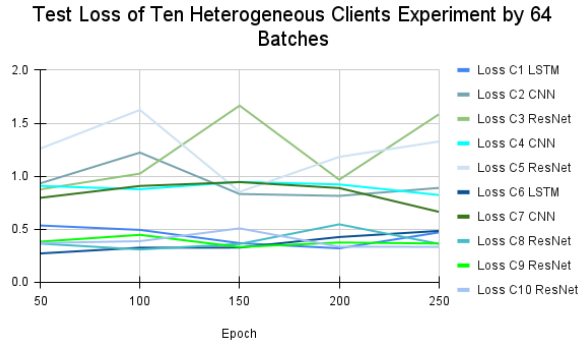
4.8. Ten Heterogeneous Client Results for Training and Evaluation

For more details of the results, Tables 5 and 6 display the experiment's outcomes. The number of times that a deep learning model technique was repeated depended on the most reliable results; therefore, ResNet was the most commonly repeated.

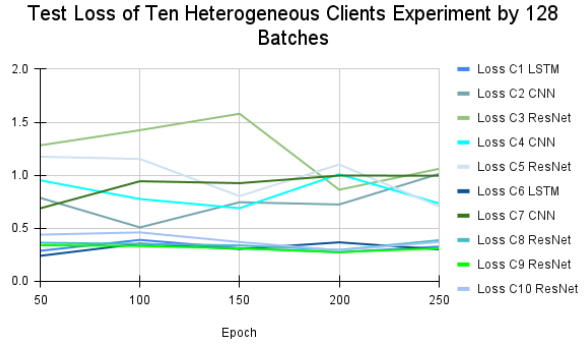
Table 5. The results of the experiment with 10 heterogeneous clients, part 1.

		First Client LSTM		**Second Client CNN**		**Third Client ResNet**		**Fourth Client CNN**		**Fifth Client ResNet**	
Epoch	**Batch**	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**
50	64	0.5358	0.7866	1.1771	0.8835	0.876	0.9156	0.9087	0.9007	1.2608	0.9007
	128	0.288	0.8809	0.9373	0.877	1.282	0.8908	0.9519	0.8908	1.1752	0.8933
	256	0.348	0.8015	0.8574	0.878	0.8016	0.8908	0.7781	0.9082	0.8872	0.8908
	512	0.3394	0.8437	0.7879	0.8765	0.6311	0.9032	0.6011	0.9007	0.706	0.8859
100	64	0.4942	0.8387	1.2689	0.8785	1.0249	0.9032	0.8773	0.8933	1.625	0.9132
	128	0.3906	0.8288	1.0068	0.8785	1.4248	0.8983	0.7761	0.9007	1.1534	0.8983
	256	0.316	0.8387	0.8153	0.885	0.9724	0.8734	0.7339	0.8983	0.7305	0.9032
	512	0.3406	0.8511	0.7469	0.8696	0.6363	0.9007	0.6244	0.9082	0.6378	0.9007
150	64	0.3694	0.8362	1.2481	0.8795	1.6656	0.9007	0.947	0.9107	0.8526	0.9032
	128	0.3126	0.8511	0.8981	0.8756	1.5796	0.9231	0.6898	0.8983	0.803	0.9032
	256	0.2832	0.8586	0.8919	0.8721	1.0297	0.8983	0.63	0.9256	0.7994	0.9082
	512	0.2881	0.8536	0.7729	0.8721	0.728	0.8958	0.6377	0.9007	0.6747	0.8983
200	64	0.3207	0.8412	1.0503	0.8815	0.9676	0.8809	0.9227	0.9156	1.1823	0.8933
	128	0.274	0.861	0.9969	0.882	0.8634	0.8883	1.0095	0.9057	1.1019	0.9007
	256	0.3017	0.8486	0.8181	0.883	0.7379	0.8908	0.647	0.9181	0.8999	0.866
	512	0.2892	0.8561	0.7983	0.8751	0.7078	0.8983	0.6285	0.8933	0.7701	0.8958
250	64	0.4705	0.8362	0.996	0.8815	1.5837	0.8958	0.823	0.9057	1.3272	0.9206
	128	0.3273	0.866	1.001	0.8805	1.0605	0.9082	0.7351	0.8859	0.7061	0.9032
	256	0.3497	0.8462	0.863	0.8741	0.8039	0.8983	0.6514	0.8908	0.758	0.9032
	512	0.2824	0.861	0.7812	0.8671	0.6863	0.8933	0.6274	0.8983	0.6543	0.8883

