BMC Medical Informatics and Decision Making

RESEARCH

Open Access

Target informed client recruitment for efficient federated learning in healthcare

Vincent Scheltjens[1,2*], Lyse Naomi Wamba Momo[1], Wouter Verbeke[2] and Bart De Moor[1]

Abstract

Background Modern machine learning and deep learning methods have been widely incorporated in decision making processes in healthcare in the form of decision support mechanisms. In healthcare, data are abundant but typically not centrally available and, therefore, require some form of aggregation to facilitate training procedures. Aggregating sensitive data poses a significant privacy risk, which is why, both in Europe and the United States, legal frameworks regulate the treatment of such data. Whilst these measures protect the individual behind the data, they pose a significant challenge that results in extensive legal administration related to data sharing efforts. Federated learning (FL) offers a way to mitigate these challenges by allowing to learn models in distributed fashion, eliminating the need to aggregate data for the purpose of training. However, FL comes with a new set of challenges related to communication overhead, client selection and efficiency of the FL training procedure, among others.

Methods In this work, we extend on a previously proposed client recruitment approach by incorporating knowledge on the local hardware such that it becomes possible to recruit a subset of clients for the federation based on the construct of client-level representativeness, which is expressed in terms of the local target distribution divergence, sample size, and the underlying hardware.

Results We show that, for prominent, medical regression and classification tasks, the recruitment approach yields results that are on par, or better, compared to the central and federated approaches. The proposed approach requires a mere fraction of the data for training and reduces the training time by a factor of 3-4. In addition, we show that excluded clients can still significantly benefit from the resulting federated model through local fine-tuning.

Conclusions By expressing the representativeness of clients in function of the deviation in the local target distribution, the sample size and efficiency of the underlying hardware, we are able to define a recruitment approach that yields a subset of clients for the federation resulting in significantly reduced training time, without harming predictive performance, whilst improving the privacy preserving characteristics compared to the standard FL and central approaches.

Keywords Federated learning, Client recruitment, Deep learning

Background

Machine learning (ML) and deep learning (DL) methods have been widely adopted in the healthcare domain and are accompanied by a significant track record showing their value over a broad spectrum of applications [1–3]. Medical data is omnipresent in large quantities and different modalities, often generated at different medical facilities. Patients can be admitted to different hospitals, medical care facilities, different wards within hospitals,

*Correspondence:
Vincent Scheltjens
vincent.scheltjens@kuleuven.be
[1] Department of Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, Kasteelpark Arenberg 10, Leuven 3001, Belgium
[2] Faculty of Economics and Business, KU Leuven, Naamsestraat 69, Leuven 3000, Belgium

and so on. Each of these institutions generates data related to the care and stay episodes for the corresponding patients, resulting in large quantities of distributed data in different modalities. For example, patients admitted to the Intensive Care Unit (ICU) are continuously monitored and generate sequences of temporal data. When need be, these patients may be transferred to the radiology ward where medical imaging generates visual representations along with, most often, written radiology reports. As such, the data is abundant, heterogeneous and distributed. The distributed nature of the data manifests at different levels. At the intra-hospital level, data can reside on different systems pertaining to different wards. At the inter-hospital level, data pertaining to a single patient may partially reside at different institutions as a result of different care episodes or distinct specialties at each of the care facilities. By jointly leveraging all the data as input for ML and DL methods, new insights can be derived and patient care can be improved beyond what was previously possible.

However, the decentralized nature of medical data poses a major challenge for the adoption of ML and DL methods. Especially for DL, large quantities of data are required which often need to be pooled and aggregated centrally to facilitate training. However, legal and regulatory frameworks such as the General Data Protection Regulation (GDPR) [4], the European Union Artificial Intelligence Act, and the United States counterpart, the Health Insurance Portability and Accountability Act (HIPAA) [5], impose significant restrictions on such central aggregation. These frameworks are rightfully instated to warrant privacy for the individuals behind the data, nevertheless, by so doing, inherently and unintentionally, they impose a significant burden for the adoption of advanced DL methods in the healthcare domain. Currently, remediating these challenges requires for de-identification of the data and subsequent agreement upon extensive data sharing agreements.

Federated Learning (FL), as originally proposed by McMahan et al. in 2017 [6], stands as a means to mitigate the restrictions imposed by the legal frameworks and, as showcased in the literature, FL is increasingly studied for applications in healthcare. In recent work, the value of FL has been illustrated for various research and application domains such as for medical imaging, and more specifically classification and segmentation tasks [7–11]. FL allows to learn complex DL models over decentralized data without the need for direct local access. Therefore, data owners maintain full control over the data, eliminating the need to share or centrally aggregate, which directly translates in a reduction of the administrative and legal efforts that would be required otherwise.

In FL, a central server orchestrates the training procedure of a defined model over multiple local clients. These clients can correspond, in practice, to hospitals, mobile phones, government institutions, pharmaceutic companies etc., essentially any sort of party that locally hosts data and can provide computational resources. The local clients, in combination with the central server, make up the federation which communicate with each other bidirectionally to pass model updates. Learning a model in the federation generally follows the predefined FL algorithm outlined in Algorithm 1, [6]. As illustrated in Algorithm 1, the local clients provide a model update to the central server after a local iteration of training. These model updates are aggregated into a global update according to the standard *FedAvg* algorithm in which the local weight matrices are averaged into the global update [6].

Algorithm 1 Standard Federated Learning procedure with T, the number of training rounds, C, all the clients in the federation indexed by c and E, the number of epochs for each round of local training

Server-side execution:
Initialize global model parameters, w
for each training round $t = 1, 2, \ldots, T$ **do**
  Sample a fraction C_t of C clients
  for each client $c \in C_t$ **do**
    $g_c \leftarrow \text{ClientUpdate}(c, w_t)$
  $\bar{g} \leftarrow \frac{1}{|C_t|} \sum_{c \in C_t} g_c$ ▷ Aggregate local updates
  $w_{t+1} \leftarrow$ Update global model using $\bar{g}$

procedure ClientUpdate(c, w)
  $w_c \leftarrow$ Update local parameters with w
  for local epoch e from 1 to E **do**
    $g_c \leftarrow \nabla f_c(w_c)$ ▷ Local parameter update
  Send g_c back to the server

Although, FL provides a way to mitigate otherwise complex privacy related data challenges, it comes with its own set of challenges. As indicated, FL facilitates learning procedures in distributed settings, which inherently means that some form of communication between the participating instances is required. Depending on the scale and amount of participants, the required communication may result in significant overhead which directly influences the cost and the efficiency of the operation. A vastly different, but major, challenge relates to the nature of distributed data. Real-world, decentralized data is not independent and identically distributed (non-IID). Decentralized sets of data are especially characterized by different underlying distributions driven by demographic and geographic parameters as well as differences in terms of the unit of observation, sampling rate, deviating margins of error on the hardware level and different established processes for monitoring and reporting. This non-identically distributed nature of decentralized data can harm the federated training procedure, or any training procedure for that matter, due to the weight divergence in local model updates driven by the non-IID data [12]. Consequently, this not only harms the training procedure but the eventual outcome in terms of the obtained predictive performance.

In FL, the random client selection procedure for each round of training, as illustrated in Algorithm 1, has been shown to be robust against the non-IID nature of decentralized data [6]. However, a significant amount of research effort is being dedicated to defining more efficient, informed ways of selecting the best clients at each round of training [13–15]. These approaches mostly relate to selecting clients in function of previous updates, by prioritizing those that provided a more constructive update [16, 17]. For instance, in [18], the authors show how biasing client selection towards selecting clients with larger local loss, results in faster global error convergence. The major downside of these approaches is that the evaluation of the value a certain client brings, is ad hoc. Therefore, all clients need to partake in the federation and participate in at least one round of training, which is computationally expensive and can pose a burden in practice. Similarly, significant research efforts are being dedicated to the aggregation approaches for construction of the global updates. Again to mitigate the downsides of the non-IID data, as well as to allow for more robust and efficient training that reaches convergence at a faster pace compared to the standard FedAvg approach. To that extent, more advanced methods look at weighted averaging depending on the local sample size or class imbalance such as in [19]. Other approaches such as Stochastic Controlled Averaging [20], look to mitigate phenomena such as client-drift resulting from heterogeneous, non-IID data. In Table 1 we present a condensed, non-exhaustive, overview of the seminal work related to federated learning, client selection and aggregation.

In this work, we tackle the challenges related to the non-iid nature of decentralized data in FL, for which our proposed approach inherently reduces the communication overhead and improves the privacy preserving aspect. As indicated, real-world, decentralized data is non-iid in nature. Therefore, the utility of the data hosted by each of the participants may vastly differ from one host to another. More specifically, data hosted at a given participant may be limited in terms of the instances, be characterized by underlying distributions that are not representative for the average distributions across all participants, or may simply be hosted on outdated, slow, hardware. In the standard federated learning approach, all participants contribute initially to the federation, and only once the orchestrating party has gained sufficient understanding of how valuable certain participants are, it can make informed decisions about which participants partake henceforth. Such client selection procedures are computationally expensive and require for all participants to contribute at least once to the procedure. Here, we aim to remediate this by relying on client recruitment, the foundations for which are provided in the work by Ruan et al. [22]. In doing so, we try to evaluate, a priori, which participants can yield valuable contributions. More formally, a subset of participants can be recruited for which it is known, to some extent, that they are representative of the global population and will yield valuable contributions, resulting in faster convergence without having to sacrifice predictive performance. Essentially,

Table 1 Condensed overview of seminal work related to federated learning approaches, client selection and aggregation

Approach	Authors	Year	Key features
FedAvg	McMahan et al.	2017	Averages model updates from multiple clients to create a global model [6]
FedProx	Li et al.	2020	Extends FedAvg by addressing system heterogeneity and varying amounts of client data. [21]
FedCS	Nishio et al.	2019	Optimizes client selection based on resource conditions to enhance training efficiency [14]
SCAFFOLD	Karimireddy et al.	2019	Combines adaptive sampling and stochastic weight averaging to improve training efficiency and accuracy. [20]

participants with non-representative local data can yield harmful model updates at training time. Therefore, these participants are excluded from partaking in the training procedure, consequently, omitting the need for more intelligent selection and aggregation strategies. To do so, a means to evaluate the local representativeness needs to be constructed. Specifically in this work, we extend our proposed client recruitment approach [23] by incorporating knowledge on the local underlying hardware architecture. As such, the resulting recruitment approach depends on the updated construct of client-level representativeness which is expressed in terms of the local divergence in the target distribution, local sample size, and a proxy of the local training time computed based on the hardware information. All of which constitute high-level, non-privacy infringing statistics that are readily available for each of the participants.

