# Privacy-Preserving and Bandwidth-Efficient Federated Learning: An Application to In-Hospital Mortality Prediction

Raouf Kerkouche, Gergely Acs, Claude Castelluccia, Pierre Genevès

# Privacy-Preserving and Bandwidth-Efficient Federated Learning: An Application to In-Hospital Mortality Prediction

Raouf Kerkouche
raouf.kerkouche@inria.fr
Privatics team, Univ. Grenoble Alpes, Inria

Gergely Ács
acs@crysys.hu
Crysys Lab, BME-HIT

Claude Castelluccia
claude.castelluccia@inria.fr
Privatics team, Univ. Grenoble Alpes, Inria

Pierre Genevès
pierre.geneves@cnrs.fr
Tyrex team, Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG

## ABSTRACT

Machine Learning, and in particular Federated Machine Learning, opens new perspectives in terms of medical research and patient care. Although Federated Machine Learning improves over centralized Machine Learning in terms of privacy, it does not provide provable privacy guarantees. Furthermore, Federated Machine Learning is quite expensive in term of bandwidth consumption as it requires participant nodes to regularly exchange large updates. This paper proposes a bandwidth-efficient privacy-preserving Federated Learning that provides theoretical privacy guarantees based on Differential Privacy. We experimentally evaluate our proposal for in-hospital mortality prediction using a real dataset, containing Electronic Health Records of about one million patients. Our results suggest that strong and provable patient-level privacy can be enforced at the expense of only a moderate loss of prediction accuracy.

## 1 INTRODUCTION

An Electronic Health Record (EHR) is a digital version of the patient's medical information. EHR data open new perspectives, especially with the development of machine learning. EHR data can be used to train predictive models in order to predict patient's medical conditions and help medical doctors to develop appropriate care [18, 36].

However, medical data is considered as sensitive information that can lead to some real and serious damage to the patient if any leakage happens. For example, medical data can be exploited by insurance companies to adapt their insurance fees, by banks to deny loans, or by politicians to discredit their opponents. Therefore, the privacy of such kind of sensitive data must be guaranteed and privacy-preserving predictive models are needed.

Predictive models are typically built using machine learning algorithms that are trained on centralized datasets. When a model is trained on multiple datasets, collected for example by several hospitals, the centralization of all datasets on a single server introduces additional, and often unacceptable, privacy risks. To mitigate this problem, Federated learning (FL) was proposed as a new learning protocol. Federated Learning consists of distributing the learning process on the different entities providing data: instead of aggregating the data on a single server, the training is performed locally by each participating entities and the models are then shared and aggregated [27, 38]. Although Federated Learning mitigates the privacy risks by design, recent results have shown that some attacks, such as membership and property inference attacks, are still possible [29, 33]. Moreover, complete training samples can also be reconstructed purely from the captured gradients [43, 44].

Furthermore, since participating entities must collaborate by exchanging their model updates, the required bandwidth during the training phase is often significant and prohibitive [22].

*Contribution.* This paper proposes a bandwidth-efficient privacy-preserving Federated Learning scheme that provides theoretical privacy guarantees. Our proposal guarantees Differential Privacy with practical utility even on highly imbalanced training data. This is challenging as imbalanced data increases the injected noise required by Differential Privacy and hence substantially degrades model quality. Our solution relies on the extreme quantization of the gradients in order to reduce communication costs as well as on downsampling of mini-batches to diminish the noise needed for Differential Privacy. We experimentally evaluate the performance of our solution for in-hospital mortality prediction using real EHR data, containing about one million records of patients. Our results suggest that patient-level privacy can be enforced at the expense of only a moderate loss of prediction accuracy.

*Outline.* We describe the background in Section 2. We introduce our privacy-preserving scheme in Section 3. We report on experiments with real-world data in Section 4. Finally we discuss related works in Section 5 before concluding in Section 7.

## 2 BACKGROUND

### 2.1 Federated Learning (FL-STANDARD)

In federated learning [27, 38], multiple parties (clients) build a common machine learning model on the union of their training data without sharing them with each other. At each round of the training, some clients retrieve the global model from the parameter server, update the global model based on their own training data, and send back their updated model to the server. The server aggregates the updated models of all clients to obtain a global model that is re-distributed to some selected parties in the next round.

In particular, a subset $\mathbb{K}$ of all $N$ clients are randomly selected at each round to update the global model, and $C = |\mathbb{K}|/N$ denotes the fraction of selected clients. At round $t$, a selected client $k \in \mathbb{K}$ executes $T_{\text{gd}}$ local gradient descent iterations on the common model $\mathbf{w}_{t-1}$ using its own training data $D_k$ ($D = \cup_{k\in\mathbb{K}} D_k$), and obtains the updated model $\mathbf{w}_t^k$, where the number of weights is denoted by $n$ (i.e., $|\mathbf{w}_t^k| = |\Delta\mathbf{w}_t^k| = n$ for all $k$ and $t$). Each client $k$ submits the update $\Delta\mathbf{w}_t^k = \mathbf{w}_t^k - \mathbf{w}_{t-1}^k$ to the server, which then updates the common model as follows: $\mathbf{w}_t = \mathbf{w}_{t-1} + \sum_{k\in\mathbb{K}} \frac{|D_k|}{\sum_j |D_j|} \Delta\mathbf{w}_t^k$, where $|D_k|$ is known to the server for all $k$ (a client's update is weighted with the size of its training data). The server stops training after a fixed number of rounds $T_{\text{cl}}$, or when the performance of the common model does not improve on a held-out data.

Note that each $D_k$ may be generated from different distributions (i.e., Non-IID case), that is, any client's local dataset may not be representative of the population distribution [27]. This can happen, for example, when not all output classes are represented in every client's training data. The federated learning of neural networks is summarized in Alg. 1. In the sequel, each client is assumed to use the same model architecture.

**Algorithm 1:** FL-STANDARD: Federated Learning

**Server:**
  Initialize common model $w_0$
  **for** $t = 1$ **to** $T_{\text{cl}}$ **do**
    Select $\mathbb{K}$ clients uniformly at random
    **for** *each client* $k$ *in* $\mathbb{K}$ **do**
      $\Delta\mathbf{w}_t^k = \textbf{Client}_k(\mathbf{w}_{t-1})$
    **end**
    $\mathbf{w}_t = \mathbf{w}_{t-1} + \sum_k \frac{|D_k|}{\sum_j |D_j|} \Delta\mathbf{w}_t^k$
  **end**
  **Output:** Global model $\mathbf{w}_t$

**Client**$_k(\mathbf{w}_{t-1}^k)$**:**
  $\mathbf{w}_t^k = \text{SGD}(D_k, \mathbf{w}_{t-1}^k, T_{\text{gd}})$
  **Output:** Model update $(\mathbf{w}_t^k - \mathbf{w}_{t-1}^k)$

The motivation of federated learning is three-fold: first, it aims to provide confidentiality of each participant's training data by sharing only model updates instead of potentially sensitive training data. Second, in order to decrease communication costs, clients can perform multiple local SGD iterations before sending their update back to the server. Third, in each round, only a few clients are required to perform local training of the common model, which further diminishes communication costs and makes the approach especially appealing with a large number of clients.

