



Using Federated Learning and Channel State Information-Based Sensing for Scalable and Realistic At-Home Healthcare

Jeroen Klein Brinke*
j.kleinbrinke@utwente.nl
University of Twente
Pervasive Systems
Enschede, The Netherlands

Martijn van der Linden*
University of Twente
Enschede, The Netherlands

Paul Havinga
p.j.m.havinga@utwente.nl
University of Twente
Pervasive Systems
Enschede, The Netherlands

ABSTRACT

This paper explores the use of federated learning in a realistic household employing existing infrastructure to add new devices and locations by rotating the role of the transmitter among smart devices in a multi-person scenario. Current solutions employ channel state information-based sensing for health care monitoring in various ways to propagate knowledge efficiently; however, these solutions often consider (i) ideally placed devices in (ii) single-participant scenarios and (iii) do not consider the different roles of these devices in a network. Data is collected from four smart devices in a household, assuming three participants, one of which is monitored and the other two function as noise, are assigned to perform activities to replicate a realistic household scenario. Insights are provided on using federated learning in realistic at-home health care when adding a new activity location and client devices, both transmitter-only and full communication devices. Results indicate new devices and locations can quickly be adopted with less data by the federated model without intensive retraining, even in multi-person environments, when doing extensive pre-training.

CCS CONCEPTS

• **Human-centered computing** → *Ubiquitous and mobile computing*; • **Computer systems organization** → **Sensor networks**; • **Computing methodologies** → *Transfer learning*.

KEYWORDS

federated learning, channel state information, human activity recognition, realistic scenarios, data stability, data scalability

ACM Reference Format:

Jeroen Klein Brinke, Martijn van der Linden, and Paul Havinga. 2024. Using Federated Learning and Channel State Information-Based Sensing for Scalable and Realistic At-Home Healthcare. In *European Interdisciplinary Cybersecurity Conference (EICC 2024), June 05–06, 2024, Xanthi, Greece*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3655693.3660254>

*Both authors contributed equally to this research.



This work is licensed under a Creative Commons Attribution-Share Alike International 4.0 License.

EICC 2024, June 05–06, 2024, Xanthi, Greece
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1651-5/24/06
<https://doi.org/10.1145/3655693.3660254>

1 INTRODUCTION

Wi-Fi channel state information is commonly used to unobtrusively monitor human activity recognition and physiology to facilitate smart cities and good health and well-being [14] besides the more established methods such as computer vision, wearable motion sensors, and acoustic-based methods [22]. Channel state information shows the multipath propagation between a transmitter and receiving antenna at the receiver side; it is a phenomenon caused by environmental influences on the signal, such as scattering, absorbing, and reflecting. These environmental influences include humans, and when monitoring the changes in the channel state information over time, the activities of them. This truly unobtrusive approach has proven valuable in numerous healthcare applications, such as recognizing human activities [1, 6], monitoring vital signs [2, 19], tracking sleep patterns, and localizing individuals [5, 23]. Therefore, it is an interesting solution to consider in elderly or nursing homes, where continuous monitoring may be required, but existing solutions may not be a good fit for the target demographic.

However, due to the different propagation paths of radio waves, channel-state information-based sensing is highly sensitive to environmental changes and moving obstacles. Due to the unpredictable impact of these obstacles on channel state information, current solutions struggle with adapting new domains (devices, locations, participants) efficiently. This often results in requiring large training models, artificially generated data [20, 21, 26] or other approaches that depend on the pre-training of models [4, 7, 24] for any new domain. Additionally, as these solutions are based on existing data, there may be privacy concerns due to channel state information potentially containing privacy-sensitive data. A possible solution that mitigates the need for data generation and minimizes the need for training new models is federated learning [3, 11], as models are aggregated in a distributed fashion while training.

Federated learning could increase security as only model information is shared between domains, meaning the privacy-sensitive information of participants is stored locally. However, it comes with increased training overhead to converge on a global model that new domains can quickly adapt. More specifically, while the application of channel state information-based sensing for human activity recognition in federated learning environments has shown promise for a single participant with activities in similar locations relative to the placed devices [11], several challenges impacting its practical implementation need to be explored before they may be adapted in real secure healthcare applications:

- **Existing infrastructure.** Current state-of-the-art considers device locations with similar positions to the monitored activity, which does not accurately reflect real-life scenarios

where wireless devices might be found in less ideal locations (in TV cabinets, as fridges, or on coffee tables).

- **Multi-person households.** At the same time, current research has not considered additional humans additional humans (such as visiting relatives or informal caregivers), which are essentially moving obstacles.
- **Limited data required.** Additionally, new Wi-Fi devices may join the network randomly, and it is crucial to add them as channel state information-based sensors efficiently, which also includes using as little data as possible.

Additionally, exploring different combinations of transmitters and receivers may help prevent potential signal blockages caused by furniture rearrangement [8], additional people, or the monitored individual's movement. Leveraging existing hardware in diverse locations in a household offers unique insight by allowing observations from multiple angles, which can mitigate performance issues related to signal obstruction or increase robustness by focusing on a specific person's location. Additionally, not all devices may be able to participate in the training, but they could still function as transmitters (data generators). To this extent, federated learning is chosen as a valid option, as it nicely combines the desire to explore different combinations of devices while simultaneously generalizing the learning process.

Therefore, the main research question is: *What is the impact of adding new locations and devices with different data availability using federated learning in multi-person households utilizing existing Wi-Fi infrastructure using channel state information-based sensing for human activity recognition in terms of accuracy and convergence time?* To answer this question, experiments were conducted in an actual apartment with three people performing different activities (one designated to be monitored, the potential *patient*), while the transmitter rotates among four different devices for three locations for human activity recognition. A scenario in Fig. 1 shows how different devices may collaborate in a household to sense a person. Different parameters are considered to test the performance of federated learning in multi-person households: i) the amount of pre-training done, ii) new locations and networked devices are excluded from training and added later with different levels, and iii) different levels of data availability. The results are aggregated, and their overall F_1 -scores and convergence times are analyzed. Both their role as potential transmitters and transceivers is explored when adding devices.