The practical validity and utility of the proposed approach is evaluated in two prominent healthcare settings. The first of which entails Length of Stay (LoS) prediction on patients admitted to the ICU, using the real-world eICU dataset [24–26] for which only 47 out of the 189 potential clients are recruited. The second setting constitutes the multi-label chest radiograph (CXR) classification problem, using the MIMIC CXR data set with structured labels [27], for which 29 out of the 100 potential clients are recruited. These two settings cover both managerial and clinical problems in healthcare that call for regression and classification approaches, and models, respectively. For each of these settings, we show that, with federations constructed of recruited clients only, the resulting models yield predictive performances that either outperform or perform on par with the performances obtained from the central and standard federated models as proposed in [6], albeit, at a fraction of the required training time, whilst providing a full privacy-guarantee for the non-recruited clients. In addition, insights are provided that corroborate the validity of the approach. By evaluating performance of the trained model on clients that did not partake in training (non-contributing clients), we show that with a single round of local fine-tuning, the non-contributing clients can achieve performance on par or better compared to performance for the contributing clients. Furthermore, we visually show that in the classification setting, the model trained with the recruitment approach learns to attribute importance to the exact same sub regions as the other models, whilst having been exposed to a mere fraction of the data at training time.

The remainder of this work is structured as follows: in "Methods" section the client recruitment problem for federated learning is presented along with our proposed method and experimental settings. "Results" section provides an overview of the obtained results which are further discussed in "Discussion" section. At last, concluding remarks are provided along with suggestions for future work in "Conclusions" section.

Methods

Client recruitment

In this work, we do not further explore the alternative approaches for client selection and model aggregation. However, focus is shifted towards client recruitment, which we consider the mechanism to be invoked prior to federated learning, consistent with the work in [22, 23]. The client recruitment problem for federated learning involves recruiting a subset of clients from a larger pool to participate in the training process, essentially before any training has occurred. Formally, given a set of clients C each with local data D_c, the problem is to select a subset $C_k \subset C$ for which we can, a priori, say they will have valuable contributions to the federation based on a limited set of statistics.

By considering a set of limited, non privacy sensitive, statistics pertaining to each of the potential clients, we can define, to some extent, how valuable each potential client is to the federation. By restricting the federation to the most valuable clients, the expectation is that the cost of model training significantly reduces without harming predictive performance.

We define client recruitment as the mechanism that operates on a pool of potential clients to establish the federation for training, such that the output constitutes the clients that will partake in the federation. Consistent with the work in [22, 23] we consider a set C of c potential clients from which a subset of clients will be recruited. Each of the clients in C hosts a local dataset $D_c = \{(x_i, y_i)\}_i$ where x_i and y_i respectively denote the local inputs and corresponding targets.

To initiate the client recruitment procedure, each client in C reports a tuple (P_c, n_c, H_c) to the orchestrating server. The tuple contains three pieces of information; P_c, the local target distribution, n_c, the local sample size and H_c, the underlying hardware that will support the local rounds of training. H_c in itself represents the floating point operations per second (FLOPS) the underlying hardware is capable of computing at a utilization rate that corresponds to the batch size used at runtime. The obtained information allows for n_g and P_g to be calculated as:

$$n_g = \sum_{c=1}^{c} n_c, \quad P_g = \sum_{c=1}^{c} P_c, \tag{1}$$

with n_g the global sample size as the sum over all local sample sizes and P_g the global target distribution as the

sum over all local target distributions. This global target distribution is what will be used in the subsequent steps to calculate to which extent a particular local target distribution deviates from the global distribution, hence the 'target informed' aspect of the proposed client recruitment approach.

To further facilitate the client recruitment procedure, the local representativeness for each client, ν_c, is calculated as:

$$\nu_c = \gamma_{dv} \underbrace{\left| \frac{P_g}{n_g} - \frac{P_c}{n_c} \right|}_{\beta} + \gamma_{sa} n_c^{-0.5} + \gamma_{tr} \theta_c, \tag{2}$$

where ν_c is expressed as the weighted function of the target distribution divergence, local sample size and local computational efficiency with γ_{dv}, γ_{sa} and γ_{tr} the corresponding weight parameters for each of the three components contained in the expression. To illustrate the inner workings of (2), we revisit the structure of the tuple c_i, which consists of three components: P (the outcome distribution over local classes), n (the number of local instances), and H (the computational capacity). Consider three clients with the following characteristics:

$$c_1 = ([10, 40], 50, 2.0)$$
$$c_2 = ([9, 21], 30, 1.5)$$
$$c_3 = ([5, 15], 20, 1.0)$$

For these three clients, the global statistics amount to $P_g = [24, 76]$ and $n_g = 100$. Additionally, we will consider a batch size of 4 for this illustration. Thus, the local representativeness for c_1 is calculated as:

$$\nu_1 = \gamma_{dv} \left| \frac{[24, 76]}{100} - \frac{[10, 40]}{50} \right| + \gamma_{sa} \cdot 50^{-0.5} + \gamma_{tr} \cdot \frac{\text{flop}(50/4)}{2.0},$$

where γ_{dv}, γ_{sa}, and γ_{tr} can be set as parameters assigning weight to each of the components in (2).

Furthermore, in (2), β denotes the divergence of the local target distribution compared to the global target distribution and is calculated as the absolute value of the difference between the normalized global and local target distributions. Furthermore, by including the term $n_c^{-0.5}$, clients with larger local data sets are favored over those with smaller data sets. As discussed in [22], the larger n_c, the better the local empirical distribution, $\tilde{P}_c$, approximates P_c. This observation finds its foundation in the work discussed in [28, 29] in which is shown that $\tilde{P}_c - P_c$ converges to $\mathcal{N}(\mu, \sigma^2)$ with $\mu = 0$ at the rate of $O(n_c^{-0.5})$. The last term contributing to the local representativeness in (2), denoted by θ_c, corresponds to the local computational efficiency based on the underlying local hardware details H_c. Each of the components in the ν metric are individually min-max normalized such that the amplitude of the weight parameters affects each of the components in similar fashion. Furthermore, based on H_c, θ is calculated as:

$$\theta_c = \frac{flop(n_c \div bsize)}{H_c}, \tag{3}$$

where *flop* denotes the floating point operations required for a full forward and backward pass of the data contained in a single batch, *bsize* corresponds to the batch size used for training and H_c denotes the effective FLOPS the underlying accelerator can compute per second. Essentially, θ_c represents local computational efficiency expressed in terms of the approximate time required to process all local data in a single round of training. Intuitively, the inclusion of this term means that, ceteris paribus, clients with more potent local compute are favored over those with less compute at their disposal.

To recruit the final subset of clients for the federation, the local representativeness values, ν_c, are sorted and stored in the vector $\boldsymbol{\nu}$. We define the concept of global representativeness, ν_g, as:

$$\nu_g = \sum_{c=1}^{c} \nu_c, \tag{4}$$

which is, in turn, used to define ι, the fraction of ν_g that should be covered by the clients in the resulting federation. To that extent, the fraction is calculated as $\iota = \gamma_{th} \nu_g$ with γ_{th} a user defined parameter. The subset of recruited clients, C_k, is obtained by collecting the first k clients in the sorted vector $\boldsymbol{\nu}$ such that the fraction ν_g they jointly represent satisfies the defined threshold, ι. This subset of k most representative clients constitutes the clients recruited for the federation.

We iterate that the client recruitment procedure is invoked only once during initialization of the overall training process. It does, however, constitute an extra step compared to methods such as Federated Averaging [6]. In terms of time complexity, the proposed method involves reporting the local statistics, computing the global statistics and subsequently computing the local representativeness, all of which is done in $O(n)$ operations where n corresponds to the number of clients. Sorting the resulting values to select the top k clients requires $O(n \log \text{n})$. Therefore, our proposed method adds $O(n \log \text{n})$ complexity to the end-to-end process, under the assumption that this additional step is offset by the benefits of reduced communication and computational overhead during training.

Experimental setup

To evaluate the efficacy and utility of the client recruitment approach proposed in "Client recruitment" section, two vastly different experimental settings are defined. Both of which are of significant importance in the healthcare domain. In the first setting, client recruitment is evaluated on the LoS problem whereas the second setting assesses client recruitment for accurate multi-label classification of CXRs. The former calls for a regression approach and is more managerial in nature, whereas the latter constitutes a classification problem which is foremost clinical in nature. With this, a broad spectrum of problems in the healthcare domain is covered for which the client recruitment approach can be evaluated using different data modalities and models.

Figure 1 depicts a holistic overview of the experimental setup in which the classification and regression models along with the recruitment approach are visually represented. Figure 1 shows how the central server is in charge of establishing a pool of recruited clients by obtaining high level information from the potential client pool. The recruitment in itself is performed according to the method described in "Client recruitment" section. Subsequently, the established pool of recruited clients is used for the remainder of the standard federated learning procedure in accordance with the depicted models, for which the data, architectures, and experimental settings are presented in the subsequent sections, "Regression" and "Classification".

Regression

The performance and utility of the client recruitment approach for federated models, as described in "Client recruitment", in the regression setting, is evaluated on the LoS problem, where LoS is defined as the remaining time in ICU for a given patient. Here, LoS is calculated as the difference between time at discharge or death T_D and time at admission into ICU T_A, and is expressed in terms of fractional days. Specifically, the task is to accurately predict patient LoS in ICU, similar to the work in [30–33]. Formally, LoS is defined as follows, given a dataset D of patient records, each consisting of features x_i and corresponding LoS y_i for each individual patient, where $y_i = T_{Di} - T_{Ai}$, the goal is to learn a function $f(\mathbf{x})$ that yields $\hat{y}$ as a prediction for y with minimal error.