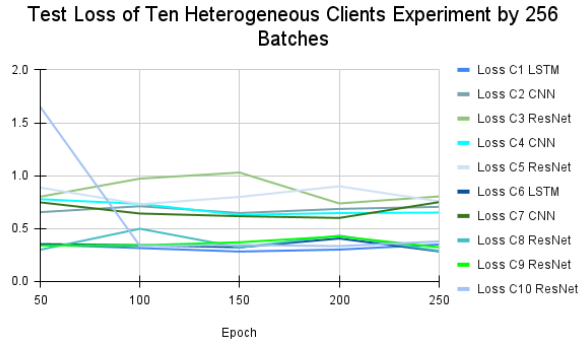
The line charts for the Loss function and Accuracy results of the 10 heterogeneous clients are presented in Figures 7 and 8, respectively.



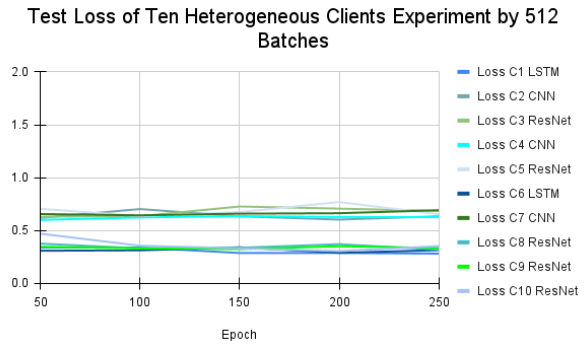
(**a**) 64 Batches



(**b**) 128 Batches

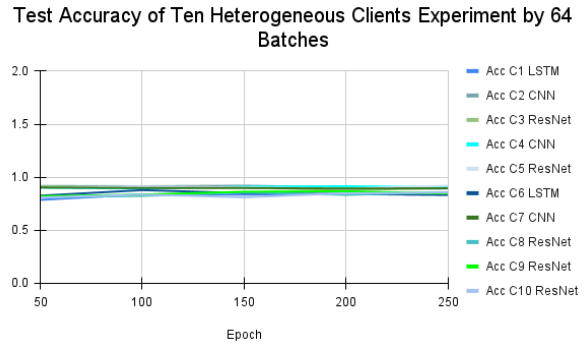


(**c**) 256 Batches

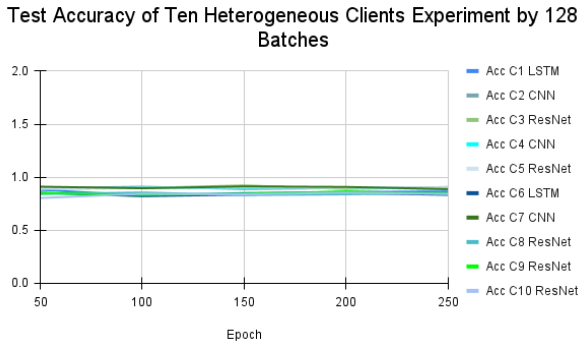


(**d**) 512 Batches

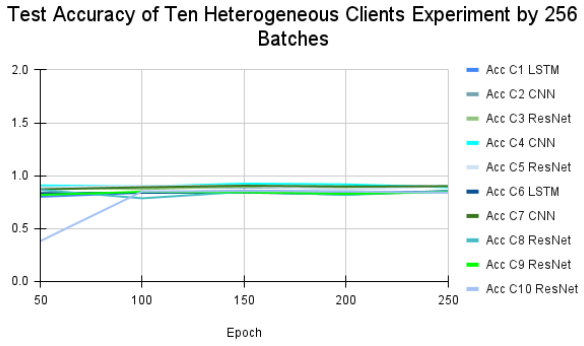
Figure 7. Loss function of the experiment with 10 heterogeneous clients with a fixed batch size and different epochs.