**Algorithm 2:** Stochastic Gradient Descent

**Input:** $D$ : training data, $T_{\text{gd}}$ : local epochs, $\mathbf{w}$ : weights
**for** $t = 1$ **to** $T_{\text{gd}}$ **do**
  Select batch $\mathbb{B}$ from $D$ randomly
  $\mathbf{w} = \mathbf{w} - \eta \nabla f(\mathbb{B}; \mathbf{w})$
**end**
**Output:** Model $\mathbf{w}$

However, several prior works have demonstrated that model updates do leak potentially sensitive information [29, 33]. Hence, simply not sharing training data *per se* is not enough to guarantee their confidentiality.

### 2.2 Differential Privacy

Differential privacy allows a party to privately release information about a dataset: a function of an input dataset is perturbed, so that any information which can differentiate a record from the rest of the dataset is bounded [17].

*Definition 2.1 (Privacy loss).* Let $\mathcal{A}$ be a privacy mechanism which assigns a value in $Range(\mathcal{A})$ to a dataset $D$. The privacy loss of $\mathcal{A}$ with datasets $D$ and $D'$ at output $O \in Range(\mathcal{A})$ is a random variable $\mathcal{P}(\mathcal{A}, D, D', O) = \log \frac{\Pr[\mathcal{A}(D)=O]}{\Pr[\mathcal{A}(D')=O]}$ where the probability is taken on the randomness of $\mathcal{A}$.

*Definition 2.2 ($(\epsilon, \delta)$-Differential Privacy [17]).* A privacy mechanism $\mathcal{A}$ guarantees $(\varepsilon, \delta)$-differential privacy if for any database $D$ and $D'$, differing on at most one record, $\Pr_{O\sim\mathcal{A}(D)}[\mathcal{P}(\mathcal{A}, D, D', O) > \varepsilon] \leq \delta$.

Intuitively, this guarantees that an adversary, provided with the output of $\mathcal{A}$, can draw almost the same conclusions (up to $\varepsilon$ with probability larger than $1 - \delta$) about any record no matter if it is included in the input of $\mathcal{A}$ or not [17]. That is, for any record owner, a privacy breach is unlikely to be due to its participation in the dataset.

*Moments Accountant.* Differential privacy maintains composition; the privacy guarantee of the $k$-fold adaptive composition of $\mathcal{A}_{1:k} = \mathcal{A}_1, \ldots, \mathcal{A}_k$ can be computed using the moments accountant method [2]. In particular, it follows from Markov's inequality that $\Pr[\mathcal{P}(\mathcal{A}, D, D', O) \geq \varepsilon] \leq \mathbb{E}[\exp(\lambda\mathcal{P}(\mathcal{A}, D, D', O))]/\exp(\lambda\varepsilon)$ for any output $O \in Range(\mathcal{A})$ and $\lambda > 0$. This implies that $\mathcal{A}$ is $(\varepsilon, \delta)$-DP with $\delta = \min_\lambda \exp(\alpha_{\mathcal{A}}(\lambda) - \lambda\varepsilon)$, where $\alpha_{\mathcal{A}}(\lambda) = \max_{D,D'} \log \mathbb{E}_{O\sim\mathcal{A}(D)}[\exp(\lambda\mathcal{P}(\mathcal{A}, D, D', O))]$ is the log of the moment generating function of the privacy loss. The privacy guarantee of the composite mechanism $\mathcal{A}_{1:k}$ can be computed using that $\alpha_{\mathcal{A}_{1:k}}(\lambda) \leq \sum_{i=1}^k \alpha_{\mathcal{A}_i}(\lambda)$ [2].

*Gaussian Mechanism.* There are a few ways to achieve DP, including the Gaussian mechanism [17]. A fundamental concept of all of them is the *global sensitivity* of a function [17].

*Definition 2.3 (Global $L_p$-sensitivity).* For any function $f : \mathcal{D} \rightarrow \mathbb{R}^n$, the $L_p$-sensitivity of $f$ is $\Delta_p f = \max_{D,D'} ||f(D) - f(D')||_p$, for all $D, D'$ differing in at most one record, where $|| \cdot ||_p$ denotes the $L_p$-norm.

The Gaussian Mechanism [17] consists of adding Gaussian noise to the true output of a function. In particular, for any function $f : \mathcal{D} \rightarrow \mathbb{R}^n$, the Gaussian mechanism is defined as adding i.i.d Gaussian noise with variance $(\Delta_2 f \cdot \sigma)^2$ and zero mean to each coordinate value of $f(D)$. Recall that the pdf of the Gaussian distribution with mean $\mu$ and variance $\xi^2$ is

$$\text{pdf}_{\mathcal{G}(\mu,\xi)}(x) = \frac{1}{\sqrt{2\pi}\xi} \exp\left(-\frac{(x-\mu)^2}{2\xi^2}\right) \quad (1)$$

In fact, the Gaussian mechanism draws vector values from a multivariate spherical (or isotropic) Gaussian distribution which is described by random variable $\mathcal{G}(f(D), \Delta_2 f \cdot \sigma \mathbf{I}_n)$, where $n$ is omitted if its unambiguous in the given context.

# 3 TOWARD FEDERATED LEARNING RECORD-LEVEL PRIVACY

## 3.1 The FL-SIGN Protocol

In the FL-STANDARD scheme, presented in Section 2.1, each selected client sends its updated model to the central server. As discussed previously, this scheme has several drawbacks in terms of bandwidth and privacy. We propose to limit these drawbacks by quantizing the model weights as in [9, 21]. More specifically, in the new scheme, referred to as FL-SIGN in the rest of this paper, each client sends only the sign of every coordinate value in its parameter update vector. The server takes the sign of the sum of signs per coordinate and scales down the result with a fixed constant $\gamma$ (which is in the order of $10^{-3}$ in practice) in order to limit the contribution of each client and adjust convergence. This scaled aggregated updates are added to the global model.

**Algorithm 3:** FL-SIGN: Sign Federated Learning

**Server:**
  Initialize common model $w_0$
  **for** $t = 1$ **to** $T_{\text{cl}}$ **do**
    Select $\mathbb{K}$ clients uniformly at random
    **for** *each client* $k$ *in* $\mathbb{K}$ **do**
      $\mathbf{s}_t^k = \textbf{Client}_k(\mathbf{w}_{t-1})$
    **end**
    $\mathbf{w}_t = \mathbf{w}_{t-1} + \gamma \text{sign}\left(\sum_k \mathbf{s}_t^k\right)$
  **end**
  **Output:** Global model $\mathbf{w}_t$

**Client**$_k(\mathbf{w}_{t-1}^i)$**:**
  $\mathbf{w}_t^k = \text{SGD}(D_k, \mathbf{w}_{t-1}^k, T_{\text{gd}})$
  **Output:** Model update $\text{sign}(\mathbf{w}_t^k - \mathbf{w}_{t-1}^k)$

More specifically, FL-SIGN (see Alg. 3) differs from the standard federated scheme FL-STANDARD (see Alg. 1) as follows:

(1) Each client returns $\mathbf{s}_t^k = \text{sign}(\mathbf{w} - \mathbf{w}_{t-1}^k)$ instead of $(\mathbf{w} - \mathbf{w}_{t-1}^k)$, where $\text{sign} : \mathbb{R}^n \rightarrow \{-1, 1\}^n$ returns the sign of each coordinate value of the input vector if it is non-zero and a sign chosen uniformly at random otherwise.
(2) The server sums the sign vectors $\mathbf{s}_t^k$ sent by each client $k$ and computes the sign vector of this sum as $\text{sign}\left(\sum_k \mathbf{s}_t^k\right)$. This is equivalent to take the median of all clients' signs at every position of the update vectors. Unlike in Alg. 1, the update $\mathbf{s}_t^k$ is *not* weighted with client $k$'s data size $|D_k|$, since that would require the client to send $|D_k|$ to the server which would enable the adversary to maliciously scale up its sign vector by sending a fabricated size of its training data.