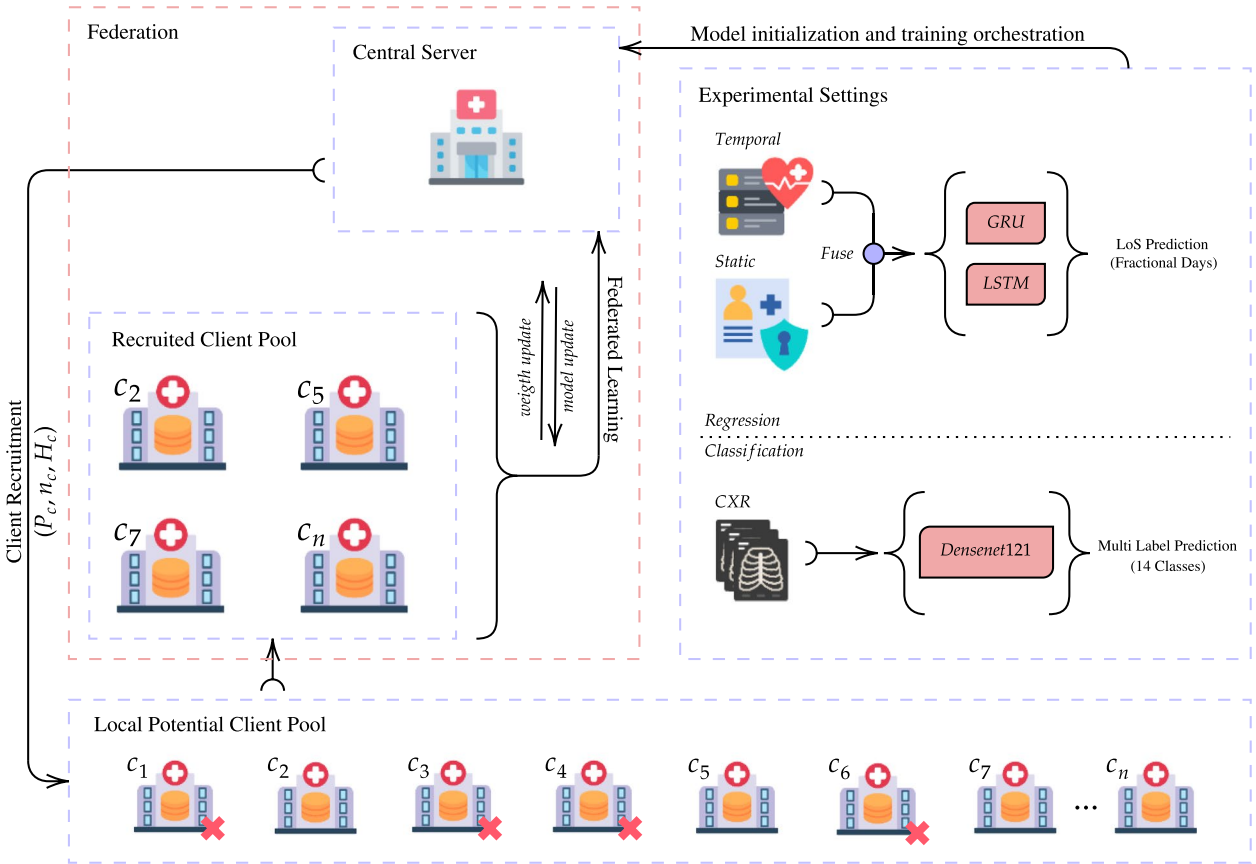


Fig. 1 Overview of the experimental setup of the client recruitment approach for federated learning in the regression and classification settings

Data

To evaluate the proposed approach in the regression setting on real-world, multi-center data, the eICU data set is used [24–26]. The eICU data set contains data pertaining to 208 US hospitals which jointly cover 139.000 uniquely admitted patients with over 200.000 registered patient stays between 2014 and 2015.

The benefit of using the eICU data set with respect to FL is two-fold in that, (i) it is comprised of real world data and (ii) all of the instances and observations contained within the data set can be mapped to the originating institution. The latter is of particular interest as this results in true non-IID data splits across the various institutions, which is the core envisioned application domain of the client recruitment approach. To obtain a final, workable cohort of data from the raw set, the data is preprocessed in accordance with the preprocessing pipeline proposed in [30]. The cohort is restricted to adult patients only, for which the first 24 hours of data into ICU are extracted and used to predict LoS. The extracted data contains a mix of temporal and static features that are fused into a single set of inputs. The set of constructed inputs is cleaned, re-sampled, one-hot encoded and imputed as needed. To track the time since last true observation for the imputed values, a decay mask is added for the temporal features. For an extensive discussion of the preprocessing pipeline we refer to the work by Rocheteau et al. [30].

The obtained data cohort, shown in Table 2, contains 89,127 unique stays which stem from a total of 189 hospitals. A detailed overview of the static and temporal features in the data cohort is provided in Table 3.

As indicated, LoS is expressed in terms of fractional days, which is continuous in nature. However, to obtain client level representativeness as shown in (2), the divergence of the local target distribution compared to the global target distribution (β) needs to be calculated. Therefore, we discretize the continuous target by constructing ten bins, such that the bins represent the frequency of target values that fall within the given bounds. The bins are defined as: $[(0,1),[1,2),[2,3),...,[7,8),[8,14),[14,+\infty)]$ and are used to perform class counting to obtain local target distributions.

Table 2 Overview of the extracted and preprocessed data cohort for the eICU dataset

Number of patient stays	89,127
Train	62,375
Validation	13,376
Test	13,376
Mean LoS	3.69
Median LoS	2.27
Number of features	35
Temporal	17
Demographic	18
Number of hospitals (clients)	189

Figure 2 displays the global target distribution, corresponding to P_g as the aggregation of all local target distributions, compared to a subset of local target distributions. Intuitively, the client recruitment approach aims to pre-exclude those clients for which the local target distribution vastly diverges from the global target distribution, the sample size is not sufficiently large or the underlying hardware results in inefficient training. Figure 2 illustrates the divergence in some of the local target distributions with respect to the global target distribution. Clients with the most divergent target distributions are pre-excluded from the federation as they can potentially contribute non-representative updates. The underlying assumption is that, in the federated setting, when local model updates are driven by non-representative data, the contribution of the local model update to the federation is of lesser value. When the local model update is considered at face value for aggregation into the global update, the effect of a non-representative update is not accounted for in the standard FedAvg setting. To account for the negative effects of such an update, weighting schemes for the aggregation procedure should be introduced. This, however, does not further enhance the privacy preserving aspect, nor does it reduce the pool of clients in the federation, resulting in reduced training time. Nevertheless, each excluded client will still obtain a trained model for local use at inference time, as will be discussed in "Results" section.

Model architecture

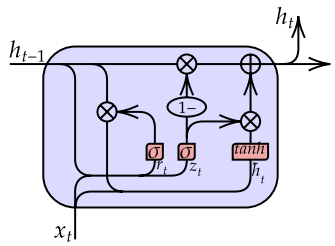
For both the central and federated training procedures, two models are employed, the Gated Recurrent Unit (GRU) [34], and the Long Short-Term Memory (LSTM) [35] networks. Both models belong to the class of deep learning models coined as the Recurrent Neural Networks (RNN) and have shown good performance when dealing with sequential data containing temporal relations. Both the GRU and LSTM cells are comprised of a set of gates that define the information flow and retention. As such, the GRU cell is made up of two sole gates. The reset and update gates, respectively denoted as r_t and z_t in (5). Here, z_t controls the information that is to be retained from the previous state whereas r_t controls the amount of information from the previous state that is to be forgotten. The reduced computational complexity stemming from the two gates is a desirable characteristic for FL where communication overhead and local computational efficiency pose major challenges.

Table 3 Extracted temporal and static features from eICU dataset

Type	Feature	Description
Temporal	FiO2	Patient's FiO2 value
	Bedside glucose	Patient's glucose level
	Cvp	Patient's cvp value
	Heartrate	Patient's heart rate value
	Noninvasivediastolic	Patient's non invasive diastolic value
	Noninvasivemean	Patient's non invasive mean value
	Noninvasivesystolic	Patient's non invasive systolic value
	Respiration	Patient's respiration value
	Sao2	Patient's spO2 value
	St1	Patient's st1 value
	St2	Patient's st2 value
	St3	Patient's st3 value
	Systemicdiastolic	Patient's diastolic value
	Systemicmean	Patient's mean pressure
	Systemicsystolic	Patient's systolic value
	Temperature	Patient's temperature value in celsius
	Hour	Time since admission
Static	Hospitalid	Surrogate key for the hospital
	Gender	Gender of the patient
	Age	Patient's age in full years
	Admissionheight	Admission height of the patient in cm
	Admissionweight	Admission weight of the patient in kg
	Intubated	Whether patient is intubated at the time of the worst ABG result
	Vent	Whether patient is ventilated at the worst respiratory rate
	Dialysis	Whether patient is on dialysis
	Eyes	GCS score (1 to 4)
	Motor	GCS score (1 to 5)
	Verbal	GCS score (1 to 6)
	Meds	Whether GCS score could not be obtained due to meds
	Ethnicity	Patient's ethnicity
	Unittype	The picklist unit type of the unit
	Unitadmitsource	Picklist location from where the patient was admitted
	Unitstaytype	Patient's unit stay type
	Physicianspeciality	Picklist specialty of the care provider
	> 89	Whether patient is over 89 years old

The governing equations for the GRU cell are shown in (5), along with the visual representation. For each discrete time step t in the input sequence, the computations outlined in the governing equations occur.

Similarly, the governing equations for the LSTM cell are outlined in (6), along with the visual representation. Compared to GRU, LSTM counts one additional gate for a total



$$
\begin{aligned}
r_t &= \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \\
z_t &= \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \\
\tilde{h}_t &= \tanh(W \cdot [r_t \odot h_{t-1}, x_t] + b) \\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t
\end{aligned}
\tag{5}
$$

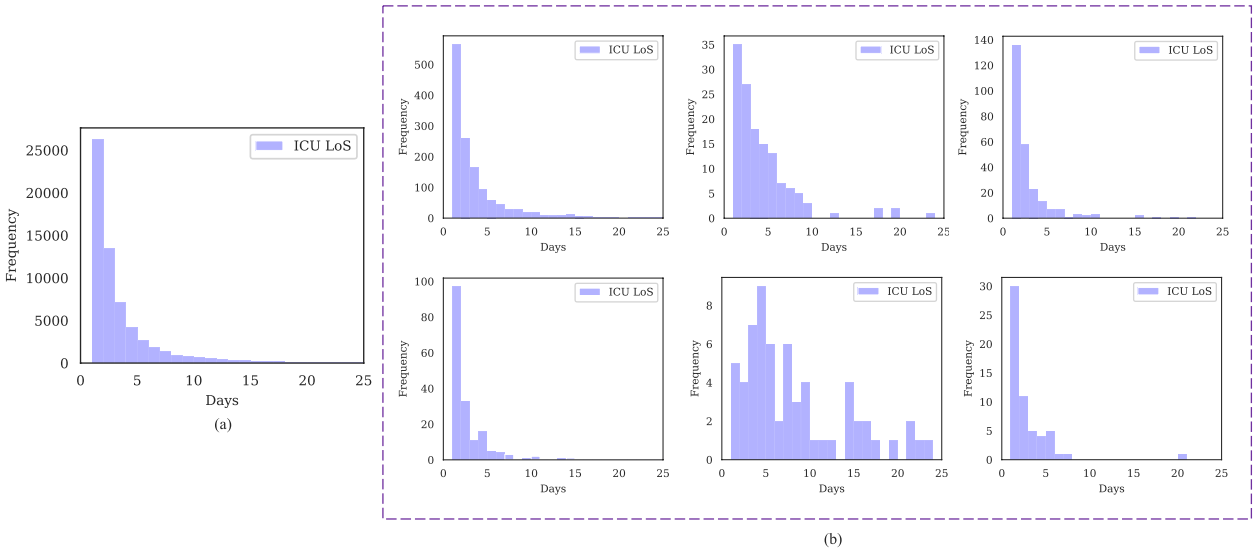
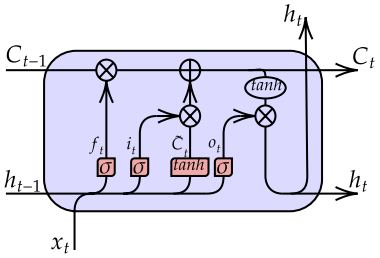


Fig. 2 The global target distribution calculated as P_g, (**a**), and a subset of six local target distributions of the, in total, 189 target distributions, (**b**). Each of the distributions shows the frequency in terms of patients with an observed LoS, restricted until 25 days, locally and globally

of three gates, denoted as f_t, i_t and o_t, which correspond to the forget gate, input gate and output gate.