The remainder of this paper is structured as follows: first, the state of the art is outlined and discussed (Section 2). Then, the data acquisition and methodology to replicate data analysis and federated implementation are described in Sections 3 and 4, respectively. The results are outlined in Section 5 and an overview of findings are discussed in Section 6. The paper concludes in Section 7.

2 STATE OF THE ART

2.1 Channel state information and human activity recognition

Channel state information is logged as the phase and amplitude of the received signal. The collective channel state information from individual antenna pairs is collected in a channel state information matrix \mathbf{H} (Equation 1), which has the shape of $N_r \times N_t \times N_s$, where N_r ,



1) Alice (yellow) is sitting on the sofa watching TV, but there are no devices nearby. This means she is in an uncovered location. 2) Bob (red) visits Alice and puts the phone on the table, and is connected to the network, which then synchronizes with the global model



3) Alice turns off the TV, which breaks a connection, but moves to a new location to converse with Bob, which is then covered by one of the temporary channels 4) As Bob leaves Alice, she picks up her laptop. The laptop can quickly join the federated network, as it is spatially closely related to the phone's location.

Figure 1: Short scenario showcasing the ideal use of federated learning to quickly allow new devices to function as sensors.

is the number of receiving antennas, N_t the number of transmitting antennas, and N_s the number of subcarriers. Every $h_{ij} \in \mathbf{H}$ is a complex number denoted as $a + bj$, where a and b represent the amplitude and phase, respectively. Due to the sensitivity of radio waves to changes in the environment and humans, the impact of human activities on the amplitude and phase can be used to fingerprint these activities.

$$\mathbf{H} = \begin{bmatrix} h_{00} & h_{01} & \dots & h_{0T} \\ h_{10} & h_{11} & \dots & h_{1T} \\ \vdots & \vdots & \ddots & \vdots \\ h_{(R-1)0} & h_{(R-1)1} & \dots & h_{(R-1)T} \\ h_{R0} & h_{R1} & \dots & h_{RT} \end{bmatrix} \quad (1)$$

2.2 Transferring cross-domain CSI knowledge

In recent years, channel state information has been widely used for human activity recognition and physiology monitoring [15, 17–19, 23], but there are significant challenges in scaling this sensing across different domains. One common approach is transfer learning [4, 7, 24], where different features (or feature sets) or embedding spaces are extracted from the dataset and are used to generate other models for new locations or participants. However, this oftentimes still requires extensive model generation or finding and extracting relevant features.

Another common approach is data generation through generative adversarial networks, or GAN [20, 21, 26], which uses existing datasets or features to generate artificial data that could correlate to different participants or locations and that can then be used to train or fine-tune existing models. However, GANs could be resource-intensive and still require additional training on the models.

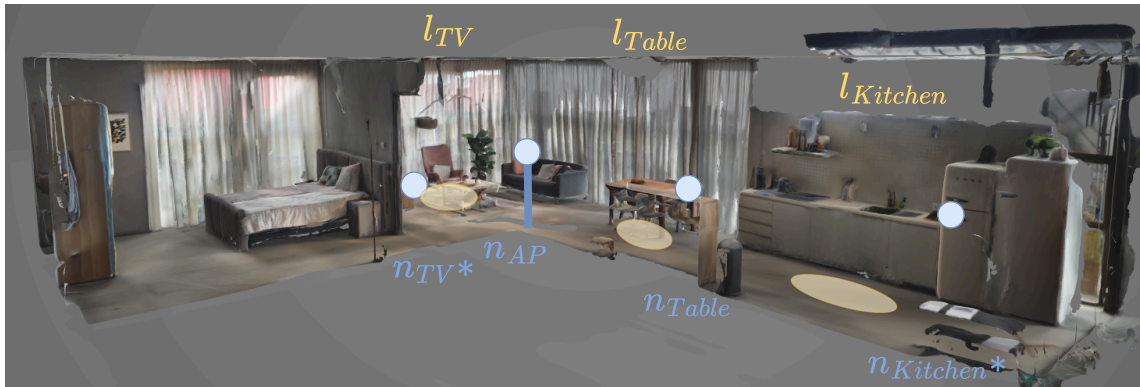


Figure 2: Lidar scan of the apartment (the *e-Health House* at the University of Twente) with all nodes (n_{TV} , n_{table} , $n_{kitchen}$ and n_{AP} , * means behind object). Locations where activities are performed are in yellow (L_{TV} , $L_{kitchen}$ and L_{table}).

2.3 Federated learning and channel state information

Federated learning could mitigate some of the downsides in human activity recognition using channel state information by sharing model knowledge. Hernandez *et al.* [10] achieved promising prediction accuracy while allowing nodes to train with limited data in an indoor environment, possibly classifying activities in unseen locations. The setup is for a fixed Wi-Fi transmitter and receiver pair that may be placed in any location, with the activity being performed in a relatively similar location compared to the devices. The results for this approach seem promising (0.90 in certain locations). The model has a small memory footprint, meaning it could nicely integrate with less powerful hardware. However, the setup does not consider different types of devices, existing infrastructure, or multiple persons to test the system.