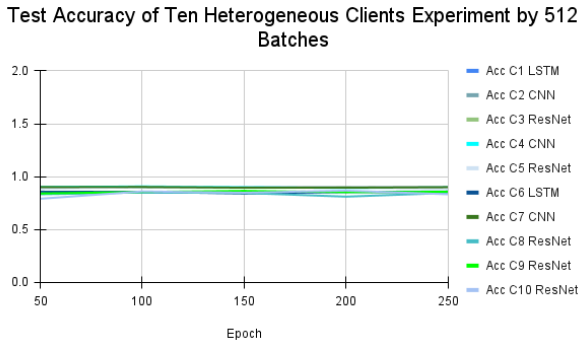
(**a**) 64 Batches



(**b**) 128 Batches



(**c**) 256 Batches



(**d**) 512 Batches

Figure 8. Accuracy of the experiment with 10 heterogeneous clients with a fixed batch size and different epochs.

Table 6. The results of the experiment with 10 heterogeneous clients, part 2.

		Sixth Client LSTM		Seventh Client CNN		Eighth Client ResNet		Ninth Client CNN		Tenth Client ResNet	
Epoch	**Batch**	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**
50	64	0.2723	0.8263	0.7965	0.9032	1.0861	0.8933	1.1707	0.9007	0.9004	0.9057
	128	0.2407	0.8536	0.6883	0.9107	1.7058	0.8859	1.3039	0.8759	1.3014	0.9007
	256	0.3537	0.8362	0.7479	0.871	1.0109	0.8412	0.909	0.8983	1.1925	0.8164
	512	0.3097	0.8586	0.6565	0.9007	0.6939	0.9032	0.6467	0.9082	0.719	0.8908
100	64	0.3265	0.8784	0.9084	0.8983	0.9428	0.9156	1.1473	0.8933	1.1495	0.8933
	128	0.3552	0.8213	0.9435	0.8958	1.1757	0.8983	0.9944	0.9181	0.9617	0.8933
	256	0.3432	0.8387	0.6424	0.8908	0.7377	0.9032	0.831	0.9107	0.8539	0.9057
	512	0.3136	0.8561	0.6444	0.9032	0.7639	0.8983	0.7028	0.9007	0.7015	0.8958
150	64	0.3275	0.8511	0.9447	0.8983	1.0406	0.9156	1.2762	0.8958	1.3362	0.9007
	128	0.3049	0.8362	0.9255	0.9132	0.9443	0.8883	0.7933	0.9007	1.5504	0.8958
	256	0.3232	0.8412	0.6173	0.9057	0.8621	0.8983	0.8862	0.9007	0.8931	0.8983
	512	0.3425	0.8412	0.6604	0.8958	0.6764	0.9132	0.7523	0.8958	0.7668	0.8983
200	64	0.4271	0.8412	0.8885	0.8933	1.5131	0.9206	1.1107	0.8834	0.9205	0.9007
	128	0.3676	0.8511	0.9967	0.9082	0.9405	0.8859	1.426	0.9007	1.2957	0.9057
	256	0.4064	0.8238	0.6007	0.8983	0.9654	0.9057	0.9166	0.8908	0.843	0.8983
	512	0.2898	0.8561	0.6645	0.8958	0.7142	0.8933	0.7053	0.9082	0.7512	0.8933
250	64	0.4846	0.8337	0.6645	0.8958	0.8769	0.9082	1.1592	0.9132	1.4928	0.9007
	128	0.3014	0.8337	0.996	0.8883	1.3826	0.8933	0.6934	0.8883	1.13	0.9181
	256	0.2848	0.8561	0.7499	0.9007	0.6501	0.9107	0.6542	0.9107	0.8399	0.8983
	512	0.3151	0.8511	0.6926	0.9007	0.7243	0.8958	0.7326	0.8834	0.6875	0.8958

Averaging and Updating

The estimated Loss and Accuracy are shown in Table 7. The average Accuracy was 0.8965, and the average Loss was 0.4352. Furthermore, these average results were used to update the subsequent global model.

Table 7. The main results of the experiment with 10 heterogeneous clients.