The extreme quantization performed by FL-SIGN reduces the communication costs of federated learning by a factor of 32 (since only one bit is sent per parameter instead of 32 bits). Note also that, if the quantized update vector is sparse, other lossless or lossy compression techniques can further improve communication efficiency [22].

## 3.2 Privacy-Preserving FL-SIGN (FL-SIGN-DP)

In FL-SIGN, a participant only sends the signs of its updates, as opposed to their actual values, hence it intuitively reveals less information about the client's dataset than the original FL-STANDARD scheme. In order to experimentally validate this intuition, we implemented the inference attack described in [29] on FL-STANDARD and FL-SIGN[1]. Results showed that the attack accuracy dropped from 92% for FL-STANDARD to 50% for FL-SIGN. While these results suggest that privacy could be preserved in practice, they do not provide any strong guarantee.

To reason about the general privacy guarantee of FL-SIGN more rigorously, consider the sign vector $\mathbf{s}_t^k = \text{sign}(\mathbf{w}_t^k - \mathbf{w}_{t-1}^k)$. Several attacks have demonstrated [29, 33] that $\Delta\mathbf{w}_t^k = \mathbf{w}_t^k - \mathbf{w}_{t-1}^k$ can be used to infer the membership of individual records in the training data due to the strong memorization property of neural networks, and overfitting in general. As taking the sign of $\Delta\mathbf{w}_t^k$ is a deterministic operation and depends on the value of $\mathbf{s}_t^k$, there is no guarantee that $\mathbf{s}_t^k$ does not leak any sensitive information.

In order to obtain theoretically private schemes, we extend FL-SIGN with Differential Privacy. Our goal is to design a differentially private scheme that is accurate and also bandwidth efficient (even for small $\varepsilon$ values).

*3.2.1 Privacy and Adversarial Models.* We consider an adversary, or a set of colluding adversaries, who can access any update vector sent by the server or any clients at each round of the protocol. The adversary is computationally unbounded but *passive* (i.e., honest-but-curious), that is, it follows the learning protocol faithfully and does not modify any update vector. A plausible adversary is a participating entity, i.e. a malicious client or server, that wants to infer the training data used by other participants.

We aim at developing a solution that protects each record of the clients' training datasets. For example, in the scenario of collaborating hospitals we aim at protecting each individual patient record of all hospital datasets.

[1] A model was trained for gender classification on the LFW dataset. The adversary's goal is to infer from the model updates whether a specific group of individuals in a client's dataset are black.

**Algorithm 4:** FL-SIGN-DP: Bandwidth-Efficient Federated Learning with Differential Privacy

**Server:**
  Initialize common model $w_0$
  **for** $t = 1$ **to** $T_{\text{cl}}$ **do**
    Select $\mathbb{K}$ clients randomly
    **for** *each client* $k$ *in* $\mathbb{K}$ **do**
      $\mathbf{s}_t^k = \textbf{Client}_k(\mathbf{w}_{t-1})$
    **end**
    $\mathbf{w}_t = \mathbf{w}_{t-1} + \gamma \text{sign}\left(\sum_k \mathbf{s}_t^k\right)$
  **end**
**Client**$_k(\mathbf{w}_{t-1}^k)$**:**
  $\tilde{\mathbf{w}}_t^k = \textbf{DPSGD}(D_k, \mathbf{w}_{t-1}^k, S, \sigma, T_{\text{gd}})$
  **Output:** $\text{sign}(\tilde{\mathbf{w}}_t^k - \mathbf{w}_{t-1}^k)$

**Algorithm 5:** FL-STANDARD-DP: Federated Learning with Differential Privacy

**Server:**
  Initialize common model $w_0$
  **for** $t = 1$ **to** $T_{\text{cl}}$ **do**
    Select $\mathbb{K}$ clients randomly
    **for** *each client* $k$ *in* $\mathbb{K}$ **do**
      $\mathbf{s}_t^k = \textbf{Client}_k(\mathbf{w}_{t-1})$
    **end**
    $\mathbf{w}_t = \mathbf{w}_{t-1} + \frac{1}{|\mathbb{K}|}\left(\sum_k \mathbf{s}_t^k\right)$
  **end**
**Client**$_k(\mathbf{w}_{t-1}^k)$**:**
  $\tilde{\mathbf{w}}_t^k = \textbf{DPSGD}(D_k, \mathbf{w}_{t-1}^k, S, \sigma, T_{\text{gd}})$
  **Output:** $\tilde{\mathbf{w}}_t^k - \mathbf{w}_{t-1}^k$

**Algorithm 6:** DPSGD$(D, \mathbf{w}, S, \sigma, T_{\text{gd}})$

**Input:** $D$ : training data, $T_{\text{gd}}$ : number of iterations, $\mathbf{w}$ : weights, $S$ : clipping threshold, $\sigma$ : noise scale
**for** $t = 1$ **to** $T_{\text{gd}}$ **do**
  Select batch $\mathbb{B}$ from $D$ randomly
  $\mathbb{B}' = \textbf{Downsampling}(\mathbb{B})$
  **for** *each record* $r$ *in* $\mathbb{B}'$ **do**
    $\nabla\hat{f}(r, \mathbf{w}) = \nabla f(r, \mathbf{w}) / \max\left(1, \frac{||\nabla f(r,\mathbf{w})||_2}{S}\right)$
  **end**
  $\mathbf{w} = \mathbf{w} - (\eta/|\mathbb{B}'|)\left(\sum_{r\in\mathbb{B}'} \nabla\hat{f}(r;\mathbf{w}) + \mathcal{G}(0, \sigma S\mathbf{I})\right)$
**end**
**Output:** Model parameters $\mathbf{w}$

**Algorithm 7:** **Downsampling**$(\mathbb{B})$

**Input:** Batch $\mathbb{B}$ with labels $L_1$ and $L_2$
Partition $\mathbb{B}$ into $\mathbb{C}_1$ and $\mathbb{C}_2$, where all samples in $\mathbb{C}_1$ has label $L_1$ and all samples in $\mathbb{C}_2$ has label $L_2$
$s_{min} = \min(|\mathbb{C}_1|, |\mathbb{C}_2|)$
$\mathbb{B}_1 \leftarrow$ select $s_{min}$ samples from $\mathbb{C}_1$ uniformly at random
$\mathbb{B}_2 \leftarrow$ select $s_{min}$ samples from $\mathbb{C}_2$ uniformly at random
**Output:** Balanced batch $\mathbb{B}_1 \cup \mathbb{B}_2$

The adversary should not be able to learn from the received model or its updates whether any particular record was used to train the model by any other participants.

We believe that this adversarial model is reasonable for the medical application that we consider in this paper: it is very unlikely that a participating hospital will take the risk of manipulating the updates that it sends to the server. However, we want to make sure that it can not infer any sensitive information from the models that it receives from the server. In other words, we make the assumption that hospitals may be "curious", but are "honest".