$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{6}$$

The input gate, i_t, defines how much new information is transferred to the current state at time t, whereas the forget gate, f_t, controls the amount of information from the previous state that is to be disregarded. The output gate, o_t, controls how much information of the new cell state can be contained in the new hidden state.

In both the GRU and LSTM architectures as outlined in (5) and (6), x_t denotes the input at time step t, h_{t-1} is the hidden state at the previous time step $t-1$ and h_t denotes the hidden state at the current time step t. Furthermore, σ is the sigmoid activation function, tanh corresponds to the hyperbolic tangent activation function and $\odot$ represents element-wise multiplication.

The hidden state h_t, i.e., the output of the cell, is provided as the input of a nonlinear Fully Connected Network (FCN), which yields a single output value representing the predicted LoS. The nonlinearity stems from the ReLU activation function leveraged in the FCN as shown in (7),

$$\hat{y}_t = ReLU\left(W_{y_t} h_t + b_{y_t}\right), \tag{7}$$

with $\hat{y}_t$ the predicted value for LoS at time t. Employing $ReLU(x) = max(0, x)$ forces the outcome to be strictly positive. It is impossible for a patient to have a negative LoS, therefore we restrict the model to only yield predictions in the positive domain.

Experimental settings

To allow for comparison between the traditional training procedure in which data is centrally aggregated, the federated setting with standard federated averaging and the federated setting with the proposed client recruitment approach, we define experimental settings for the central and federated approaches in this section. The defined experimental setup serves the main purpose to answer three central questions related to the proposed approach; (i) Can client recruitment improve efficiency without sacrificing performance compared to alternative approaches? (ii) Are findings consistent across data modalities and learning tasks? and (iii) Can

Table 4 Hyperparameter settings used for both central and federated training

Model	*L*	*N*	η	*m*	*wd*	*r*
GRU	2	32	0.005	128	0.005	0.05
LSTM	2	32	0.005	128	0.005	0.05

non-contributing clients still benefit from a resulting global model? To facilitate comparison, several of the settings are fixed over the different settings such that, at training time, all training procedures use *AdamW* [36] for optimization and the Mean Squared Logarithmic Error (MSLE) as the loss function, which is calculated as:

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(\log(y_i+1) - \log(\hat{y}_i+1))^2, \tag{8}$$

with y_i the true target value and $\hat{y}_i$ the predicted value. Furthermore, the model hyperparameters are fixed over all training iterations, both central and federated. The exact settings are shown in Table 4, with L the number of layers, N the hidden dimension for each of the layers, η the learning rate, m the batch size, wd the weight decay for the *AdamW* optimizer and r the dropout rate. To obtain the hyperparameters reported in Table 4, grid search was performed for GRU to determine the optimal number of hidden layers L and the number of hidden units N per layer. Subsequently, batch size (m), learning rate (η), weight decay (wd), and dropout rate (r) were obtained using population-based training [37]. The hyperparameters optimized for GRU were then kept constant for LSTM models to maintain consistency and comparability across experiments.

For evaluation of the performance, all resulting models are evaluated against the hold-out test set containing data from all 189 hospitals. In addition to the MSLE, models are evaluated using the Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and the Mean Squared Error (MSE), shown in (9). As an indication of the time complexity, the training time, denoted as τ, is reported in seconds.

$$\begin{aligned} MAE &= \tfrac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \\ MAPE &= \tfrac{1}{n}\sum_{i=1}^{n}\left|\tfrac{y_i-\hat{y}_i}{y_i}\right| \\ MSE &= \tfrac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \end{aligned} \tag{9}$$

Central training is performed consistent with the traditional DL procedure in which all data is assumed to be centrally available. The architectures presented in (5) and (6) are trained for a predetermined amount of 15 epochs using the global train and validation sets, which consist of the accumulated data over all potential clients, i.e., the 189 originating hospitals. The resulting central model is subsequently evaluated against the hold-out test set.

Federated training with and without client recruitment is simulated as a single process using the *FedML* framework as proposed in [38]. In this work, for FL without client recruitment, clients are either all considered in each round of training or a subset is randomly sampled consistent with the standard client selection implementation described in *FedAvg* [6]. For FL with client recruitment, the client recruitment process described in "Methods" is invoked prior to initiating the federation. Following the described implementation, the recruitment of clients is influenced by four user-defined hyperparameters $\gamma_{dv}, \gamma_{sa}, \gamma_{sa}$ and γ_{th} which respectively define the importance of the divergence in the target distribution, the local sample size, the importance of the efficiency of the underlying hardware, and the fraction of the global representativeness to be covered by the recruited clients.

For LoS prediction, four different strategies are implemented. The four approaches differ in terms of the number of clients that partake in the federation, denoted as ϵ, the percentage of clients in the federation that contribute to each training round, denoted as δ, and whether the federation is comprised of recruited clients, or all clients. For each of the federated models, each client trains for four epochs per round of server-client communication for a total of 15 rounds. The resulting model is subsequently evaluated against the hold-out test set.

The specifications for the four different FL strategies are; (i) **Fed-AC** (Federated All Clients): all clients make up the federation and partake in each training round, (ii) **Fed-SC** (Federated Selected Clients): all clients make up the federation, 10% of which are randomly sampled to partake in each training round, (iii) **Fed-ARC** (Federated All Recruited Clients): recruited clients make up the federation and partake in each training round and (iv) **Fed-SRC** (Federated Selected Recruited Clients): recruited clients make up the federation, 10% of which are randomly sampled to partake in each training round.

Following the introduction of the strategies, we revisit and extend Algorithm 1 to create a formal representation of the Fed-SRC strategy which reflects the client recruitment procedure, nested in the entire federated training approach. This formal representation can be found in Algorithm 2. Similarly, the formal, algorithmic representations for Fed-AC and Fed-ARC can be found in Appendix A. We note that Algorithm 1 constitutes a formal representation of the Fed-SC strategy.

Algorithm 2 Fed-SRC: Federated Learning with Selected Recruited Clients

Server-side execution:
Initialize global model parameters, w
for each client $c \in C$ **do**
  $P_c, n_c, H_c \leftarrow \text{ClientReport}(c)$
Calculate global statistics: $P_g = \sum_{c \in C} P_c$, $n_g = \sum_{c \in C} n_c$
for each client $c \in C$ **do**
  Calculate local representativeness:

$$\nu_c = \gamma_{dv} \left| \frac{P_g}{n_g} - \frac{P_c}{n_c} \right| + \gamma_{sa} n_c^{-0.5} + \gamma_{tr} \theta_c,$$

Select top k clients to yield C_k

for each training round $t = 1, 2, \ldots, T$ **do**
  Sample a fraction $C_{k,t}$ of C_k clients
  for each client $c \in C_{k,t}$ **do**
    $g_c \leftarrow \text{ClientUpdate}(c, w_t)$
  $\bar{g} \leftarrow \frac{1}{|C_{k,t}|} \sum_{c \in C_{k,t}} g_c$ ▷ Aggregate local updates
  $w_{t+1} \leftarrow$ Update global model using $\bar{g}$

procedure ClientReport(c)
  $P_c \leftarrow$ Target distribution at client c
  $n_c \leftarrow$ Number of data points at client c
  $H_c \leftarrow$ Computational capacity of client c
  return (P_c, n_c, H_c)

procedure ClientUpdate(c, w)
  $w_c \leftarrow$ Update local parameters with w
  for local epoch e from 1 to E **do**
    $g_c \leftarrow \nabla f_c(w_c)$ ▷ Local parameter update
  Send g_c back to the server

Time complexity: As indicated in the description for each of the individual strategies, each of the included strategies considers a different number of clients at each round of training depending on the clients in the federation and sampling strategy. Therefore, the overall time complexity for each of the strategies is different. To that extent, Fed-SC and Fed-SRC have the added advantage that for each round of training, only a subset of the clients is selected for training, which lowers the required communication and computational overhead. For example, the Fed-AC strategy considers all clients in every round, resulting in the highest overhead with a time complexity of $O(T \cdot n \cdot E \cdot f)$, where T is the number of global iterations, n is the number of clients, E is the number of local epochs, and f represents the complexity of local training. The Fed-SC strategy randomly selects a subset of clients each round, reducing the overhead to $O(T \cdot k \cdot E \cdot f)$, where k is the number of sampled clients, making it more efficient. The Fed-ARC strategy recruits a subset of representative clients, maintaining a similar complexity to Fed-AC but with a smaller n, as the total pool of clients in the federation has already been reduced by the recruitment procedure. Finally, the Fed-SRC strategy optimizes further by sampling from the recruited subset, significantly reducing training time and complexity to $O(T \cdot m \cdot E \cdot f)$, where m is the number of sampled recruited clients, thus achieving the lowest overhead among the proposed methods.

Additional analysis: To provide more extensive insight with respect to the proposed client recruitment approach, the approach is evaluated in two additional settings, using the GRU based Fed-SRC strategy. Without interfering in the parameter settings for Fed-SRC, the client recruitment approach is tweaked such

that in the first additional setting, a subset, equally large as in the normal Fed-SRC approach, of random clients is recruited. In the second setting, the subset, again equally large compared to the subset in the normal Fed-SRC approach, of least representative clients is recruited. Intuitively, if the standard Fed-SRC approach does in fact yield a more representative subset of clients, proposed approach should outperform the two additional experimental strategies.

Given that the client recruitment mechanism preexcludes a set of clients to take part in the federation for training on the premise that, the locally hosted data is not sufficiently large, diverges significantly in terms of the target distribution or has inefficient hardware resources for training, we, in addition, evaluate the performance of Fed-SRC on the subset of non-recruited clients with and without a single round of domain adaptation. More specifically, the globally obtained federated model is tested against the locally hosted test sets at the non-recruited clients, only. Subsequently, each of the non-recruited clients is allowed to run a single round of fine-tuning on the local data after which the fine-tuned model is again tested against local test data only.

Classification

In addition to the regression setting, the client recruitment approach is evaluated on the multi-label CXR classification problem. To this extent, the MIMIC CXR data set with structured labels [27] is used. Each of the CXR images in the data set is labeled with a multi-label, containing information on the presence or absence of the 14 possible pathologies. The task is to accurately classify any given CXR into any of the 14 classes. The task is multi-label in nature with 14 dimensions in the outcome, pertaining to the 14 classes. Similar to the work in [39], we restrict the performance evaluation to five classes of interest, namely, Atelectasis, Cardiomegaly, Consolidation, Edema and Pleural Effusion.

Data

The MIMIC CXR data set [27] with structured labels contains 227,827 observations, corresponding to 377,095 multi-view chest X-Rays. In Table 5, a descriptive summary is provided for the data set and the corresponding train, validation and test split used throughout this work. The splits are performed such that multi-view CXRs corresponding to a single study belong to the same split to avoid information leakage.

Table 5 CXR data set observation count and corresponding splits

CXR images	377,095
Observations	227,827
Train	265,019
Validation	55,807
Test	56,269
Number of clients	100

The data is labeled such that for each observation, a corresponding vector of length 14 exists for which each entry in the vector constitutes the indication of whether the corresponding class is observed (1), not observed (0), or it is uncertain (−1) as to whether the pathology is present. Following the work presented in [39], the uncertainty labels are dealt with by replacing (−1) with (1), which has been shown to yield better predictive performance. In Table 6, an overview is provided of the corresponding label counts on each of the 14 classes.