	Low Loss	**High Acc**	**Avg Loss**	**Avg Acc**
First Client LSTM	0.274	0.8809	0.4352	0.8965
Second Client CNN	0.5084	0.9206		
Third Client ResNet	0.6311	0.9231		
Fourth Client CNN	0.6011	0.9256		
Fifth Client ResNet	0.6378	0.9206		
Sixth Client LSTM	0.2407	0.8784		
Seventh Client CNN	0.6007	0.9132		
Eighth Client ResNet	0.2899	0.866		
Ninth Client ResNet	0.2755	0.8685		
Tenth Client ResNet	0.2929	0.8685		

5. Discussion

Different mixtures of the deep learning models were used in the subsequent heterogeneous case. After examining the results of the various experiments with various numbers of clients, we selected the best combination of models for each experiment based on the acceptable Accuracy results for the local models. Consequently, the experiment with two clients that merged the CNN and ResNet models obtained an Accuracy of 82.51%. In addition, we were able to merge more than two models with five and ten clients. Therefore, we merged the CNN, ResNet, and LSTM models by taking the best results from each. The experiment with five clients acquired an Accuracy of 90.8%, while the experiment with 10 clients attained an Accuracy of 89.65%. We observed that the experiment with five clients again achieved good results in the heterogeneous case.

Consequently, the FL process was based on the clients' data and developed the appropriate local model for each client individually. Because the process was heterogeneous, it

became more flexible to choose the appropriate deep learning model for each local model based on local data. Thus, it introduced a flexible platform for the FL process.

6. Conclusions

With the FL's mixed approach, we used heterogeneous local models by mixing three deep learning modes: CNN, ResNet, and LSTM. Thus, a distinct experiment was introduced via three different numbers of clients: two, five, and ten. The five clients achieved the best results compared to the others with 90.8%, followed by ten with 89.65%. The last result was recorded in two clients that mixed CNN and ResNet, with 82.51%. Based on our current understanding, the heterogeneous case for the local model can be the first experiment used according to previous works.

Author Contributions: Conceptualization, N.S.A. and D.M.I.; Methodology, N.S.A.; Software, N.S.A.; Validation, N.S.A.; Formal analysis, N.S.A.; Investigation, N.S.A.; Resources, N.S.A.; Writing—original draft preparation, N.S.A.; Writing—review and editing, D.M.I. and N.S.A.; Visualization, N.S.A. and D.M.I.; Supervision, D.M.I.; Project administration, D.M.I.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The following URL provides access to the dataset used in this study: UniMiB SHAR, available at http://www.sal.disco.unimib.it/technologies/unimib-shar/ (accessed on 12 November 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yang, J.; Lee, J.; Choi, J. Activity recognition based on RFID object usage for smart mobile devices. *J. Comput. Sci. Technol.* **2011**, *26*, 239–246. [CrossRef]
2. Maskeliūnas, R.; Damaševičius, R.; Segal, S. A review of internet of things technologies for ambient assisted living environments. *Future Internet* **2019**, *11*, 259. [CrossRef]
3. World Population Prospects 2019: Highlights. Available online: https://www.un.org/development/desa/publications/world-population-prospects-2019-highlights.html (accessed on 6 November 2025).
4. UniMiB SHAR. Available online: http://www.sal.disco.unimib.it/technologies/unimib-shar/ (accessed on 30 April 2025).
5. Stisen, A.; Blunck, H.; Bhattacharya, S.; Prentow, T.S.; Kjærgaard, M.B.; Dey, A.; Sonne, T.; Jensen, M.M. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*; ACM: New York, NY, USA, 2015; pp. 127–140.
6. Jiang, W.; Miao, C.; Ma, F.; Yao, S.; Wang, Y.; Yuan, Y.; Xue, H.; Song, C.; Ma, X.; Koutsonikolas, D.; Xu, W. Towards environment independent device free human activity recognition. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*; ACM: New York, NY, USA, 2018.
7. Bhattacharya, S.; Lane, N.D. From smart to deep: Robust activity recognition on smartwatches using deep learning. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*; IEEE: New York, NY, USA, 2016; pp. 1–6.
8. Khan, M.A.A.H.; Roy, N.; Misra, A. AScaling human activity recognition via deep learning-based domain adaptation. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*; IEEE: New York, NY, USA, 2018; pp. 1–9.
9. Radu, V.; Lane, N.D.; Bhattacharya, S.; Mascolo, C.; Marina, M.K.; Kawsar, F. Towards multimodal deep learning for activity recognition on mobile devices. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*; ACM: New York, NY, USA, 2016; pp. 185–188.
10. Lane, N.D.; Georgiev, P.; Qendro, L. Deepear: Robust smartphone audio sensing in unconstrained acoustic environments using deep learning. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*; ACM: New York, NY, USA, 2015; pp. 283–294.