We use Differential Privacy (DP) because it was proposed to achieve this goal. DP guarantees plausible deniability. Therefore, any negative privacy impact on an individual, i.e. a patient in the dataset, cannot be attributed to his involvement in the training phase (up to $\varepsilon$ and $\delta$). For example, if an insurance company accesses the model updates or the common model and decides to increase the price of a patient's insurance fee, it cannot be because of the patient's data.

*3.2.2 Operation.* To guarantee differential privacy for any individual record of a training data, we propose FL-SIGN-DP, depicted in Alg. 4, which is a synergy of FL-SIGN and differentially private gradient descent (DPSGD) from [2]. In particular, instead of running traditional SGD on its local training data, every client executes DPSGD (depicted in Alg. 6), which guarantees that its output $\tilde{\mathbf{w}}_t^k$ does not leak any information that is specific to a single training sample (up to $\varepsilon$ and $\delta$) by clipping the $L_2$-norm of the gradients and perturbing the result with Gaussian noise. The noise scale is calibrated to $S$ and $\sigma$, where the latter directly gives $\varepsilon$ and $\delta$ as shown below. Hence, any further computation which uses $\tilde{\mathbf{w}}_t^k$ is also differentially private.

Notice that the batch is created using *downsampling* [19, 31] (see Alg. 7) in order to overcome the imbalanced classes of the training data. Downsampling guarantees that every batch contains identical number of samples from every class, and therefore they have similar magnitude of gradients on average.

Likewise FL-SIGN, FL-SIGN-DP also sends only signs for aggregation, and hence is equally bandwidth efficient.

*3.2.3 Privacy analysis.* The privacy guarantee of FL-SIGN-DP is quantified using the moments accountant method from [2]. Let $\eta_0(x|\xi, q) = \text{pdf}_{\mathcal{G}(0,\xi)}(x)$ and $\eta_1(x|\xi, q) = (1-q)\text{pdf}_{\mathcal{G}(0,\xi)}(x) + q\text{pdf}_{\mathcal{G}(1,\xi)}(x)$ where $q$ is the sampling probability of a single record in a single round. Let

$$\alpha_{\mathcal{G}}(\lambda|q) = \log\max(E_1(\lambda, \xi, q), E_2(\lambda, \xi, q)) \tag{2}$$

where

$$E_1(\lambda, \xi, q) = \int_{\mathbb{R}} \eta_0(x|\xi, q) \cdot \left(\frac{\eta_0(x|\xi, q)}{\eta_1(x|\xi, q)}\right)^{\lambda} dx$$

and

$$E_2(\lambda, \xi, q) = \int_{\mathbb{R}} \eta_1(x|\xi, q) \cdot \left(\frac{\eta_1(x|\xi, q)}{\eta_0(x|\xi, q)}\right)^{\lambda} dx$$

Theorem 3.1 (Privacy of FL-SIGN-DP). *For any $\delta > 0$, FL-SIGN-DP is $(\min_\lambda(T_{\text{cl}}\cdot\alpha_{\mathcal{G}}(\lambda|q_1) + T_{\text{cl}}\cdot(T_{\text{gd}}-1)\cdot\alpha_{\mathcal{G}}(\lambda|q_2) - \log\delta)/\lambda, \delta)$-DP, where $\alpha_{\mathcal{G}}$ is defined in Eq. (2), $q_1 = \frac{C\cdot|\mathbb{B}|}{\min_k|D_k|}$, and $q_2 = \frac{|\mathbb{B}|}{\min_k|D_k|}$.*

| | |
|---|---|
| FL-SIGN | $O\left(\frac{1}{\sqrt{T_{\text{cl}}CN}}\right)$ |
| FL-SIGN-DP | $O\left(\frac{nS\sigma}{\sqrt{T_{\text{cl}}CN}}\right)$ |

**Table 1: Convergence rates when $\gamma = O(1/\sqrt{T_{\text{cl}}})$, $T_{\text{gd}} = 1$, $|\mathbb{B}| = T_{\text{cl}}$**

The proof follows from Theorem 2 in [2] and the fact that a record is sampled in the very first SGD iteration of every round if (1) the corresponding client is sampled, which has a probability of $C$, and (2) the batch sampled locally at this client contains the record, which has a probability of at most $\frac{|\mathbb{B}|}{\min_k |D_k|}$. However, the adaptive composition of consecutive SGD iterations are considered where the output of a single iteration depends on the output of the previous iterations. Therefore, the sampling probability for the very first batch is $q_1 = \frac{C \cdot |\mathbb{B}|}{\min_k |D_k|}$, while the sampling probability for every subsequent SGD iteration within the same round is at most $q_2 = \frac{|\mathbb{B}|}{\min_k |D_k|}$ *conditioned* on the result of the first iteration (see the proof of Theorem 2 in [2]).

Given a fixed value of $\delta$, $\varepsilon$ is computed numerically as in [2, 30].

*3.2.4 Convergence analysis.* In Appendix A.1, we analytically compute that FL-SIGN-DP has a convergence rate of $O\left(\frac{nS\sigma}{\sqrt{T_{\text{cl}}CN}}\right)$. Compared to FL-SIGN (see Table 1), the convergence rate is increased with a factor of $nS\sigma$ which is attributed to the Gaussian noise and can be considered as the "cost of privacy".

# 4 EXPERIMENTAL RESULTS

The goal of this section is to evaluate the performance of our proposed FL-SIGN-DP scheme on a realistic in-hospital mortality prediction scenario. We aim at evaluating its performance with different levels of privacy (i.e. different values of $\epsilon$) and comparing it with the performance of the following learning protocols:

- *(Non-federated) CENTRALIZED training*: The training data of all hospitals are merged and a single model is trained on this merged data without any privacy guarantee.
- *FL-STANDARD* is described in Section 2.1.
- *FL-SIGN* is described in Section 3.1.
- *FL-STANDARD-DP* is specified in Alg. 5. It has the same privacy guarantee as FL-SIGN-DP[2] but is less bandwith efficient. Specifically, unlike in FL-SIGN-DP, each client sends the original (non-quantized) update vector $\mathbf{s}_t^k = \mathbf{w}_t^k - \mathbf{w}_{t-1}^k$ to the server, which computes the model update as $\mathbf{w}_t = \mathbf{w}_{t-1} + \frac{1}{|\mathbb{K}|}\left(\sum_k \mathbf{s}_t^k\right)$. Both FL-SIGN-DP and FL-STANDARD-DP use downsampling (in Alg. 7) to create batches.

In order to improve the reproducibility of our results, we published all the code used in our experiments [3].

[2] the privacy analysis in Section 3.2.3 also applies to FL-STANDARD-DP

[3] https://github.com/raouf-kerkouche/Privacy-preserving-and-Bandwith-Efficient-Federated-Learning-An-Application-to-In-Hospital-Mortality

## 4.1 The In-hospital Mortality Prediction Scenario

The ability to accurately predict the risks in the patient's perspectives of evolution is a crucial prerequisite in order to adapt the care that certain patients receive [18].

We consider the scenario where several hospitals are collaborating to train models for in-hospital mortality prediction using our Federated Learning schemes. This well-studied real-world problem consists in trying to precisely identify the patients who are at risk of dying from complications during their hospital stay [5, 18, 36]. As commonly found in the literature [18], for such predictions, we focus on hospital admissions of adults hospitalized for at least 3 days, excluding elective admissions.