The CXR images contained in the data set are derived from the raw DICOM files. To that extent the pixel values were normalized to the range [0, 255], and subsequently histogram equalized to enhance contrast. To do so, pixels are forced towards 0 or 255 such that all pixels in [0, 255] appear equally frequent in a given CXR. Subsequently, the images are converted to JPG. In addition to dealing with the uncertainty labels, the raw JPG files are preprocessed by center cropping each CXR to the dimension 224×224. This is done to fit the input dimensions expected by the deep neural network that is employed, without losing valuable information in the image. In addition, a random horizontal flip is performed on the input in the train set to allow for more robust training.

To facilitate federated learning on the CXR data, we construct 100 silos, each of which is assigned with a

Table 6 Overview of the positive, negative and uncertain label observations for each of the 14 classes in the MIMIC CXR data set

Observation	Positive	Negative	Uncertain
Atelectasis	45,808	1,531	10,327
Cardiomegaly	44,845	15,911	6,043
Consolidation	10,778	7,967	4,331
Edema	27,018	25,641	13,174
Enlarged Cardiomediastinum	7,179	5,283	9,375
Fracture	4,390	886	555
Lung Lesion	6,284	862	1,141
Lung Opacity	51,525	3,069	3,831
No Finding	75,455	152,373	0
Pleural Effusion	54,300	27,158	5,814
Pleural Other	2,011	126	765
Pneumonia	16,556	24,338	18,291
Pneumothorax	10,358	42,356	1,134
Support Devices	66,558	3,486	237

sample rate. Subsequently, the data is split into 1000 data shards from which each of the silos samples shards in accordance to the previously assigned sample rate. More specifically, each silo samples between 1 and 22 shards of data. This yields a data set that is imbalanced in terms of local sample sizes. Nevertheless, slight label imbalance as a result of the structure of the data is also present. To facilitate client recruitment, and more specifically, the calculation of β in (2), the local class counts for each of the 14 classes in the target are considered as the local target distribution.

Model architecture

For the central and federated training approaches in the classification setting, a single deep convolutional neural network, DenseNet-121 [40], is used. The DenseNet-121 architecture is based on the Densely Connected Convolutional Networks design principles and consists of several core components. The major building blocks the architecture is comprised of, are a combination of convolutional layers, pooling layers, transition layers, dense blocks and a final classification layer. The dense blocks constitute a sequence of convolutional layers with batch normalization and activations to densely connect the feature maps. The dense connections signify shorter connections between layers close to the input and output by connecting each layer to every other subsequent layer. By doing so, for any layer, all feature maps from previous layers are used as input to that layer [40]. The task of the transition layers is to downsample the feature maps to fit to the input of the next dense block through convolutions, batch normalization and pooling. The consecutive combination of each of these components for the DenseNet-121 architecture used in this work is summarized in Table 7, for which the last layer corresponds to the classification layer that yields the predictions for each of the 14 class labels.

Table 7 DenseNet-121 model architecture

Layer type	In/Out size	Architecture
Convolution	112 × 112	7 × 7 conv, stride 2
Pooling	56 × 56	3 × 3 max pool, stride 2
Dense Block 1	56 × 56	$\begin{bmatrix} 1 \times 1 \ \ conv \\ 3 \times 3 \ \ conv \end{bmatrix} \times 6$
Transition Layer 1	56 × 56	
	28 × 28	
Dense Block 2	28 × 28	$\begin{bmatrix} 1 \times 1 \ \ conv \\ 3 \times 3 \ \ conv \end{bmatrix} \times 12$
Transition Layer 2	28 × 28	
	14 × 14	
Dense Block 3	14 × 14	$\begin{bmatrix} 1 \times 1 \ \ conv \\ 3 \times 3 \ \ conv \end{bmatrix} \times 24$
Transition Layer 3	14 × 14	
	7 × 7	
Dense Block 4	7 × 7	$\begin{bmatrix} 1 \times 1 \ \ conv \\ 3 \times 3 \ \ conv \end{bmatrix} \times 16$
Classification Layer	1 × 1	7 × 7 global average pool
		14 dimensional fully-connected

Experimental settings

For classification, we again maintain the same model and hyperparameter settings across the different learning approaches. As such, we utilize the Binary Cross Entropy (BCE) loss with logits, which is equivalent to wrapping regular BCE in a sigmoid activation function. In addition, *AdamW* [36] is used for optimization throughout the training iterations with the learning rate set to $1e-3$ and the weight decay set to $1e-5$. Both the learning rate and decay, were selected based on population-based training [37] for hyperparameter selection in the central setting. The achieved values were found to provide a good balance between convergence speed and model accuracy, centrally. Upon centrally establishing the parameters, they were kept constant throughout all federated experiments. Furthermore, all training procedures are performed with a batch size of 128. For evaluation of the model performance, the area under the receiver operator curve (AUC) is calculated for each of the 5 classes of interest and the training time, denoted as τ, is reported to assess time complexity of the different training procedures.

Central training: Similar to the regression setting, all CXR data is assumed to be centrally available. The DenseNet-121 architecture is fully fine-tuned on the preprocessed CXR data for 3 consecutive epochs and evaluated against the hold out test set, shown in Table 5.

Federated training: Here, two different federated strategies are considered. Fed-AC and Fed-ARC are omitted, the reason for which is two-fold; (i) these strategies are less likely to occur in a real world setting with many clients as they rapidly become infeasible to compute within reasonable and actionable time, and (ii), due to computational limitations it is not feasible to compute these approaches in the experimental setup. As such, the two remaining strategies entail, (i) **Fed-SC**, for which all of the clients make up the federation and a subset is randomly selected to contribute to each round of training and (ii) **Fed-SRC**, for which the recruited clients make up the federation and a subset is randomly selected to contribute to each round of training. The global

parameters for training are fixed across both approaches to ensure fair comparison. To that extent, during each round of training, each of the contributing clients to that given round locally trains the model for two subsequent epochs before providing the central server with a parameter update for aggregation. The total number of training rounds is set to 20. Upon completion, the models are evaluated against the hold-out test set for which the per-class AUC values are reported.

Additional analysis: As an additional step in the analysis of the resulting classification models, a multi-view CXR labeled with Cardiomegaly, Edema and Pleural Effusion is selected from the test set for which the Gradient-weighted Class Activation Maps (Grad-CAMs) [41] are computed. Grad-CAM yields visual explanations with respect to the important regions in an image for a corresponding prediction. These visualizations, or activation maps, portray which areas of an image the model attributes importance to with respect to a given outcome. To this extent, the Grad-CAMs for a multi-view CXR with postero anterior and corresponding lateral views are computed, and shown for each of the approaches in the experiment. This allows for visual evaluation of the behaviour of each of the methods. More specifically, this allows to determine whether the Fed-SRC approach learns to attribute importance similar to the central and Fed-SC approaches, whilst relying on significantly less training data.

To verify whether the observations based on the visual analysis hold for all data, we quantify the similarity of the activation maps for a larger subset of images in the test set. To this extent, we compute the Structural Similarity Index (SSIM) [42] and Frechet Inception Distance (FID) [43] for the activation maps with respect to each of the classes of interest. Both of these metrics measure similarity between two sets of images. SSIM and FID are both often used in the evaluation of image synthesis methods, such as Generative Adverserial Networks. They can, however, also be used to assess similarity and quality between two sets of images, of which one, should constitute the ground truth. SSIM evaluates the structural similarity between two specific images. By averaging the obtained similarity values over a set of images, the average similarity between two sets of images can be obtained. FID looks at the similarity, or dissimilarity between two sets of images in high dimensional space. That high dimensional space is obtained by extracting feature embeddings from both sets of images using the Inception v3 network. The actual distance is computed as:

$$FID = \|\mu - \mu_w\|^2 + tr\Big(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{\frac{1}{2}}\Big), \quad (10)$$

with $\mathcal{N}(\mu, \Sigma)$ the multivariate normal distribution estimated from the obtained feature embeddings on the Grad-CAMs for the central model and $\mathcal{N}(\mu_w, \Sigma_w)$, the multivariate normal distribution estimated from the obtained feature embeddings on the Grad-CAMs for either Fed-SC or Fed-SRC.

To analyze the similarity in the activation maps, a random subset of 2000 images from the test set is selected. For each of the CXRs in the subset, the activation maps using Grad-CAM are computed for the respective models and corresponding classes of interest. The resulting analysis is such that the obtained scores correspond to the similarity between (i) Central and Fed-SC, and (ii) Central and Fed-SRC, allowing for comparison between Fed-SC and Fed-SRC.

Results

In Table 8, an overview is provided of the parameter settings for each of the different federated approaches, with or without client recruitment, in the regression and classification settings as described in "Experimental settings" and "Experimental settings". This section outlines the obtained results in terms of predictive performance and time complexity and, in addition, provides the obtained results for the respective additional analysis, further validating the proposed client recruitment approach.

Table 8 Paramater settings for the client recruitment approach in both the regression and classification settings, with ϵ the total number of clients in the federation, δ the number of clients randomly sampled from ϵ in each round of training and (γ_{dv}, γ_{sa}, γ_{tr}, γ_{th}) the hyperparameters for client recruitment

Setting	Strategy	ϵ	δ	γ_{dv}	γ_{sa}	γ_{tr}	γ_{th}
Regression	Fed-AC	189	189	-	-	-	-
	Fed-SC	189	19	-	-	-	-
	Fed-ARC	47	47	0.4	0.2	0.1	0.1
	Fed-SRC	47	5	0.4	0.2	0.1	0.1
Classification	Fed-SC	100	10	-	-	-	-
	Fed-SRC	29	6	0.5	0.5	0.4	0.2

Table 9 Model performance for central and federated models with and without client recruitment. Statistical significance among the federated models in comparison to Fed-SC is indicated as [a] at the 5% significance level and [b] at the 1% significance level

Model	Strategy	MAE	MAPE	MSE	MSLE	τ(s)
GRU	Central	2.21 ± 0.02	0.58 ± 0.06	21.94 ± 0.63	0.33 ± 0.01	2129 ± 18
	Fed-AC	2.26 ± 0.05	0.64 ± 0.07[b]	**21.58 ± 0.69**[b]	**0.33 ± 0.02**[b]	5232 ± 27[b]
	Fed-SC	2.26 ± 0.06	0.46 ± 0.06	23.98 ± 1.27	0.41 ± 0.06	1470 ± 35
	Fed-ARC	2.27 ± 0.12	0.57 ± 0.17	22.67 ± 1.83	0.37 ± 0.05	3359 ± 25[b]
	Fed-SRC	**2.21 ± 0.03**[b]	**0.46 ± 0.05**	23.44 ± 0.85	0.38 ± 0.04[a]	**546 ± 26**[b]
LSTM	Central	2.19 ± 0.02	0.53 ± 0.05	22.39 ± 0.47	0.34 ± 0.01	1892 ± 15
	Fed-AC	**2.20 ± 0.03**[b]	0.47 ± 0.05[a]	**23.21 ± 0.49**[b]	**0.37 ± 0.02**[b]	4668 ± 50[b]
	Fed-SC	2.27 ± 0.08	0.45 ± 0.03	24.22 ± 1.16	0.43 ± 0.06	1313 ± 20
	Fed-ARC	2.26 ± 0.04	**0.43 ± 0.01**	24.42 ± 0.46	0.43 ± 0.03	1493 ± 34[b]
	Fed-SRC	2.22 ± 0.03[a]	0.45 ± 0.02	23.81 ± 0.47	0.40 ± 0.02	**616 ± 45**[b]

Regression

Table 9 outlines the results for the training procedures in the regression setting for both GRU, LSTM and the corresponding central and federated approaches. The best performing approaches per metric have been highlighted in bold and the statistical significance, with respect to the Fed-SC approach, is indicated at the 1% and 5% confidence level.