11. Gupta, N.; Gupta, S.K.; Pathak, R.K.; Jain, V.; Rashidi, P.; Suri, J.S. Human activity recognition in artificial intelligence framework: A narrative review. *Artif. Intell. Rev.* **2022**, *55*, 4755–4808. [CrossRef] [PubMed]
12. Gaglio, S.; Re, G.L.; Morana, M.; Ruocco, C. Smart assistance for students and people living in a campus. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*; IEEE: New York, NY, USA, 2019; pp. 132–137
13. Concone, F.; De Paola, A.; Re, G.L.; Morana, M. Twitter analysis for real-time malware discovery. In *2017 AEIT International Annual Conference*; IEEE: New York, NY, USA, 2017; pp. 1–6.
14. Bordonaro, A.; Concone, F.; De Paola, A.; Re, G.L.; Das, S.K. Modeling efficient and effective communications in vanet through population protocols. In *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*; IEEE: New York, NY, USA, 2021; pp. 305–310.
15. Garcia-Constantino, M.; Konios, A.; Mustafa, M.A.; Nugent, C.; Morrison, G. Ambient and wearable sensor fusion for abnormal behaviour detection in activities of daily living. In *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*; IEEE: New York, NY, USA, 2020; pp. 1–6.
16. Syed, L.; Jabeen, S.; Alsaeedi, A. Smart healthcare framework for ambient assisted living using IoMT and big data analytics techniques. *Future Gener. Comput. Syst.* **2019**, *101*, 136–151. [CrossRef]
17. Rajamohan, K.; Nazz, S. Smart home activity recognition for Ambient Assisted Living (AAL). *Ecs Trans.* **2022**, *107*, 20253. [CrossRef]
18. Ghadi, Y.Y.; Batool, M.; Gochoo, M.; Alsuhibany, S.A.; Al Shloul, T.; Jalal, A.; Park, J. Improving the ambient intelligence living using deep learning classifier. *Comput. Mater. Contin* **2022**, *73*, 1037–1053. [CrossRef]
19. Lee, S.; Nirjon, S. Fast and scalable in-memory deep multitask learning via neural weight virtualization. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*; ACM: New York, NY, USA, 2020; pp. 175–190.
20. Jaggi, M.; Smith, V.; Takác, M.; Terhorst, J.; Krishnan, S.; Hofmann, T.; Jordan, M.I. Communication-efficient distributed dual coordinate ascent. *arXiv* **2014**, arXiv:1409.1458. [CrossRef]
21. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [CrossRef]
22. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and Statistics*; PMLR: Cambridge, MA, USA, 2017; pp. 1273–1282.
23. Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C.; Konečný, J.; Mazzocchi, S.; McMahan, B.; Van Overveldt, T.. Towards federated learning at scale: System design. *Proc. Mach. Learn. Syst.* **2019**, *1*, 374–388.
24. Xu, J.; Glicksberg, B.S.; Su, C.; Walker, P.; Bian, J.; Wang, F. Federated learning for healthcare informatics. *J. Healthc. Inform. Res.* **2021**, 5, 1–19. [CrossRef] [PubMed]
25. Agate, V.; Concone, F.; Ferraro, P. A resilient smart architecture for road surface condition monitoring. In *Proceedings of the International Conference on Smart City Applications*; Springer International Publishing: Cham, Switzerland, 2021; pp. 199–209.
26. Abdallah, A.; Maarof, M.A.; Zainal, A. Fraud detection system: A survey. *J. Netw. Comput. Appl.* **2016**, *68*, 90–113. [CrossRef]
27. Wu, Q.; He, K.; Chen, X. Personalized federated learning for intelligent IoT applications: A cloud-edge based framework. *IEEE Open J. Comput. Soc.* **2020**, *1*, 35–44. [CrossRef] [PubMed]
28. Khan, L.U.; Alsenwi, M.; Yaqoob, I.; Imran, M.; Han, Z.; Hong, C.S. Resource optimized federated learning-enabled cognitive internet of things for smart industries. *IEEE Access* **2020**, *8*, 168854–168864. [CrossRef]
29. Oh, S.; Park, J.; Jeong, E.; Kim, H.; Bennis, M.; Kim, S.L. Mix2FLD: Downlink federated learning after uplink federated distillation with two-way mixup. *IEEE Commun. Lett.* **2020**, *24*, 2211–2215. [CrossRef]
30. Kevin, I.; Wang, K.; Zhou, X.; Liang, W.; Yan, Z.; She, J. Federated transfer learning based cross-domain prediction for smart manufacturing. *IEEE Trans. Ind. Inform.* **2021**, *18*, 4088–4096.
31. Kirsten, K.; Pfitzner, B.; Löper, L.; Arnrich, B. Sensor-based obsessive-compulsive disorder detection with personalised federated learning. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*; IEEE: New York, NY, USA, 2021; pp. 333–339.
32. Sozinov, K.; Vlassov, V.; Girdzijauskas, S. Human activity recognition using federated learning. In *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*; IEEE: New York, NY, USA, 2018; pp. 1103–1111.
33. Ek, S.; Rombourg, R.; Portet, F.; Lalanda, P. Federated self-supervised learning in heterogeneous settings: Limits of a baseline approach on har. In *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*; IEEE: New York, NY, USA, 2022; pp. 557–562.
34. Arikumar, K.S.; Prathiba, S.B.; Alazab, M.; Gadekallu, T.R.; Pandya, S.; Khan, J.M.; Moorthy, R.S. FL-PMI: Federated learning-based person movement identification through wearable devices in smart healthcare systems. *Sensors* **2022**, *22*, 1377. [CrossRef] [PubMed]