## 4.2 The Premier Healthcare Database

We used EHR data from the Premier healthcare database[4] which is one of the largest clinical databases in the United States, collecting information from millions of patients over a period of 12 months from 415 hospitals in the USA [18]. These hospitals are supposedly representative of the United States hospital experience [18]. Each hospital in the database provides discharge files that are dated records of all billable items (including therapeutic and diagnostic procedures, medication, and laboratory usage) which are all linked to a given patient's admission [18, 24].

The initial snapshot of the database used in our work (before pre-processing step) comprises the EHR data of 1,271,733 hospital admissions. Electronic Health Record (EHR) is a digital version of a patient's paper chart readily available in hospitals. For developing supervised learning and specifically deep learning models, we focus on a specific set of features from EHR data. The features of interest that capture the patients information are summarized in Table 2. There is a total of 24,428 features per patient, mainly due to the variety of drugs possibly served.

The Medication regimen complexity index (MRCI) [26] is an aggregate score computed from a total of 65 items, whose purpose is to indicate the complexity of the patient's situation. The minimum MRCI score for a patient is 1.5, which represents a single tablet or capsule taken once a day as needed (single medication). However the maximum is not defined since the number of medications increases the score [26]. In our case, after statistical analysis of our dataset, we consider the MRCI score as ranging from 2 to 60.

Most real datasets like ours are generally imbalanced with a skewed distribution between the classes. In our case, the positive cases (patients who die during their hospital stay) represent only 3% of all patients. Table 3 gives more details about this distribution after the pre-processing step which is discussed in 4.3.1.

## 4.3 Data pre-processing & experimental setup

This section describes the experimental setting which is used to evaluate the accuracy and the privacy of our proposals.

*4.3.1 Preprocessing.*

(1) **Features normalization**: we extract from the dataset the values of each feature represented in Table 2. For gender, we

[4] https://www.premierinc.com/newsroom/education/premier-healthcare-database-whitepaper

**Table 2: Descriptions of features**

| *Features* | *Descriptions* |
|---|---|
| Age | Value in the range of 15 and 89 |
| Gender | Male, Female or Unknown |
| Admission type | Emergency, Urgent, Trauma Center: visits to a trauma center/hospital or Unknown |
| MRCI | Medication regimen complexity index score (ranging from 2 to 60) |
| Drugs and ICD9 codes | Drugs given to the patient on the $1^{st}$ day of hospitalization. The ICD9 codes [16] are composed of procedures and diagnosis codes, the first gives details about the medical procedures performed on the patient and the second about the doctor's diagnosis of the patient. There is a total of 24,419 possible drugs and ICD9 codes. |

**Table 3: Number of instances for our case study.**

| Data | Positive cases | Negative cases | Ratio | Total |
|---|---|---|---|---|
| Train | 30,775 | 947,152 | 3.15% | 977,927 |
| Test | 7,891 | 236,736 | 3.23% | 244,627 |

**Table 4: Statistics on the size of the training and testing data over all the clients**

| Data | Min | Max | Mean | Std |
|---|---|---|---|---|
| Train | 804 | 12,447 | 3,114.42 | 1,913.39 |
| Test | 201 | 3,112 | 779.07 | 478.39 |

use one-hot encoding: Male, Female and Unknown. Similarly, for admission type we use 4 features: Emergency, Urgent, Trauma Center, and Unknown[5]. For drugs, we extract 24,419 features which correspond to the different drugs (name and dosage). A given patient receives only a few of the possible drugs served, resulting in a very sparse patient's record. We use a MinMax normalization for age and MRCI in order to rescale the values of these features between 0 and 1 (using MinMaxScaler class of scikit-learn[6]). The labels that we consider are boolean: true means that the patient died during his hospital stay while false means she survived.

(2) **Hospitals filtering**: The dataset contains 415 hospitals, however, we choose to consider only hospitals with at least 1,000 patients, which results at the end in 314 hospitals. The reason is to have enough data per hospital for both training and testing. We split randomly the dataset of each hospital into disjoint training and testing data (80% and 20% respectively). We merge the test data of all hospitals for the evaluation, which we consider fairer than averaging the metrics over all the clients (hospitals). The final dataset for testing contains 244,627 patients, with 7,891 deceased patients and 236,736 non-deceased patients (see Table 3). The statistics on the size of the clients' dataset are depicted in Table 4.

(3) **Patients filtering**: We consider patient and drug information of the first day at the hospital so that we can make predictions 24 hours after admission (as commonly found in the literature [18, 36]). We filter out the pregnant and new-born patients because the medication types and admission services are not the same for theses two categories of patients. Our model prediction is built without patients' historical medical data. This has the advantage to require minimum patient's information and to work for new patients.

*4.3.2 Imbalanced data.* The dataset of each hospital is imbalanced because the proportion of patients that leave the hospital alive is, fortunately, much larger than in-hospital dead patients. To deal with this well-known problem, a standard solution is to use the Weighted loss function technique or different sampling techniques [19, 31]. In [25, 41] the authors compare empirically the performance of the weighted loss technique and the sampling techniques, however, they were not able to define a clear winner as the results differ for each dataset. In our case, weighted loss[7] outperforms downsampling[8] technique with FL-STANDARD and FL-SIGN.

However, weighting the loss function results in very inaccurate models with Differential Privacy. Indeed, the gradients of the under-represented class (dead patients) are boosted and are therefore larger than the gradients of the other class. The larger the gap between the gradients of the two classes, the more difficult to choose a single clipping threshold $S$ to guarantee Differential Privacy. In particular, if $S$ is calibrated to large gradients (i.e., that of samples from the under-represented class), the added Gaussian noise, whose variance is $S^2\sigma^2$, will also be large yielding poor model accuracy. On the other hand, if $S$ is calibrated to the small gradients (i.e., that of samples from the over-represented class), then samples from the under-represented class will have very small impact on the training which eventually also results in weak model accuracy (the model will be biased towards the majority class).

Instead, as it is described in Section 3.2, we use downsampling [19, 31] which does not require re-weighting the loss and hence overcomes the above artifact caused by clipping (see Alg. 7 for more details). We used downsampling in our experiments with all differentially private learning protocols (FL-STANDARD-DP and FL-SIGN-DP) in Section 4.4[9].

*4.3.3 Model architecture.* As in [5], we use a fully connected neural network model with the following architecture: two hidden layers of 200 units, which use a ReLU activation function followed by an output layer of 1 unit with sigmoid activation function and a binary

[5]https://www.resdac.org/cms-data/variables/claim-inpatient-admission-type-code-ffs

[6]https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[7]https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html

[8]oversampling is not considered because of the privacy constraint.

[9]We report only the results of the best sampling technique for each scheme (see Table 5 for details).