2.4 Challenges

As can be seen from state of the art, there is a need for more scalable and robust ways to propagate the knowledge of channel state information-based sensing that do not require extensive retraining or artificial data generation. Federated learning allows devices to collaborate without the need to collect, transmit, and/or process a large amount of data. However, current research on federated learning in combination with channel state information in realistic scenarios ignores realistic scenarios. This limits the use case for wide-scale adaptability. To progress towards a more wide-scale solution, this work explores how robust federated learning is in a life-like setting where wireless devices are placed in representable life-like positions (e.g., living room TV on its cabinet, smartphone on a table, smart refrigerator in a kitchen) in a multi-person household scenario.

Therefore, this study’s work explores the adaptability of new locations and devices in real-life multi-person households for different levels of data availability.

3 DATA ACQUISITION

The channel-state information must first be captured to explore the feasibility of federated learning and channel-state information in

Table 1: Node identifiers and their corresponding locations, with the relevance of presence in the indoor environment.

Node	Location	Relevance
n_{TV}	In between the television and the armchair.	Smart TV, home assistant, casting device
n_{table}	On the table between the living room and kitchen.	Laptop, mobile phone
$n_{kitchen}$	On the kitchen countertop, next to the fridge.	Smart fridge or Wi-Fi-enabled smart kitchen device
n_{AP}	In the middle of the eHealth House elevated (simulating being mounted on the ceiling)	Access point, router

realistic households. This section describes the hard- and software used for capturing, outlines the experimental setup, participants and activities, and discusses the resulting dataset.

3.1 Hardware and software

The Linux CSI Tool [9] was combined with the Intel Ultimate Wi-Fi Link 5300 NIC with a centre frequency of 5.32 GHz. Two antennas were used for transmitting and receiving was done using three antennas (2×3 MIMO) with a packet transmission rate of 100 Hz. The rate of 100 Hz was chosen to easily capture all required details of the activities: human movement usually lies between 0 and 20 Hz, with daily activities between 0.3 and 3.5 Hz [12]. Voluntary human movement generally does not exceed a frequency of 10 Hz. The nodes were not connected over a Wi-Fi network: the transmitter was put in the *injector mode* and broadcasted random packets, while the receivers were placed in *monitor mode* to listen to a specific MAC address (conforming to the 802.11n specifications). Due to the limitations of the Linux CSI Tool, 30 subcarriers are captured per measurement, meaning the shape of the channel state matrix H is $2 \times 3 \times 30$, following $N_t \times N_r \times N_s$. The amplitude is calculated as the absolute of $|a + bj| = \sqrt{a^2 + b^2}$.

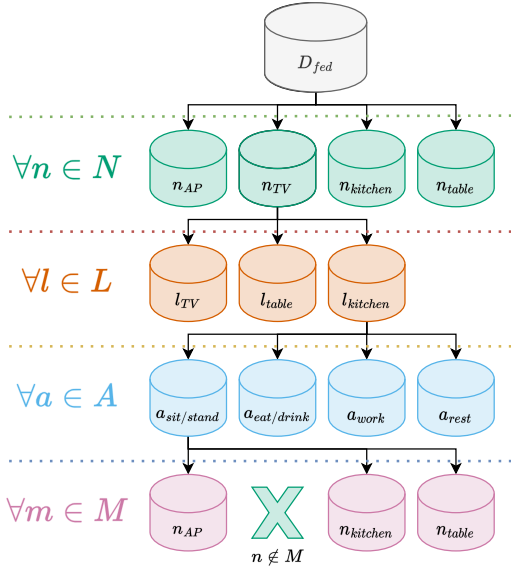


Figure 3: Visualization of the federated dataset D_{fed} , where N is the data received by all nodes, L the subset of received data per locations, A the individual sets of all main activities, and M the subset $N \setminus \{n\}$ of received information from a node m . Note that $m \notin M$, as a receiver n cannot receive data from itself. Additionally, note that all element in N, L, A have the same type of branches, but with their respective received elements.

3.2 Experimental setup

Experiments were conducted in a replicated, fully functioning apartment (the *e-Health House* at the University of Twente), where participants performed activities in different realistic locations, with the nodes in different locations inside the area apartment and receiving packets from which the channel state information can be extracted. In total, four nodes were used to measure the channel state information over time, and they were placed in four locations where one might expect actual wireless devices. Figure 2 shows a Lidar scan of the eHealth House, with node identifiers indicating the location of devices. Table 1 lists the identifiers and locations of each node, together with its relevance in a realistic environment.

3.3 Participants and activities

The experiment involves four activities (sitting/standing, eating/drinking, working, resting) inspired by the Activities of Daily Living (ADL). Three participants were asked to enter the apartment per run, one per location. One of these participants is designated to perform all activities while all nodes have had their chance to transmit. The other two participants perform randomly assigned activities (to generate noise), ensuring that each combination of activities occurs only once for each set of three participants to prevent the learning algorithm from recognising activity combinations instead of individual activities. Each run took twelve minutes per activity for all nodes, meaning the whole activity recognition part of the experiment took 48 minutes per set of participants.

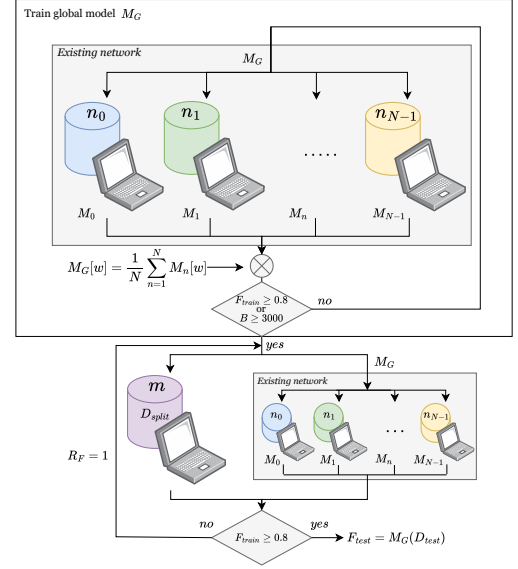


Figure 4: Knowledge flow when adding a new client. A global model M_G is trained using $N - 1$ clients (n_{N-1}) for different numbers of epochs and federated rounds. The weights of the models ($M_n[w]$) are aggregated using FedAvg [25]. When either of the criteria is reached, the new client is added with a specific split of the data (D_{split}). Note that in the second part, all updates are shared immediately after training $E = 1, R_F = 1$ until all nodes report $F_{test} \geq 0.80$ or convergence was reached.