The results in Table 9 reflect the performances of the federated approaches as well as the central approach for both GRU and LSTM. For GRU, our proposed approach, Fed-SRC, yields the best performance amongst the federated models for MAE, MAPE and τ. Whereas Fed-AC returns the best performance on MSE and MSLE. For LSTM, Fed-AC yields the best MAE, MAPE and MSLE. Fed-SRC obtains similar predictive performance with the best time to convergence, τ. We note that for both models, Fed-ARC has a higher time to convergence compared to Fed-SC. Even though the Fed-ARC strategy establishes a federation containing recruited clients, the strategy mandates that all recruited clients partake in every round of training. In Fed-SC only 10% partakes in training each round. In Table 8, δ denotes the number of clients in each round. For Fed-SC and Fed-ARC this amounts to 19 and 47 respectively, which is in turn why Fed-SC converges faster.

For both GRU and LSTM, the proposed Fed-SRC strategy significantly outperforms the traditional FL approach, Fed-SC, in terms of MAE. For both models, the observed performance for MAE is similar to, or on par with the central approaches at approximately a fourth of the training time.

Additional insights

The subsequent section outlines the results pertaining to the additional experiment in which the client recruitment strategy is informed to (i) randomly recruit a subset of clients, or (ii) recruit the subset of least representative clients. For both approaches, the parameter settings reported in Table 8 remain unchanged, yielding a subset of 47 random clients, and a subset of the 47 least representative clients respectively. The results in Table 10 show that when GRU is trained using the Fed-SRC strategy with either random or the worst clients in the recruited subset, the models yield a predictive performance with a MAE of 2.31 days and 2.33 days respectively. Overall, the performance stemming from the procedure with randomly selected clients is observably better compared to the worst clients strategy.

Table 11 presents the results with respect to the second additional experiment in which the obtained federated model is tested against data from non-contributing clients only, yielding a MAE of 2.30 days. Notably, after a single round of fine-tuning on the local data, the obtained average performance on the local test sets, for the non-contributing clients, amounts to a MAE of 2.05 days. This performance increase, as a result of the local fine-tuning, is also reflected in the improved MSE and MSLE.

Table 10 Model performance for the Fed-SRC approach retrained with a subset of recruited clients that either constitute the least representative clients according the the recruitment approach or a set of random clients

Model	Strategy	MAE	MAPE	MSE	MSLE
GRU (Fed-SRC)	Random	2.31 ± 0.17	0.46 ± 0.04	24.48 ± 1.63	0.45 ± 0.12
	Least representative	2.33 ± 0.12	0.44 ± 0.02	24.84 ± 1.22	0.47 ± 0.08

Table 11 Model performance for Fed-SRC tested against data from non-recruited clients only with and without a single round of domain adaptation (DA)

Model	Strategy	MAE	MAPE	MSE	MSLE
GRU (Fed-SRC)	Non-recruited	2.30 ± 0.03	0.47 ± 0.07	26.25 ± 1.24	0.40 ± 0.05
	Non-recruited + DA	2.05 ± 0.06	0.47 ± 0.05	23.41 ± 0.85	0.38 ± 0.05

Table 12 Model performance expressed in terms of AUC scores for each of the classes of interest for central and federated models with and without client recruitment

Model	Strategy	Atelectasis	Cardiomegaly	Consolidation	Edema	Pleural effusion
DenseNet-121	Central	0.77 ± 0.01	0.77 ± 0.01	0.74 ± 0.00	0.84 ± 0.00	0.88 ± 0.01
	Fed-SC	0.78 ± 0.00	0.77 ± 0.01	0.74 ± 0.00	0.85 ± 0.01	0.89 ± 0.00
	Fed-SRC	0.76 ± 0.06	0.75 ± 0.05	0.72 ± 0.08	0.83 ± 0.04	0.87 ± 0.03

Table 13 Classification training time and AUC over the classes of interest

Model	Strategy	Average AUC	τ(s)
DenseNet-121	Central	0.80 ± 0.06	12607 ± 1137
	Fed-SC	**0.80 ± 0.05**	13330 ± 1221
	Fed-SRC	0.79 ± 0.06	**3141 ± 496**

Classification

The results for classification are reported in Tables 12 and 13. Table 12 outlines the AUC scores for each of the classes of interest, whereas Table 13 shows the training time for each of the corresponding approaches along with the average AUC over the classes of interest.

The results in Table 12, represent the achieved per-class AUC for the central, Fed-SC and Fed-SRC approaches. The central approach and FED-SC approaches perform on nearly on par across the board. Amongst the federated approaches, the proposed Fed-SRC approach performs slightly worse compared to the traditional Fed-SC approach. In this instance, the observed performance difference between the federated approaches, across all the classes of interest, is limited to 0.02 in terms of the AUC.

In Table 13, the average AUC over the classes of interest is presented along with the training time required for each of approaches. The proposed Fed-SRC approach outperforms both the central and Fed-SC approaches in terms of the required training time to achieve the reported predictive performance, by a factor of 4. The traditional Fed-SC approach, however, reaches convergence at a slower rate than the central approach.

Additional insights

For classification, Fig. 3 and Table 14 jointly represent the findings for the additional experiment, in which we visually and numerically analyze how the different approaches attribute importance to the input with respect to a certain outcome.

Figure 3 shows for a single multi-view CXR observation in the test set that each of the models consistently attribute importance to the same sub regions. Regardless of the minor variations, the general importance attribution across models and across pathologies is consistent with respect to the outcome.

As a means to corroborate the visual results shown in Fig. 3 Table 14 presents the quantified results with respect to similarity in the importance maps for the different methods over a subset of 2000 images in the test set. As such, for both the Fed-SC and Fed-SRC, the obtained results for SSIM and FID represent the similarity in the activation maps from the respective model and the central model. The results show that, depending on the class of interest, irrespective of the similarity metric, Fed-SRC performs slightly better, or worse compared to Fed-SC. In general, no significant discrepancies, that would signify divergent behaviour in how one of the federated models attributes importance, are observed.

Discussion

Regression

Based on the results shown in Table 9, we note that when considering MAE, Fed-SRC for both GRU and LSTM achieves comparable performance compared to the central approaches for the respective models. With respect

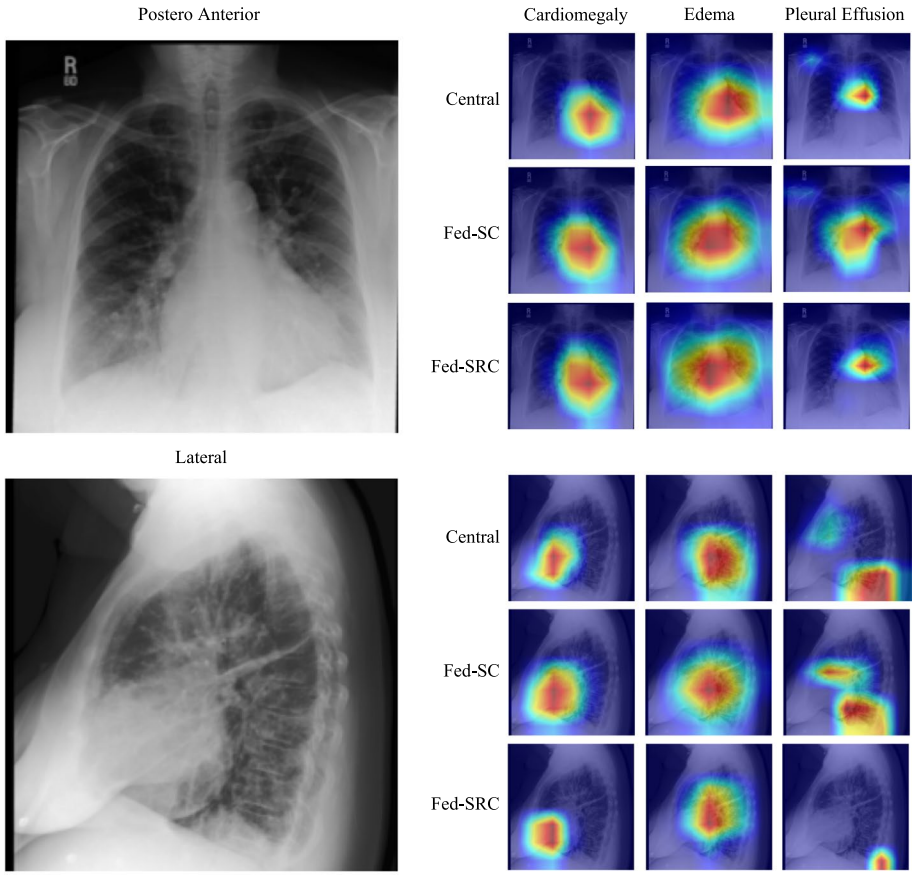


Fig. 3 Grad-CAM visualizations of the pixel importance for each of the 3 labels corresponding to the multi-view CXR selected from the test set. Grad-CAM visualizations are shown on the right and, the raw postero anterior and lateral views are shown on the left

Table 14 Similarity in terms of FID (0 would signify perfect similarity) and SSIM (1 would signify perfect similarity) for the Grad-CAM activation maps corresponding to 2000 images in the test set and the Fed-SC and Fed-SRC models respectively with activation maps from the central model serving as the ground truth

Metric	Strategy	Atelectasis	Cardiomegaly	Consolidation	Edema	Pleural effusion
↑ SSIM	Fed-SC	0.864	0.887	0.991	0.884	0.881
	Fed-SRC	0.831	0.897	0.959	0.886	0.870
↓ FID	Fed-SC	67.88	46.20	6.21	45.11	46.82
	Fed-SRC	70.31	36.92	15.10	39.51	43.85

to the federated approaches, a discrepancy is observed for Fed-AC, in that for GRU it performs comparable to Fed-SC and Fed-SRC, whereas for LSTM, the Fed-AC approach yields the best performance. As LSTMs are prone to overfitting, a partial explanation can be found in the quantity of data used for Fed-AC. In Fed-AC, all data is used, in contrast to the remaining approaches that leverage a mere fraction of all available data, which could potentially lead to overfitting. In addition, for both models, Fed-AC obtains the best MSLE score, which can again be attributed to a better overall fit to the data, as in Fed-AC, the models have been presented with data from all clients, including those with diverging target distributions. When considering Fed-SC and Fed-ARC,

consistency is observed across all metrics when compared to Fed-SRC. Both Fed-SC and Fed-AC underperform compared to our proposed approach.