**Table 5: Parameter settings. $T_{\text{cl}}$ is the number of federated runs, $T_{\text{gd}}$ is the number of gradient descent iterations per federated run. $T_{\text{gd}} = Epochs \cdot \frac{|D_k|}{|\mathbb{B}|}$ for client $k$ in federated learning, where $Epochs$ is fixed for all clients.**

| Algorithms | Parameters |
|---|---|
| FL-SIGN-DP | $N = 314$; $C = 3/314$; $\|\mathbb{B}\| = 300$; $DPSGD(\eta = 0.05)$; $S = 2$; $T_{\text{cl}} = 300$; $T_{\text{gd}} = 1$; $\gamma = 0.005$; with downsampling |
| FL-STANDARD-DP | $N = 314$; $C = 3/314$; $\|\mathbb{B}\| = 300$; $DPSGD(\eta = 0.05)$; $S = 2$; $T_{\text{cl}} = 300$; $T_{\text{gd}} = 1$; with downsampling |
| FL-STANDARD | $N = 314$; $C = 3/314$; $\|\mathbb{B}\| = 100$; $SGD(\eta = 0.01)$; $Epochs = 5$; $T_{\text{cl}} = 300$; with weighted loss function |
| FL-SIGN | $N = 314$; $C = 3/314$; $\|\mathbb{B}\| = \|D_k\|$; $SGD(\eta = 0.01)$; $Epochs = 5$; $T_{\text{cl}} = 300$; $T_{\text{gd}} = 5$; $\gamma = 0.001$; with weighted loss function |
| CENRALIZED | $\|D\| = 977927$; $\|\mathbb{B}\| = 100$; $SGD(\eta = 0.01)$; $Epochs = 300$; with weighted loss function |

cross entropy loss function. This results in 4,926,201 parameters in total. The hyperparameters used by each of the considered schemes are summarized in Table 5.

*4.3.4 Computational environment.* Our experiments were performed on a server running Ubuntu 18.04 LTS equipped with a Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz, 192GB RAM, and two NVIDIA Quadro P5000 GPU card of 16 Go each. We use Keras 2.2.0 [13] with a TensorFlow backend 1.12.0 [1] and NumPy 1.14.3 [34] to implement our models and experiments. We use Python 3.6.5 and our code runs on a Docker container to simplify reproducibility.

*4.3.5 Performance Metrics.* We use the following metrics:

- *Balanced accuracy* [10] [7] is computed as $1/2 \cdot (\frac{TP}{P} + \frac{TN}{N}) = \frac{TPR+TNR}{2}$ and is mainly used with imbalanced data. Here, $TPR$ is the *True Positive Rate* and $TNR$ is the *True Negative Rate*; which is calculated as: $TPR = \frac{TP}{P}$ and $TNR = \frac{TN}{N}$, where $P$ and $N$ are the number of positive and negative instances, respectively, and $TP$ and $TN$ are the number of true positive and true negative instances. We note that traditional ("non-balanced") accuracy metrics such as $\frac{TP+TN}{P+N}$ can be misleading for very imbalanced data [3]: in our dataset, the minority class has only 3% of all the training samples (see Table 3), which means that a biased (and totally useless) model always predicting the majority class would have a (non-balanced) accuracy of 97%.
- The *Area under the receiver operating characteristic curve [32]* (*AuROC* ) is also a frequently used accuracy metric [6, 18, 35]. The ROC curve is calculated by varying the prediction threshold from 1 to 0, when *TPR* and *FPR* are calculated at each threshold. The area under this curve is then used to measure the quality of the predictions. A random guess has an *AuROC* value of 0.5, whereas a perfect prediction has the largest *AuROC* value of 1.

*4.3.6 Hyperparameters selection.* For each scheme, $\eta$ was tuned from 0.01 to 0.09 with an increment value of 0.01.

The batch $|\mathbb{B}|$ is selected from [50,100,400,800,$|D_k|$], where client $k$'s data size $|D_k|$ differs for each client; the number of epochs $Epochs$ is selected from [1,5,10,15,20] for each federated scheme, and from [100,150,200,250,300,350,400] for the centralized case. For the federated schemes, we have an additional parameter which is the number of global rounds $T_{\text{cl}}$, which is selected from [100,150,200,250,300,350,400]. The sensitivity $S$ was selected from the reasonable set of [0.5,1,1.5,2,2.5,3]. As in [21], we set $\gamma$ to 0.001 for the non-private scheme FL-SIGN, and we increase it to 0.005 for the private extension FL-SIGN-DP.

The number of hospitals used in the federated schemes are selected from [1,2,3,4,5]. We have to choose one which is large enough to not slow down the convergence and at the same time small enough in order to not deteriorate privacy by increasing the sampling probability, which is one of the principal parameter used in the moments accountant method [2, 30] to compute $\epsilon$.

The values of $\sigma$ can be 1.08, 0.81, 0.63, such that we can reach an $\epsilon$ budget of 1, 2, 4 respectively, after $T_{\text{cl}} = 300$ rounds, which is needed for convergence.

We reported for each scheme in Table 5 the best parameters and also the best technique used to handle the imbalanced data problem.

*4.3.7 Evaluation Method.* We perform $k$-fold cross validation with $k = 5$; first, we split randomly the dataset of each hospital into disjoint training and testing data (80% and 20% respectively). An entire federated run is executed with this split, and all the metrics are evaluated in every round on the union of all clients' testing data. All metric values of the round with the best balanced metric are recorded. The whole run is repeated 4 times each with a new random split of training and testing data, and the minimum and maximum of the recorded performance metrics over all the 5 runs are reported.

## 4.4 Results

The results are summarized in Table 6. A single federated run is composed of 300 rounds, and the best and the worst value of each performance metric over the 5 federated runs are reported. In each round, 3 hospitals are selected randomly for aggregation. Three privacy levels are considered with FL-SIGN-DP and FL-STANDARD-DP: $\varepsilon = 1, 2, 4$ each with $\delta = 1/\max_j |D_j| \leq 1.3 \cdot 10^{-5}$. These values of $\varepsilon$ requires to add Gaussian noise to the gradients with $\sigma = 1.08, 0.81, 0.63$, respectively[10].

We make several observations:

- As mentioned in [14], the performance of FL-STANDARD and CENTRALIZED are close: the balanced accuracy is **0.74** and **0.77**, respectively, which confirms experimentally that Federated Learning is a viable approach for our medical application. These results are consistent with the results reported in [18], which uses the same dataset with the same features to train a logistic regression model in a centralized manner (see results of $D_1$, Table II, in [18]). For example, *AuROC* is **77.2%** − **77.7%** in [18], whereas we get an *AuROC* of

[10] computed numerically based on [2, 30]

| *Privacy* | Algorithms | *Performance* | |
|---|---|---|---|
| | | *AuROC* | *Balanced Accuracy* |
| $\varepsilon = 1$ | FL-SIGN-DP | (0.67,0.68) | (0.63,0.64) |
| | FL-STANDARD-DP | (0.65,0.68) | (0.61,0.63) |
| $\varepsilon = 2$ | FL-SIGN-DP | (0.68,0.71) | (0.64,0.66) |
| | FL-STANDARD-DP | (0.68,0.69) | (0.62,0.64) |
| $\varepsilon = 4$ | FL-SIGN-DP | (0.71,0.72) | (0.65,0.66) |
| | FL-STANDARD-DP | (0.70,0.72) | (0.64,0.66) |
| N/A | FL-SIGN | (0.76,0.77) | (0.68,0.70) |
| | FL-STANDARD | (0.79,0.81) | (0.73,0.74) |
| | CENTRALIZED | (0.82,0.84) | (0.76,0.77) |