3.4 Resulting datasets

The resulting federated dataset D_{fed} is visualized in Figure 3. In the resulting data, $m \in M$ may be defined as shape of $180 \times 100 \times 2 \times 3 \times 30$, where 180 is the time in seconds for each transmission, 100 is the transmission rate per second, and $2 \times 3 \times 30$ the aforementioned channel state matrix H , resulting in approximately 3,240,000 elements per m , with the total being $\approx 466,560,000$ elements. Here, each element is a complex number converted into the amplitude by taking its absolute value, as outlined in Section 3.1. In total, two of these datasets were collected and combined into a single dataset. Removing a branch location l from L ($L \setminus \{l_l\}$) or a node m from N, M ($N \setminus \{n_n\}$) represents training without location l or node n , respectively. Note that node m is removed from both N and M to properly simulate no nodes receiving any data from node m , while node m may not receive any data.

4 METHODOLOGY

4.1 Data preprocessing

While packets were transmitted at 100 Hz, packet loss may occur at the receiver side, resulting in non-uniform inter-packet arrival times. To guarantee a balanced dataset with sufficient information to allow a network to learn to perform accurate classification, the data was first interpolated to account for missing data points [16]. This is achieved utilizing linear interpolation. Signals with a lower sampling rate than desired can be replicated appropriately using

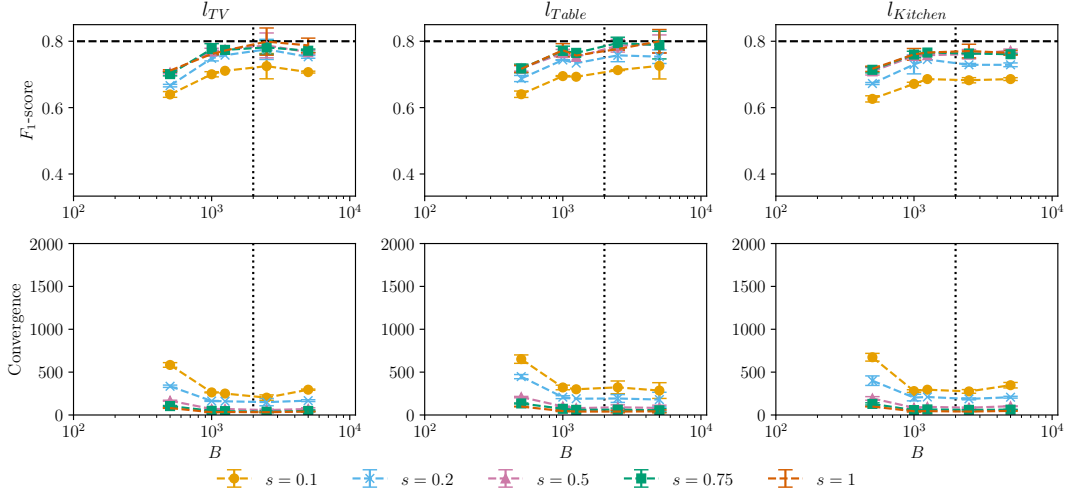


Figure 5: F_1 -score (top) and convergence time (in epochs, bottom) visualized per budget $B = E * R_F$ for different locations (TV, Table, and Kitchen for left, middle, and right, respectively) and different data availability.

interpolation if the sampling rate equals the Nyquist rate or higher. For the activities performed in this research, the Nyquist rate is around 5 Hz [13]. After interpolation, data was normalized and a rolling filter was used to mitigate the effect of signal interference and environmental noise, which could cause sudden changes in amplitude not caused by the activity [10].

4.2 Neural network and federated model

The study employs a dense neural network for classification, utilizing a 2D input array of shape 180×100 , derived from the channel state matrix \mathbf{H} to accommodate a transmission rate of 100 Hz [10]. The network architecture comprises three fully connected layers with 100 hidden units each, using ReLu activations and dropout of 0.5. The model aims to minimize mean-square error loss through stochastic gradient descent. While activity regularization is excluded due to limitations in the used framework, L_2 kernel regularization is applied in a federated learning context to minimize weights via FedAvg [3] optimization. The models are aggregated into a global model following $M_G[w] = \frac{1}{N} \sum_{n=1}^N M_n[w]$, where M_G is the global model, M_n model belonging to a node n , and $[w]$ the weights belonging to a model. A learning rate of 10^{-5} , and the server learning rate of 1. For optimisation, a split of 60% of the original data is used, whereas 40% is preserved for testing the model.

4.3 Introducing new locations and clients

Federated learning is applied to introduce new (unseen) locations and clients to a pre-trained network after different training durations to analyze its performance. The pre-trained model is trained on a subset of locations ($L \setminus \{l_l\}$) or nodes ($N \setminus \{n_n\}$) until F_{train} -score ≥ 0.8 , to prevent overfitting, allowing spatially distributed devices to more easily fine-tune their model afterwards. The model is trained for different budgets defined as $B = ER_F$, where E is the number of local epochs and R_F is the number of federated rounds.

It should be noted that the data associated with a new location or node only consists of data that was not collected after it was added to the network; rather, the data was collected simultaneously with the data collected for the pre-trained model. Therefore, the methodology only serves as a proof of concept and represents a real-life environment only up to a certain point.