In this setting, the main comparative discussion revolves around the central approach, Fed-SC and Fed-SRC. From the results in Table 9, we note that for both GRU and LSTM, the client recruitment approach, Fed-SRC, performs significantly better than the standard approach, Fed-SC, with respect to all metrics. Most importantly, the training time reduces significantly whilst the predictive performance improves or remains on par compared to Fed-SC. We further note that, for GRU, Fed-SRC outperforms the central approach, potentially as a result of reduced noise in the training set. As such, the client recruitment approach in the regression setting, for both GRU and LSTM, outperforms the standard FL approach in a fraction of the required training time. In addition, this means that the individuals behind the data hosted at the non-recruited clients, do not run any privacy risk whatsoever, which constitutes a significant improvement over Fed-SC where all data is subject to some form of privacy risk.

The results in Table 10 corroborate that the client recruitment approach does in fact recruit a subset of clients that results in better predictive performance. Both of the additional approaches, in which the least representative clients or a set of random clients make up the federation, yield significantly worse predictive performance compared to the proposed Fed-SRC approach, shown in Table 9.

In practice, non-recruited clients would still be provided with a copy of the resulting federated model, which is primarily optimized in function of the global target distribution. This can yield a model that does not generalize well to the non-recruited clients, given that they were excluded from partaking in the federation. As expected, the results in Table 11, show that when evaluating the resulting federated model, as-is, against non-recruited clients only, performance is subpar compared to the results in Table 9. However, when allowing for a single round of domain adaptation on the local data and subsequent evaluation on the corresponding test set for the local client, performance improves to an extent that, locally, the fine-tuned model yields significantly better predictive performance with an average MAE of 2.05 days across all non-recruited clients. This shows how, when obtaining a global federated model, running one additional round of fine-tuning can, locally, yield performances that significantly outperform the performance of the global federated model. Thus, even for excluded clients, the federated model to which they did not contribute, can still be of significant value. Another benefit relates to the privacy enhancing aspect. The proposed methodology ensures that non-recruited clients do not participate in any of the training rounds and only share a tuple of non-sensitive information when the process is initiated. During recruitment, each client reports the target distribution (P_c), the number of data points (n_c), and the computational capacity (H_c), which are used to calculate local representativeness (ν_c). This shared information does not include any privacy-sensitive details, maintaining the confidentiality of local datasets. Formally, let C be the set of all clients, C_k the set of recruited clients, and $C_{nr} = C \setminus C_k$ the set of non-recruited clients. The non-recruited clients C_{nr} are excluded from training, ensuring that the recruited clients C_k, and the overall training procedure do not access or depend on any information stored at the clients in C_{nr}. This approach ensures that non-recruited clients do not transmit any individual records. Although they do not contribute data during training, they receive the global model for local fine-tuning. This merely requires one-way communication from server to client, where the non-recruited client does not reveal any information with respect to its local data. Finetuning can then be performed locally, with no further data exchange or requirement to be connected to the federation, preserving the privacy of non-recruited client data.

Classification

The main observation based on Table 12, is that each of the different approaches, central, Fed-SC and Fed-SRC, yield similar predictive performance with Fed-SRC performing slightly worse or on par depending on the class of interest. Noteworthy, however, is that Fed-SRC obtains similar performance in a fraction of the time required for the Fed-SC or central approach. More precisely, Fed-SRC reduces the training time by a factor of approximately 4 compared to the other approaches, as shown in Table 13. A major drawback from the Fed-SC approach, resulting in high training time, is that the validation during the training procedure runs against all clients in the federation. This imposes significant communication and computational overhead. This setting is identical for Fed-SRC with the benefit that for Fed-SRC the recruited client pool only contains 29 clients, resulting in faster validation times, and therefore, reduced overall training time.

The resulting visualizations in Fig. 3 show that there is consistency across the models in terms of activated regions in each of the images for the respective pathologies. This representation shows that the approaches identify the pathologies consistently independent of the orientation. We note that the activation maps on the remaining 11 classes are mainly empty. Naturally, the models are not perfect, and thus, misclassifications occur, for which the activation maps will incorrectly show importance. From this single representation and exploratory analysis, we note that the models consistently attribute importance to the same sub-regions to make the prediction. It should be noted here, that the model resulting from the Fed-SRC approach does so whilst only having seen a fraction of the data available at training time compared to the other methods, and additionally, converges in less than a fourth of the required training time.

The results presented in Table 14 show that, depending on the class of interest, either Fed-SC or Fed-SRC performs slightly better. It should be noted that a SSIM of 1 represents perfect similarity whereas a lower FID represents higher similarity between the sets of images. Nevertheless, performance across the board is noted to be similar. When comparing with the activation maps obtained from the central model, both Fed-SC and Fed-SRC, attribute importance to similar regions of interest in the images with respect to a certain prediction. This quantitative approach confirms that the visual observations from Fig. 3 hold for a larger subset of data. This shows that Fed-SRC, whilst having seen significantly less data, still learnt to correctly attribute importance to the same sub-regions compared to the central and Fed-SC approach, in a fraction of the training time. For some of the classes, Fed-SRC shows higher similarity compared to the ground truth, especially when looking at FID. Similar to the Fed-SRC model in regression outperforming the central approach, this can also be partially attributed to the reduction in noisy data as a consequence of the client recruitment approach.

Conclusions

In this work, we present the extended client recruitment approach such that clients are recruited in function of the local sample size, the local target divergence and a proxy of the local training time calculated based on the hardware information. Furthermore, we evaluate the proposed approach in the regression and classification settings, encompassing both managerial and clinical problem sets in the healthcare domain. In both settings, we show how models trained with federations made up of recruited clients, outperform, or at least perform on par, in terms of predictive power, compared to the standard federated or central approaches at a fraction of the training time. In addition, we provide evidence with respect to the validity of the client recruitment approaches by retraining the Fed-SRC model with a subset of randomly recruited clients, as well as the subset of least representative clients. In both scenarios, Fed-SRC, with the normal set of recruited clients, outperforms the two alternative approaches. Furthermore, in the classification setting, we visually show how, for a single image in the test set, each of the models attribute importance to similar regions of the image with respect to a given prediction using Grad-CAM importance maps. We further quantify the similarity in the activation maps for each of the models on a subset of 2000 test images and show, how Fed-SRC performs similar to the central model and Fed-SC at a fraction of the training time whilst only having seen a limited subset of the most representative data. These experiments and corresponding additional insights, show that target informed client recruitment yields models that perform better or on par compared to alternative approaches in terms of predictive power in different, relevant healthcare settings whilst significantly reducing the required training time, and providing improved privacy enhancing characteristics.

By introducing the hardware architecture component in the client recruitment approach, we open up potential future research avenues that allow for more detailed assessment of how local training efforts can be expressed in terms of more complex cost functions. These cost functions can encompass hardware efficiency, related CO_2 emissions, physical distance, demographic factors, etc., among others. This approach could eventually yield a framework that allows to outline the cost structure of intra- and inter-institutional FL endeavours. Future work will assess the performance of the client recruitment approach in a non-simulated setting, which is the current main drawback of this work. The findings presented herein, will still hold in real world implementations as the core concept of client recruitment does not change. However, network latency and communication overhead become an important factor, which can be additional components to be considered in the client recruitment approach.

Appendix A federated algorithms

Algorithm 3 Fed-AC: Federated Learning with All Clients

Server-side execution:
Initialize global model parameters, w
for each training round $t = 1, 2, \ldots, T$ **do**
  for each client $c \in C$ **do**
    $g_c \leftarrow \text{ClientUpdate}(c, w_t)$
  $\bar{g} \leftarrow \frac{1}{|C|} \sum_{c \in C} g_c$ ▷ Aggregate local updates
  $w_{t+1} \leftarrow$ Update global model using $\bar{g}$

procedure ClientUpdate(c, w)
  $w_c \leftarrow$ Update local parameters with w
  for local epoch e from 1 to E **do**
    $g_c \leftarrow \nabla f_c(w_c)$ ▷ Local parameter update
  Send g_c back to the server

Algorithm 4 Fed-ARC: Federated Learning with All Recruited Clients

Server-side execution:
Initialize global model parameters, w
for each client $c \in C$ **do**
  $P_c, n_c, H_c \leftarrow \text{ClientReport}(c)$
Calculate global statistics: $P_g = \sum_{c \in C} P_c$, $n_g = \sum_{c \in C} n_c$
for each client $c \in C$ **do**
  Calculate local representativeness:

$$\nu_c = \gamma_{dv} \left| \frac{P_g}{n_g} - \frac{P_c}{n_c} \right| + \gamma_{sa} n_c^{-0.5} + \gamma_{tr} \theta_c,$$

Select top k clients to yield C_k

for each training round $t = 1, 2, \ldots, T$ **do**
  for each client $c \in C_k$ **do**
    $g_c \leftarrow \text{ClientUpdate}(c, w_t)$
  $\bar{g} \leftarrow \frac{1}{|C_k|} \sum_{c \in C_k} g_c$ ▷ Aggregate local updates
  $w_{t+1} \leftarrow$ Update global model using $\bar{g}$

procedure ClientReport(c)
  $P_c \leftarrow$ Target distribution at client c
  $n_c \leftarrow$ Number of data points at client c
  $H_c \leftarrow$ Computational capacity of client c
  return (P_c, n_c, H_c)

procedure ClientUpdate(c, w)
  $w_c \leftarrow$ Update local parameters with w
  for local epoch e from 1 to E **do**
    $g_c \leftarrow \nabla f_c(w_c)$ ▷ Local parameter update
  Send g_c back to the server

Abbreviations

AUC	Area Under the Receiver Operator Curve
BCE	Binary Cross Entropy
CXR	Chest Radiograph
DL	Deep Learning
FID	Fréchet Inception Distance
FL	Federated Learning
GDPR	General Data Protection Regulation
Grad-CAM	Gradient-weighted Class Activation Maps
GRU	Gated Recurrent Unit
HIPAA	Health Insurance Portability and Accountability Act
ICU	Intensive Care Unit
LoS	Length of Stay
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MSE	Mean Squared Error
MSLE	Mean Squared Logarithmic Error
Non-IID	Non Independent and Identically Distributed
RNN	Recurrent Neural Network
SSIM	Structural Similarity Index Measure

Acknowledgements
Not applicable.

Author' contributions
VS established the research outline for this work and is the main contributor for the development and implementation of the methodological approach. In addition, VS is a major contributor to the writing process of the resulting text. LNWM contributed mainly to the development of the client recruitment approach and experimental setup. Both WV and BDM contributed to shaping the scope, focus and structure of the text. All authors read and approved the final manuscript.