**Table 6: Summary of results. The worst and best value of each metric over 5 federated runs are reported.**

**82**% − **84**%, **79**% − **81**% and **76**% − **77**% with CENTRALIZED, FL-STANDARD and FL-SIGN, respectively.

- The performance of FL-SIGN is slightly worse than the performance of FL-STANDARD; the balanced accuracy is **0.70** for FL-SIGN and **0.74** for FL-STANDARD. However, FL-SIGN and FL-SIGN-DP reduce the bandwidth consumption by a factor of 32. Table 7 shows that each client sends only 1.76 Megabytes with FL-SIGN and FL-SIGN-DP, while 56.48 Megabytes are sent with FL-STANDARD and FL-STANDARD-DP. The bandwidth consumption is calculated by measuring the average number of bits sent by a client to the server over the rounds when the client is selected for aggregation. This is computed as $(C \cdot T_{\text{cl}} \cdot \text{model\_size})$ for FL-SIGN and FL-SIGN-DP, and $(32 \cdot C \cdot T_{\text{cl}} \cdot \text{model\_size})$ for FL-STANDARD and FL-STANDARD-DP, where model_size is the number of model parameters (i.e., 4,926,201).
- FL-SIGN-DP performs very similarly to FL-STANDARD-DP, which means that bandwidth efficiency has no real cost when Differential Privacy is also applied. In fact, the performance gap between FL-STANDARD and FL-STANDARD-DP is larger than between FL-SIGN and FL-SIGN-DP especially with stronger privacy guarantee (i.e., smaller $\varepsilon$). This shows that taking the sign of the noisy update and then the median of the noisy signs over all clients on the server (in Line 8 of Alg.4) is more robust against perturbation than taking the simple average of the noisy update vectors (in Line 8 of Alg. 5).
- The results show in general that strong privacy protection can be provided at the cost of a relatively small performance degradation. In fact, the balanced accuracy drops by only 10% when $\varepsilon = 1$ with FL-SIGN-DP. Furthermore, the performance degrades very smoothly as the value of $\varepsilon$ decreases (i.e. as the privacy guarantee gets stronger): the balanced accuracy of FL-SIGN-DP is **0.66** for $\varepsilon = 4$, and only drops to **0.64** when $\varepsilon = 1$.

# 5 RELATED WORK

This section describes the related work to our proposal. We start by presenting the work related to the use of machine learning in medical applications. We then summarize the work related to differentially private federated learning. Finally, we consider the work related to the problem of bandwidth reduction in Federated Learning.

**Table 7: Average bandwidth consumption from a client to the server.**

| Scheme | Bandwidth consumption (Megabytes) |
|---|---|
| FL-SIGN and FL-SIGN-DP | 1.76 |
| FL-STANDARD and FL-STANDARD-DP | 56.48 |

## 5.1 Medical prediction

The paper [5] investigates possibilities offered by the use of Deep Learning and Electronic Health Record (EHR) in order to provide and improve the quality of end-of-life care for hospitalized patients. Having the information about the patients one year before the date of the prediction, the authors define four uneven slices windows. The information collected during the slices windows are used as features to train a model. The model is then used to predict all causes of mortality within a period 3–12 month after the date of the prediction. The authors of [36] also use predictive deep learning models with EHR data (provided by two hospitals) for tasks such as predicting inhospital mortality, 30-day unplanned readmission, prolonged length of stay and all of a patient's final discharge diagnoses. The EHR data of each hospital are used separately, and two personalized models are trained. The EHR data include the data of adult patients who are hospitalized for at least 24 hours.

In [28], a binary logistic regression analysis is performed in order to predict which patients will need Palliative Care Needs (PCNs) based on six risk factors which are: cancer, metastases, age, absence of relatives, liver cirrhosis, and high level of care at admission. During the discharge, the treating physician had to report if the patient had PCNs or not.

The paper [18] develops interpretable models for predicting the risk of complications during hospital stays. The predictive models are based on stacked logistic regressions specifically designed to leverage the evolution of the drugs served during hospital stays. The models can scale with very large volumes of EHR data but they do not consider privacy-related issues.

The paper [14] proposes to use Federated learning with DP, more precisely, it uses objective perturbation [11][12]. An empirical evaluation using two real-world health datasets is performed. However, using objective perturbation implies to deal with convex optimization problems. Hence, logistic regression, perceptron and SVM models are used for the learning tasks. The paper highlights also that the performance of FL without DP are close to the performance of the traditional learning protocol, where the data is shared and centralized in the same place for the training.

### 5.2 Bandwidth Optimization in Federated Learning

Different quantization methods have been proposed to save the bandwidth and reduce the communication costs in federated learning. They can be divided into two main groups: unbiased and biased methods. The unbiased approximation techniques use probabilistic quantization schemes to compress the stochastic gradient and attempt to approximate the true gradient value as much as possible [4][42][40][22]. However, biased approximations of the stochastic gradient can still guarantee convergence both in theory and practice [8, 23, 37]. In signSGD [8], all the clients calculate the stochastic gradient based on a single mini-batch and then send the sign vector of this gradient to the server. The server calculates the aggregated sign vector by taking the median (majority vote) and sends the signs of the aggregated signs back to each client.

The main differences between our scheme (FL-SIGN) and signSGD are as follows:

- FL-SIGN aims to train a common model that is distributed to a random subset of all clients in every round. However, in signSGD, all clients start with the same initialized common model and the server sends the same aggregated model update to *every* client at each round. Selecting only a random subset of clients in each round has at least three benefits. First, FL-SIGN becomes more robust against temporary node failures. Second, FL-SIGN reduces the communication costs upstream to the server. Finally, sampling boosts privacy due to the uncertainty that a specific user's or client's data is used for training or not.
- In FL-SIGN, each client can perform multiple SGD iterations locally using multiple mini-batches before computing the model update. In contrast, signSGD always performs one local SGD iteration with a single mini-batch at every client.
- As all the clients participate at each round in signSGD, the server only transfers the sign of the aggregated signs to the clients in every round. Therefore, only a single bit is transferred per parameter downstream to the clients. In FL-SIGN, the whole model is transferred but only to a random subset of clients.

### 5.3 Differentially Private Federated Learning

Similarly to our FL-SIGN-DP algorithm, another approach [39] also uses DPSGD [2] in order to hide the record of each client's dataset, but the noise is generated in a distributed manner, that is, untrusted server is assumed. Indeed, the noise is added by each client during the training, and then the noisy update is sent to the server. The sum of these noisy updates is sufficiently noised to provide differential privacy. To protect individual updates which are not differentially private, homomorphic encryption is used to guarantee that the adversary can only access the aggregated update which is sufficiently noised. Notice that the noise can be generated distributively, because each client performs only a single mini-batch to compute their model update (i.e., $T_{\text{gd}} = 1$). By contrast, our approach (FL-SIGN-DP) works even if $T_{\text{gd}} > 1$ at the cost of adding larger magnitude of noise and sends only signs for aggregation.

The paper [20] proposed a solution which faithfully follows the SignSGD protocol but is not based on federated learning protocol. The authors use local DP to guarantee client-level-DP. However, it is widely accepted that the large noise needed for local DP decreases accuracy significantly, as the aggregation of the DP updates increases the noise variance. The paper [21] adapts the SignSGD protocol to federated learning for a client-level-DP guarantee. The proposed scheme adds noise in a distributed manner such that the final noise after the aggregation corresponds to the minimum noise needed to ensure DP. However, their proposal, that uses a discrete Gaussian mechanism and needs several bits per parameter, is less bandwidth efficient than [20] that only sends one bit per parameter.

Our solution is based on [21] but considers record-level guarantee instead of client-level guarantee. It therefore requires less perturbations and reduces bandwidth by sending only one bit per parameter.

Differential Private Federated Machine Learning has been studied in the context of medical applications to provide client-level privacy guarantee [35] or record-level privacy guarantee [6, 15]. Our solution improves the state of the art as it provides record-level privacy guarantee and optimizes bandwidth efficiency by sending only one bit per parameter. Furthermore, as opposed to most published papers that use public, often synthetic, datasets with limited size, we evaluated our scheme using a large cohort of real-world data.