Adding a new location. When $L \setminus l$ reaches $F_1 \geq 0.8$ when training on a from a subset of existing locations, l joins federated learning rounds. Retraining involves evaluating for different E and R_F models until all devices using the federated model reach $F_{test} \geq 0.8$ after the new location joins.

Adding a new client. Training new client data involves training models with varying local epochs and federated rounds for $|N| - 1$ nodes. Removing a node is done by excluding its transmitted and received data from the remaining nodes and is denoted as $N \setminus \{n\}$ where n is the removed node. Each node is added in two different ways:

- **Transmitter only.** The global model is not shared with the new device; it functions solely as a data generator for other devices, such as smartphones, contributing valuable insights without burdening itself with neural networks or data collection.
- **Transceiver.** The global model is shared with the new device to fully participate in the network; the new device is integrated into the network as a full communication device (e.g., other access points or smart devices), allowing it to train in a federated manner, facilitating communication.

Figure 4 shows a schematic overview of using the *client* set to train on data from new clients. Note that for all scenarios after a new device or node is added, all weights are shared after every epoch ($R_F = 1$).

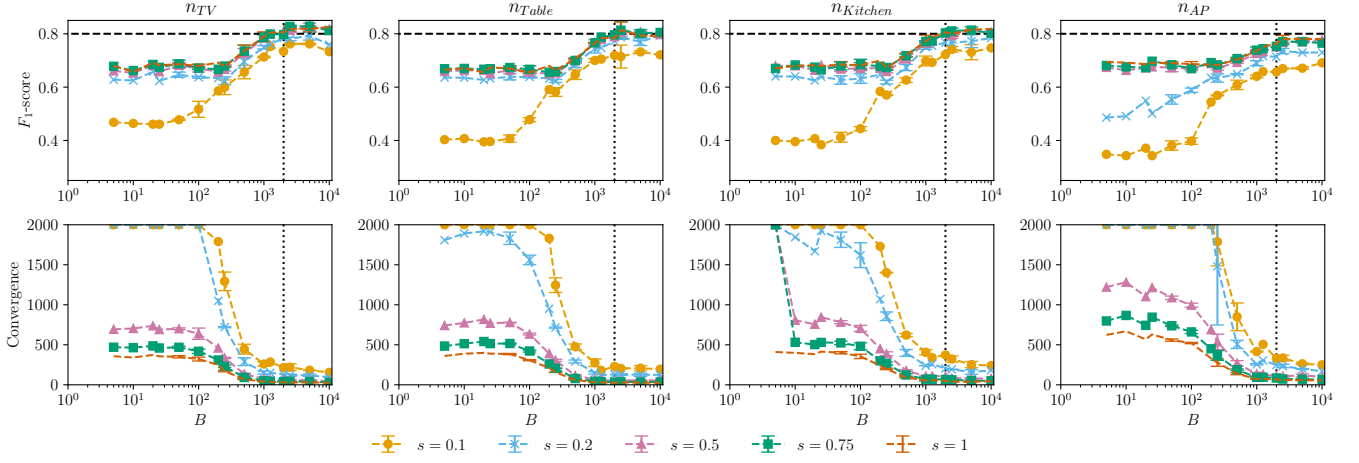


Figure 6: F_1 -score (top, higher is better) and convergence time (in epochs, bottom, lower is better) visualized per budget $B = E * R_F$ for different transmitters (from left to right: TV, Table, Kitchen, and AP) and different data availability (s).

4.3.1 Simulation of limited data availability. Reducing the amount of data needed for pre-training is crucial. Therefore, different fractions of the total available data are used for training to simulate limited data scenarios, namely a split $sin[0.1, 0.2, 0.5, 0.75, 1]$.

5 EVALUATION

The reported F_1 -scores are the overall performance of the federated network. Each node only had access to the channel state information from packets it received from the other clients, and the data of all transmitters was combined into one large dataset for classification unless otherwise specified (such as in the new location and new client set). The reported convergence time (text inside) is based on aggregating the federated model after every locally trained step during personalization.

5.1 Adding a new activity location

When adding an unseen location (Fig. 5), it can be observed that when assuming all data ($s = 1$), the network can restore, or come close to restoring, its performance for all $l \in L$ in a shorter time only for $B \geq 2000$ (the vertical dotted line in the top row). For l_{TV} and l_{Table} with all data, the average restored accuracy starts to drop when compared to $B = 2500$, by $\Delta F_{test} = -0.05, -0.1$ for $B = 100, 500$, respectively, which may be caused by overfitting. The original performance is never fully restored for $l_{Kitchen}$, but reaches $F_1 = 0.77$ for $B \geq 2500$. The convergence times for $s = 1$ are similar for all locations, settling after $B \geq 1000$ at 40 ± 10 epochs. Note that the error bars displayed are based on different combinations for $B = E * R_F$. Generally speaking, it was observed that the performance was better for more local computations (higher values for E).

Individually, it is implied that different locations may be harder to add than others: adding l_{TV} appears to be the easiest to add to an existing network (reaching $F_{test} \geq 0.8$ for $B = 2500$, followed by adding l_{Table} ($F_{test} \geq 0.77, 0.8$ for $B = 2500, 5000$, respectively), and finally adding $l_{kitchen}$, as it always stayed below $F_{test} \leq 0.76$

for $B = 2500$. This may be because different locations have different levels of coverage from the devices: $l_{Kitchen}$ is only close to $n_{Kitchen}$, while l_{TV} and l_{Table} are closer in proximity to n_{AP} , n_{TV} , and n_{Table} .