35. Gudur, G.K.; Perepu, S.K. Federated learning with heterogeneous labels and models for mobile activity monitoring. *arXiv* **2020**, arXiv:2012.02539. [CrossRef]
36. Khoa, T.A.; Nguyen, D.V.; Dao, M.S.; Zettsu, K. Fed xData: A federated learning framework for enabling contextual health monitoring in a cloud-edge network. In *2021 IEEE International Conference on Big Data (Big Data)*; IEEE: New York, NY, USA, 2021; pp. 4979–4988.
37. Wu, Q.; Chen, X.; Zhou, Z.; Zhang, J. Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring. *IEEE Trans. Mob. Comput.* **2020**, *21*, 2818–2832. [CrossRef]
38. Presotto, R.; Civitarese, G.; Bettini, C. Fedclar: Federated clustering for personalized sensor-based human activity recognition. In *2022 IEEE international conference on pervasive computing and communications (PerCom)*; IEEE: New York, NY, USA, 2022; pp. 227–236.
39. Concone, F.; Ferdico, C.; Re, G.L.; Morana, M. A federated learning approach for distributed human activity recognition. In *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*; IEEE: New York, NY, USA, 2022; pp. 269–274.
40. Kumar, A.S. *Federated Inverse Averaging: Toward Pareto-Optimal Minima in Human Activity Recognition*; North Carolina State University: Raleigh, NC, USA, 2021.
41. Sarkar, A.; Sen, T.; Roy, A.K. Grafehty: Graph neural network using federated learning for human activity recognition. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*; IEEE: New York, NY, USA, 2021; pp. 1124–1129.
42. He, X.; Su, X.; Chen, Y.; Hui, P. Federated learning on wearable devices: Demo abstract. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*; ACM: New York, NY, USA, 2020; pp. 613–614.
43. Bettini, C.; Civitarese, G.; Presotto, R. Personalized semi-supervised federated learning for human activity recognition. *arXiv* **2021**, arXiv:2104.08094. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.