Funding
This work was supported by KU Leuven: Research Fund (projects C16/15/059, C3/19/053, C24/18/022, C3/20/117, C3I-21-00316), Industrial Research Fund (Fellowships 13-0260, IOFm/16/004, IOFm/20/002) and several Leuven Research and Development bilateral industrial projects; Flemish Government Agencies: FWO: EOS Project no G0F6718N (SeLMA), SBO project S005319N, Infrastructure project I013218N, TBM Project T001919N; PhD Grants (SB/1SA1319N, SB/1S93918, SB/1S1319N), EWI: the Flanders AI Research Program VLAIO: CSBO (HBC.2021.0076) Baekeland PhD (HBC.20192204) and Innovation mandate (HBC.2019.2209) European Commission: European Research Council under the European Union's Horizon 2020 research and innovation programme (ERC Adv. Grant grant agreement No 885682); Other funding: Foundation 'Kom op tegen Kanker', CM (Christelijke Mutualiteit)
We stipulate that the funding bodies played no role in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

Data availability
The datasets analysed during the current study are available in the Physionet repository:
MIMIC CXR with Structured Labels: https://doi.org/10.13026/8360-t248, eICU Collaborative Research Database: https://doi.org/10.13026/C2WM1R.
The code implementations, including instructions for reproducing the results are made publicly available on GitHub under the following repository: https://github.com/vscheltjens/eicu-cl-recr.

Declarations

Ethics approval and consent to participate
The study exempt from institutional review board approval due to the retrospective design, lack of direct patient intervention, and the security schema, for which the re-identification risk was certified as meeting safe harbor standards by an independent privacy expert (Privacert, Cambridge, MA) (Health Insurance Portability and Accountability Act Certification no. 1031219-2).

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 31 December 2023 Accepted: 29 November 2024
Published online: 18 December 2024

References

1. Bhardwaj R, Nambiar AR, Dutta D. A Study of Machine Learning in Healthcare. In: 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), vol. 2. 2017. pp. 236–241. https://doi.org/10.1109/COMPSAC.2017.164.
2. Shailaja K, Seetharamulu B, Jabbar MA. Machine Learning in Healthcare: A Review. In: 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA). 2018. pp. 910–914. https://doi.org/10.1109/ICECA.2018.8474918.
3. Callahan A, Shah NH. Chapter 19 - Machine Learning in Healthcare. In: Sheikh A, Cresswell KM, Wright A, Bates DW, editors. Key Advances in Clinical Informatics. Academic Press; 2017. pp. 279–291. https://doi.org/10.1016/B978-0-12-809523-2.00019-4.
4. European Commission. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance). European Commission; 2016. https://eur-lex.europa.eu/eli/reg/2016/679/oj.
5. Centers for Medicare & Medicaid Services. The Health Insurance Portability and Accountability Act of 1996 (HIPAA). 1996. http://www.cms.hhs.gov/hipaa/. Accessed 9 June 2023.
6. McMahan B, Moore E, Ramage D, Hampson S, Arcas BAy. Communication-Efficient Learning of Deep Networks from Decentralized Data. In: Singh A, Zhu J, editors. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 54. PMLR; 2017. pp. 1273–1282. https://proceedings.mlr.press/v54/mcmahan17a.html.
7. Rieke N, Hancox J, Li W, Milletarì F, Roth HR, Albarqouni S, et al. The future of digital health with federated learning. NPJ Digit Med. 2020;3(1):119. https://doi.org/10.1038/s41746-020-00323-1.
8. Li W, Milletari F, Xu D, Rieke N, Hancox J, Zhu W, et al. Privacy-preserving Federated Brain Tumour Segmentation. CoRR. 2019. arXiv:1910.00962.
9. Sheller MJ, Reina GA, Edwards B, Martin J, Bakas S. Multi-institutional Deep Learning Modeling Without Sharing Patient Data: A Feasibility Study on Brain Tumor Segmentation. In: Crimi A, Bakas S, Kuijf H, Keyvan F, Reyes M, van Walsum T, editors. Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. Cham: Springer International Publishing; 2019. pp. 92–104.
10. Xu J, Glicksberg BS, Su C, Walker P, Bian J, Wang F. Federated Learning for Healthcare Informatics. J Healthc Inform Res. 2021;5(1):1–19. https://doi.org/10.1007/s41666-020-00082-4.
11. Mehrjou A, Soleymani A, Buchholz A, Hetzel J, Schwab P, Bauer S. Federated Learning in Multi-Center Critical Care Research: A Systematic Case Study using the eICU Database. CoRR. 2022. arXiv:2204.09328.
12. Zhu H, Xu J, Liu S, Jin Y. Federated Learning on Non-IID Data: A Survey. CoRR. 2021. arXiv:2106.06843.
13. Xia W, Quek TQS, Guo K, Wen W, Yang HH, Zhu H. Multi-Armed Bandit-Based Client Scheduling for Federated Learning. IEEE Trans Wirel Commun. 2020;19(11):7108–23. https://doi.org/10.1109/TWC.2020.3008091.
14. Nishio T, Yonetani R. Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC). 2019. pp. 1–7. https://doi.org/10.1109/ICC.2019.8761315.
15. Xu J, Wang H. Client Selection and Bandwidth Allocation in Wireless Federated Learning Networks: A Long-Term Perspective. IEEE Trans Wirel Commun. 2021;20(2):1188–200. https://doi.org/10.1109/TWC.2020.3031503.

16. Zhang H, Xie Z, Zarei R, Wu T, Chen K. Adaptive Client Selection in Resource Constrained Federated Learning Systems: A Deep Reinforcement Learning Approach. IEEE Access. 2021;9:98423–32. https://doi.org/10.1109/ACCESS.2021.3095915.
17. Yoshida N, Nishio T, Morikura M, Yamamoto K. MAB-based Client Selection for Federated Learning with Uncertain Resources in Mobile Networks. In: 2020 IEEE Globecom Workshops GC Wkshps. 2020. pp. 1–6. https://doi.org/10.1109/GCWkshps50303.2020.9367421.
18. Cho YJ, Wang J, Joshi G. Client Selection in Federated Learning: Convergence Analysis and Power-of-Choice Selection Strategies. CoRR. 2020. arXiv:2010.01243.
19. Ma Z, Zhao M, Cai X, Jia Z. Fast-convergent federated learning with class-weighted aggregation. J Syst Archit. 2021;117: 102125. https://doi.org/10.1016/j.sysarc.2021.102125.
20. Karimireddy SP, Kale S, Mohri M, Reddi SJ, Stich SU, Suresh AT. SCAFFOLD: Stochastic Controlled Averaging for On-Device Federated Learning. CoRR. 2019. arXiv:1910.06378.
21. Sahu AK, Li T, Sanjabi M, Zaheer M, Talwalkar A, Smith V. On the Convergence of Federated Optimization in Heterogeneous Networks. CoRR. 2018. arXiv:1812.06127.
22. Ruan Y, Zhang X, Joe-Wong C. How Valuable Is Your Data? Optimizing Client Recruitment in Federated Learning. In: 2021 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt). 2021. pp. 1–8. https://doi.org/10.23919/WiOpt52861.2021.9589776.
23. Scheltjens V, Momo LNW, Verbeke W, De Moor B. Client Recruitment for Federated Learning in ICU Length of Stay Prediction. In: 2023 IEEE 19th International Conference on e-Science (e-Science). 2023. pp. 1–9. https://doi.org/10.1109/e-Science58273.2023.10254908.
24. Pollard TJ, Johnson AEW, Raffa JD, Celi LA, Mark RG, Badawi O. The eICU Collaborative Research Database, a freely available multi-center database for critical care research. Sci Data. 2018;5(1): 180178. https://doi.org/10.1038/sdata.2018.178.
25. Pollard TJ, Johnson AEW, Raffa J, Badawi O. The eICU Collaborative Research Database. physionet.org; 2017. https://doi.org/10.13026/C2WM1R.
26. Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, et al. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation. 2000;101(23):E215–20.
27. Johnson AEW, Pollard TJ, Berkowitz SJ, Greenbaum NR, Lungren MP, Deng C, et al. MIMIC-CXR: A large publicly available database of labeled chest radiographs. CoRR. 2019. arXiv:1901.07042
28. Homem-de Mello T. On Rates of Convergence for Stochastic Optimization Problems Under Non-Independent and Identically Distributed Sampling. SIAM J Optim. 2008;19(2):524–51. https://doi.org/10.1137/060657418.
29. Shapiro A. Monte Carlo Sampling Methods. In: Stochastic Programming. Handbooks in Operations Research and Management Science, vol. 10. Elsevier; 2003. pp. 353–425. https://doi.org/10.1016/S0927-0507(03)10006-0.
30. Rocheteau E, Tong C, Velickovic P, Lane ND, Liò P. Predicting Patient Outcomes with Graph Representation Learning. CoRR. 2021. arXiv:2101.03940.
31. Al-Dailami A, Kuang H, Wang J. Predicting length of stay in ICU and mortality with temporal dilated separable convolution and context-aware feature fusion. Comput Biol Med. 2022;151(Pt A):106278.
32. Rocheteau E, Liò P, Hyland S. Temporal Pointwise Convolutional Networks for Length of Stay Prediction in the Intensive Care Unit. In: Proceedings of the Conference on Health, Inference, and Learning. CHIL '21. New York: Association for Computing Machinery; 2021. pp. 58–68. https://doi.org/10.1145/3450439.3451860.
33. Vandenberghe A, Wamba Momo LN, Scheltjens V, De Moor B. Multimodal Deep Learning for Early Length of Stay Prediction using Patient Similarity Embeddings. In: Proc. of BNAIC/BeNeLearn. Mechelen; 2022.
34. Cho K, van Merrienboer B, Gülçehre Ç, Bougares F, Schwenk H, Bengio Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. CoRR. 2014. arXiv:1406.1078.
35. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Comput. 1997;9(8):1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.
36. Loshchilov I, Hutter F. Fixing Weight Decay Regularization in Adam. CoRR. 2017. arXiv:1711.05101.
37. Jaderberg M, Dalibard V, Osindero S, Czarnecki WM, Donahue J, Razavi A, et al. Population Based Training of Neural Networks. CoRR. 2017. arXiv:1711.09846.
38. He C, Li S, So J, Zhang M, Wang H, Wang X, et al. FedML: A Research Library and Benchmark for Federated Machine Learning. Advances in Neural Information Processing Systems, Best Paper Award at Federate Learning Workshop. 2020.
39. Irvin J, Rajpurkar P, Ko M, Yu Y, Ciurea-Ilcus S, Chute C, et al. CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison. CoRR. 2019. arXiv:1901.07031.
40. Huang G, Liu Z, Weinberger KQ. Densely connected convolutional networks. CoRR. 2016. arXiv:1608.06993.
41. Selvaraju RR, Das A, Vedantam R, Cogswell M, Parikh D, Batra D. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization. CoRR. 2016. arXiv:1610.02391.
42. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process. 2004;13(4):600–12. https://doi.org/10.1109/TIP.2003.819861.
43. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Klambauer G, Hochreiter S. GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium. CoRR. 2017. arXiv:1706.08500.

Publisher's Note