In terms of data splits (s), it appears a split of $s \geq 0.5$ is viable for $\forall l \in L, B = 2500$ with comparable F_{test} -scores ($\Delta F_1 = -0.025$) and convergence times ($+35 \pm 10$ additional epochs). For l_{TV} , $s = 0.2$ appears to be a viable option, achieving $F_{test} = 0.78$, while for $l_{Kitchen}$ the performance drops to $F_{test} \leq 0.73$. However, note that the convergence time increases by over 300% in most cases (≥ 180). The ability to classify for different levels of data availability is likely also linked to the coverage of a specific location.

5.2 Adding a new device

Transmitter only. When adding a new transmitter (Fig. 6), it can be observed that when assuming all data ($s = 1$), the network can restore, or come close to restoring, its performance for all $n \in N$ in a shorter time only for $B \geq 2000$ (the vertical dotted line in the top row), similar to adding a new location. The effect of overfitting appears less, though the restored accuracy drops slightly ($\Delta F_{test} = -0.02$) for $B \geq 10,000$. Node n_{AP} only reaches $F_1 = 0.77$ for $B \geq 2500$, being the only node not to restore performance. The convergence times for $s = 1$ are different for the devices, n_{TV} and n_{Table} converge after 30 ± 5 epochs, $n_{Kitchen}$ after 45 ± 5 epochs, and 60 ± 5 epochs for n_{AP} $B \geq 2000$. Performance seems to change most between $225 < B < 2500$ ($F_{test} = 0.8, 0.65$, convergence time 50, 250, respectively) and not increase until $B \geq 250$ for both F_1 -scores and convergence times. Similar to locations, the error bars displayed are based on different combinations for $B = E * R_F$. Likewise, performance appeared better for more local computations (higher values for E).

Like locations, adding new devices has different levels of ease of being added: n_{TV} and n_{Table} appear to be the easiest to add (reaching $F_{test} \geq 0.815$ for $B = 2500$ in the shortest convergence time). While $n_{kitchen}$ also allows for $F_{test} \geq 0.81$, the convergence time

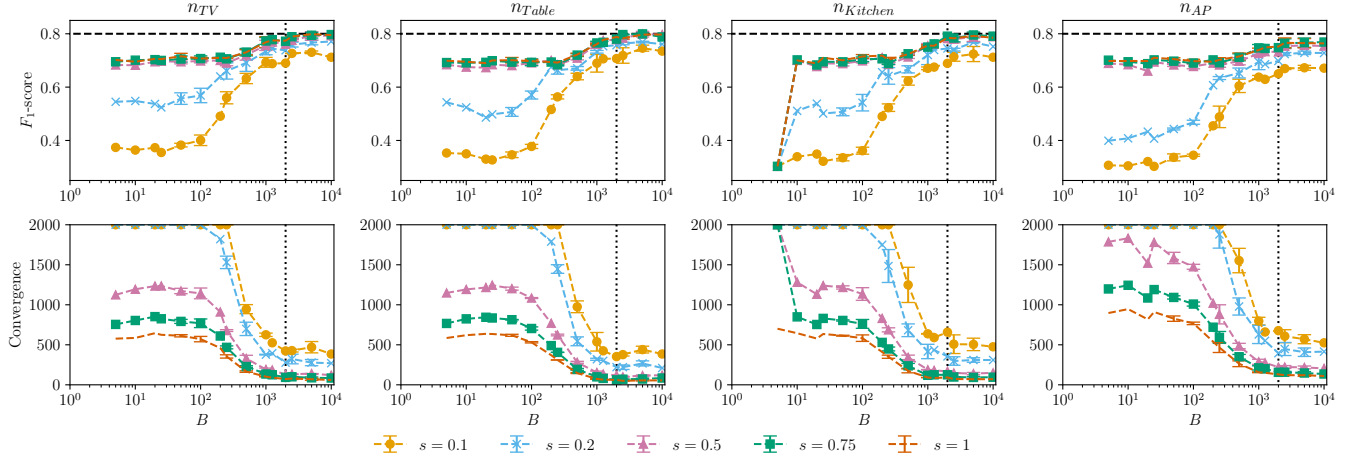


Figure 7: F_1 -score (*top*) and convergence time (in epochs, *bottom*) visualized per budget $B = E * R_F$ for different transceivers (from left to right: TV, Table, Kitchen, and AP) and different data availability (s).

is slightly longer, as outlined before. As noted before, n_{AP} scores lowest while taking the longest to convergence. This could be because it adds significantly different information into the network: the other three nodes are close to locations where activities are performed (thus more immediately impacting the signal), while there was no activity performed to n_{AP} , and it was placed significantly higher than the other three nodes to replicate a ceiling-mounted device.

For $\forall n \in N$, it appears all $s \geq 0.5$ results in comparable F_{test} -scores for $B = 2500$, akin to adding a new location. Likewise, the convergence time takes longer, following a pattern of $conv_{split} = \frac{1}{s} conv_{s=1}$. For all added nodes except n_{AP} , $s = 0.2$ results in a comparable F_{test} , at the lowest $F_{test} = 0.77$ for $n_{Kitchen}$. This indicates significantly less data is needed for convergence, though the time (in epochs) to convergence is increased by a factor 3 on average. This implies the local number of epochs and required additional data could be balanced depending on the available data and training capacity of a node.

Transceiver. When a transceiver is added (Fig. 7), the main observation is that it follows similar patterns to adding a new transmitter, except there appears to be a scaling factor for the convergence time in epochs of on average 2.14 averaged over $s = 0.5, 0.75, 1$ for convergence times that result in a slightly lower restored performance of $F_{test} \geq 0.75$. This is likely because the network needs to adapt to both the newly added data into the federated network on other clients and a newly added client simultaneously, compared to the transmitter where only new data is injected at other clients. This implies that, unlike adding a client as a transmitter-only, it may be beneficial to continue federated learning (rather than just training locally after receiving the global model). This also becomes apparent looking at the restored performance: only for $B \geq 5000$ are added clients n_{TV} and n_{Table} able to restore the performance to $F_{test} \geq 0.8$.

Unlike adding a transmitter, for certain nodes, the model seems to be overfitted for $B \geq 5000$ for n_{Table} and $n_{Kitchen}$ for $s = 0.5, 0.75$.

The total number of additional epochs ranges between 10 – 20 epochs (+10 – 30% when compared to the last non-overfitted entry). However, it should be noted that, realistically, no pre-training of $B \geq 5000$ will happen due to the resource consumption and network overhead required.

6 DISCUSSION

New location. A comparison has been made between the training of existing models on data from various unseen locations. The models' performance is similar for each omitted location for the same data, with $B = 2500$ being the lowest number needed to recover the performance. Pre-training the models extensively improves the F_1 -score as expected, but different locations affect the model's learning curve and may inherently limit its growth. Adding new locations to be recognized within a household can overall be dealt with efficiently by using existing models if these are sufficiently trained beforehand, which could mean either a) a longer existing network that occasionally updates its weights or b) fewer powerful devices that can be used to off-load the training to so that less powerful devices only have to focus on personalization.

New Devices. It is revealed that the location of nodes compared to the activity, their proximity to sharp corners, or their height impacts the accuracy: situations where n_{AP} is added require more convergence time while resulting in lower performance, indicating that the data added and model tuned by this client are inherently noisier. This study implies that in federated learning, the physical location of a new client impacts the network's convergence time and efficiency, with some clients accelerating adaptation to new data due to closer data similarity of other devices. The findings imply the possibility of using federated learning for new devices, though further investigation is required to optimize client placement for improved performance outcomes or automatic client selection.

Limited data. This study finds that while using more data typically leads to faster convergence in model training, 75% of the total data often suffices to achieve equivalent performance, and in certain

cases, even 50 or 25% can be adequate. The balance between data volume, performance, and convergence speed must consider computational and memory constraints and timing needs. Extensive data collection was limited by the practicalities of time, suggesting that an exhaustive exploration of variables (activities, locations, and interference levels) would offer more precise insights but at the cost of significant time investment for each participant.

Device availability. The research presented in this paper assumes that indoor environments contain sufficient computing devices to establish a federated model that may be sufficiently scaled up for other tasks. This study also implies that extensive retraining is required, which may necessitate the need for either devices that are on for extended periods of time (and can handle training) or a lower number of more powerful computers (such as servers or more powerful computers).

7 CONCLUSION AND FUTURE WORK

This main research question of this paper was: *What is the impact of adding new locations and devices with different data availability using federated learning in multi-person households utilizing existing Wi-Fi infrastructure using channel state information-based sensing for human activity recognition in terms of accuracy and convergence time?* While the proposed solution takes longer to pre-train (i) compared to local approaches in state of the art, the federated approach shows that a minimum amount of epochs are required when personalising on unseen locations and nodes (ii), allowing for dynamic environments with changing activity location and device participation, assuming the training can be done in static, more powerful devices (such as computers). Finally, devices can start joining a network sooner, as only a fraction of the data is needed (iii). Overall, federated learning may be used to efficiently add new devices and locations in environments where one participant needs to be monitored, among others, when using existing Wi-Fi infrastructure after enough pre-trained. In the future, research should consider looking into ways to automatically select which devices could be paired to minimize the pre-training needed.

REFERENCES

- [1] Baha A. Alsaify, Mahmoud M. Almazari, Rami Alazrai, and Mohammad I. Daoud. 2020. A dataset for Wi-Fi-based human activity recognition in line-of-sight and non-line-of-sight indoor environments. *Data in Brief* 33 (12 2020), 106534. <https://doi.org/10.1016/j.dib.2020.106534>
- [2] Aboajeila Milad Ashleibta, Qammer H. Abbasi, Syed Aziz Shah, Muhammad Arslan Khalid, Najah Abed AbuAli, and Muhammad Ali Imran. 2021. Non-Invasive RF Sensing for Detecting Breathing Abnormalities Using Software Defined Radios. *IEEE Sensors Journal* 21 (2 2021), 5111–5118. Issue 4. <https://doi.org/10.1109/JSEN.2020.3035960> Transmitted 1kHz, received about 0.86 kHz.
- [3] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017* 54 (2017). arXiv:1602.05629
- [4] Q. Bu, X. Ming, J. Hu, T. Zhang, J. Feng, and J. Zhang. 2022. TransferSense: towards environment independent and one-shot wifi sensing. *Personal and Ubiquitous Computing* 26 (2022), 555–573. <https://doi.org/10.1007/s00779-020-01480-6>
- [5] Rongjie Che and Honglong Chen. 2023. Channel State Information Based Indoor Fingerprinting Localization. *Sensors* 23, 13 (2023). <https://doi.org/10.3390/s23135830>
- [6] Yuh-Shyan Chen, Chun-Yu Li, and Tong-Ying Juang. 2022. Dynamic Associate Domain Adaptation for Human Activity Recognition Using WiFi Signals. *2022 IEEE Wireless Communications and Networking Conference (WCNC) 2022-April*, 1809–1814. <https://doi.org/10.1109/WCNC51071.2022.9771677>
- [7] Yuanrun Fang, Biyun Sheng, Haiyan Wang, and Fu Xiao. 2020. WiTransfer. *Proceedings of the ACM Turing Celebration Conference - China*, 59–63. <https://doi.org/10.1145/3393527.3393538> two self collected datasets (100, 200 Hz), and WiDar3.0 (1000 Hz)-br/>.
- [8] Linlin Guo, Lei Wang, Chuang Lin, Jialin Liu, Bingxian Lu, Jian Fang, Zhonghao Liu, Zeyang Shan, Jingwen Yang, and Silu Guo. 2019. Wiar: A Public Dataset for Wifi-Based Activity Recognition. *IEEE Access* 7 (2019), 154935–154945. <https://doi.org/10.1109/ACCESS.2019.2947024>
- [9] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool Release: Gathering 802.11n Traces with Channel State Information. *ACM SIGCOMM CCR* 41, 1 (Jan. 2011), 53.
- [10] Steven M. Hernandez and Eyuphan Bulut. 2021. WiFederated: Scalable WiFi Sensing using Edge Based Federated Learning. *IEEE Internet of Things Journal* 9, 14 (2021), 12628–12640. <https://doi.org/10.1109/JIOT.2021.3137793>
- [11] Steven M. Hernandez and Eyuphan Bulut. 2022. WiFederated: Scalable WiFi Sensing Using Edge-Based Federated Learning. *IEEE Internet of Things Journal* 9, 14 (2022), 12628–12640. <https://doi.org/10.1109/JIOT.2021.3137793>
- [12] Rinat Khusainov, Djamel Azzi, Ifeyinwa E. Achumba, and Sebastian D. Bersch. 2013. Real-Time Human Ambulation, Activity, and Physiological Monitoring: Taxonomy of Issues, Techniques, Applications, Challenges and Limitations. *Sensors* 13, 10 (2013), 12852–12902. <https://doi.org/10.3390/s131012852>
- [13] Jeroen Klein Brinke, Alessandro Chiumento, and Paul J.M. Havinga. 2023. Channel State Information for Human Activity Recognition with Low Sampling Rates. *UMUM 2023: Second Workshop on Ubiquitous and Multi-domain User Modeling (2023)*, 614–620.
- [14] Jian Liu, Hongbo Liu, Yingying Chen, Yan Wang, and Chen Wang. 2020. Wireless Sensing for Human Activity: A Survey. *IEEE Communications Surveys and Tutorials* 22, 3 (2020), 1629–1645. <https://doi.org/10.1109/COMST.2019.2934489>
- [15] Muhammad Muaz, Ali Chelli, Martin Wulf Gerdes, and Matthias Pätzold. 2022. Wi-Sense: a passive human activity recognition system using Wi-Fi and convolutional neural network and its integration in health information systems. *Annals of Telecommunications* 77 (4 2022), 163–175. Issue 3-4. <https://doi.org/10.1007/s12243-021-00865-9>
- [16] Sebastian Raubitzek and Thomas Neubauer. 2021. A fractal interpolation approach to improve neural network predictions for difficult time series data. *Expert Systems with Applications* 169 (5 2021). <https://doi.org/10.1016/j.eswa.2020.114474>
- [17] Eman Shalaby, Nada ElShennawy, and Aman Sarhan. 2022. Utilizing deep learning models in CSI-based human activity recognition. *Neural Computing and Applications* 34 (4 2022), 5993–6010. Issue 8. <https://doi.org/10.1007/s00521-021-06787-w>
- [18] Xinbin Shen, Lingchao Guo, Zhaoming Lu, Xiangming Wen, and Shuang Zhou. 2021. WiAgent: Link Selection for CSI-Based Activity Recognition in Densely Deployed Wi-Fi Environments. *2021 IEEE Wireless Communications and Networking Conference (WCNC) 2021-March*, 1–6. <https://doi.org/10.1109/WCNC49053.2021.9417335>
- [19] Itsuki Shirakami and Takashi Sato. 2021. Heart Rate Variability Extraction using Commodity Wi-Fi Devices via Time Domain Signal Processing. *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, 1–4. <https://doi.org/10.1109/BHI50953.2021.9508523>
- [20] Dazhuo Wang, Jianfei Yang, Wei Cui, Lihua Xie, and Sumei Sun. 2020. Robust CSI-based Human Activity Recognition using Roaming Generator. In *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. 1329–1334. <https://doi.org/10.1109/ICARCV50220.2020.9305332>
- [21] Chunjing Xiao, Daojun Han, Yongsun Ma, and Zhiguang Qin. 2019. CsiGAN: Robust Channel State Information-Based Activity Recognition With GANs. *IEEE Internet of Things Journal* 6, 6 (2019), 10191–10204. <https://doi.org/10.1109/JIOT.2019.2936580>
- [22] Santosh Kumar Yadav, Siva Sai, Akshay Gundewar, Heena Rathore, Kamlesh Tiwari, Hari Mohan Pandey, and Mohit Mathur. 2022. CSITime: Privacy-preserving human activity recognition using WiFi channel state information. *Neural Networks* 146 (2 2022), 11–21. <https://doi.org/10.1016/j.neunet.2021.11.011>
- [23] Mohammad Zeeshan, Ankur Pandey, and Sudhir Kumar. 2022. CSI-based Device-Free Joint Activity Recognition and Localization using Siamese Networks. *2022 14th International Conference on COMmunication Systems & NETWORKS (COM-SNETS)*, 260–264. <https://doi.org/10.1109/COMSNETS53615.2022.9668391>
- [24] J. Zhang, Z. Tang, M. Li, D. Fang, P. Nurmi, and Z. Wang. 2018. CrossSense: Towards cross-site and large-scale WiFi sensing. *MobiCom 2018*, 305–320.
- [25] Tuo Zhang, Lei Gao, Chaoyang He, Mi Zhang, Bhaskar Krishnamachari, and Salman Avestimehr. 2022. Federated Learning for Internet of Things: Applications, Challenges, and Opportunities. arXiv:2111.07494 [cs.LG]
- [26] Augustinas Zinys, Bram van Berlo, and Nirvana Meratnia. 2021. A Domain-Independent Generative Adversarial Network for Activity Recognition Using WiFi CSI Data. *Sensors* 21, 23 (2021). <https://doi.org/10.3390/s21237852>