## 6 ETHICAL CONSIDERATIONS

Our study was approved by our Institutional Review Board (IRB) process before any research activity began. The EHR dataset is stored on a server whose security was audited by Inria security teams.

## 7 CONCLUSIONS

Real-world data are generally highly imbalanced, our solution aims to handle this well-known problem while it provides both bandwidth efficiency and differentially private guarantee. We experimentally evaluate the performance of our solution for in-hospital mortality prediction using the Premier Healthcare database, containing about one million records of patients. We consider a scenario where 314 hospitals are collaborating to train, using our Federated Learning scheme, a prediction model without exchanging any of their patients' data. Our scheme guarantees that no internal or external adversary that has access to the final model, intermediate updates or even all the messages that are exchanged during the training phase can infer any information about any of the patient data that were used by each hospital.

The accuracy performance results are very encouraging. They show in general that strong privacy protection can be provided at

the cost of a relatively small performance degradation. Furthermore, our scheme reduces the bandwidth consumption by a factor of 32 compared to standard federated learning schemes, reducing it from 56.48 to 1.76 Megabytes.

We believe that this paper reports the first large-scale experimental assessment in favor of using privacy-preserving federated learning for the purpose of in-hospital mortality prediction. We demonstrate that it is possible to benefit from the power of machine learning without sacrificing the privacy of patients. Hospitals, and other medical institutions, are reluctant to collaborate because they often consider their patient medical records as their own intellectual properties. Our scheme protects these intellectual properties of participating entities on record-level.

## 8 ACKNOWLEDGMENTS

This article was developed in the framework of the Grenoble Alpes Data Institute, supported by the French National Research Agency under the "Investissements d'avenir" program (ANR-15-IDEX-02). The research was supported by the NRDI fund of the Ministry of Innovation and Technology NRDI Office, and also within the framework of the Artificial Intelligence National Laboratory Program. This project has received support from the EU/EFPIA Innovative Medicines Initiative 2 Joint Undertaking (MELLODDY grant $n^\circ$ 831472). This project has received support from the ANR project ANR-16-CE25-0010.

## REFERENCES

[1] Martín Abadi, , et al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/ Software available from tensorflow.org.
[2] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *ACM CCS*.
[3] Josephine Akosa. 2017. Predictive accuracy: a misleading performance measure for highly imbalanced data. In *Proceedings of the SAS Global Forum*. 2–5.
[4] Dan Alistarh, Jerry Li, Ryota Tomioka, and Milan Vojnovic. 2016. QSGD: Randomized Quantization for Communication-Optimal Stochastic Gradient Descent. *CoRR* abs/1610.02132 (2016). arXiv:1610.02132 http://arxiv.org/abs/1610.02132
[5] Anand Avati, Kenneth Jung, Stephanie Harman, Lance Downing, Andrew Ng, and Nigam H. Shah. 2018. Improving palliative care with deep learning. *BMC Medical Informatics and Decision Making* 18, 4 (12 Dec 2018), 122. https://doi.org/10.1186/s12911-018-0677-8
[6] Brett K. Beaulieu-Jones, William Yuan, Samuel G. Finlayson, and Zhiwei Steven Wu. 2018. Privacy-Preserving Distributed Deep Learning for Clinical Data. arXiv:cs.LG/1812.01484
[7] Mohamed Bekkar, Hassiba Djema, and T.A. Alitouche. 2013. Evaluation measures for models assessment over imbalanced data sets. *Journal of Information Engineering and Applications* 3 (01 2013), 27–38.
[8] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Anima Anandkumar. 2018. signSGD: compressed optimisation for non-convex problems. *CoRR* abs/1802.04434 (2018). arXiv:1802.04434 http://arxiv.org/abs/1802.04434
[9] Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. 2018. signSGD with Majority Vote is Communication Efficient And Byzantine Fault Tolerant. *CoRR* abs/1810.05291 (2018). arXiv:1810.05291 http://arxiv.org/abs/1810.05291
[10] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. 2010. The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition*. IEEE, 3121–3124.
[11] Kamalika Chaudhuri and Claire Monteleoni. 2009. Privacy-preserving logistic regression. In *Advances in neural information processing systems*. 289–296.
[12] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research* 12, Mar (2011), 1069–1109.
[13] François Chollet et al. 2015. Keras. https://keras.io.
[14] Olivia Choudhury, Aris Gkoulalas-Divanis, Theodoros Salonidis, Issa Sylla, Yoonyoung Park, Grace Hsu, and Amar Das. 2019. Differential Privacy-enabled Federated Learning for Sensitive Health Data. arXiv:cs.LG/1910.02578
[15] Olivia Choudhury, Aris Gkoulalas-Divanis, Theodoros Salonidis, Issa Sylla, Yoonyoung Park, Grace Hsu, and Amar Das. 2020. Differential Privacy-enabled Federated Learning for Sensitive Health Data. arXiv:cs.LG/1910.02578
[16] Marta TERRON CUADRADO. 2019. ICD-9-CM: International Classification of Diseases, Ninth Revision, Clinical Modification. https://ec.europa.eu/cefdigital/wiki/display/EHSEMANTIC/ICD-9-CM%3A+International+Classification+of+Diseases%2C+Ninth+Revision%2C+Clinical+Modification.
[17] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3–4 (2014).
[18] A. Fejza, P. Genevès, N. Layaïda, and J. Bosson. 2018. Scalable and Interpretable Predictive Models for Electronic Health Records. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. 341–350. https://doi.org/10.1109/DSAA.2018.00045
[19] Haibo He and Edwardo A Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering* 21, 9 (2009), 1263–1284.
[20] Richeng Jin, Yufan Huang, Xiaofan He, Tianfu Wu, and Huaiyu Dai. 2020. Stochastic-Sign SGD for Federated Learning with Theoretical Guarantees. arXiv:cs.LG/2002.10940
[21] Raouf Kerkouche, Gergely Ács, and Claude Castelluccia. 2020. Federated Learning in Adversarial Settings. arXiv:cs.CR/2010.07808
[22] Jakub Konecný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. *CoRR* abs/1610.05492 (2016). arXiv:1610.05492 http://arxiv.org/abs/1610.05492
[23] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. 2018. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. In *International Conference on Learning Representations, ICLR 2018*. https://openreview.net/forum?id=SkhQHMW0W
[24] Rupa Makadia and Patrick B. Ryan. 2014. Transforming the Premier Perspective® Hospital Database into the Observational Medical Outcomes Partnership (OMOP) Common Data Model. In *EGEMS*.
[25] Kate McCarthy, Bibi Zabar, and Gary Weiss. 2005. Does cost-sensitive learning beat sampling for classifying rare classes?. In *Proceedings of the 1st international workshop on Utility-based data mining*. 69–77.
[26] Margaret Mcdonald, Timothy Peng, Sridevi Sridharan, Janice Foust, Polina Kogan, Liliana Pezzin, and Penny Feldman. 2012. Automating the medication regimen complexity index. *Journal of the American Medical Informatics Association : JAMIA* 20 (12 2012). https://doi.org/10.1136/amiajnl-2012-001272
[27] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2016. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*.
[28] Cornelia Meffert, Gerta Rücker, Isaak Hatami, and Gerhild Becker. 2016. Identification of hospital patients in need of palliative care – a predictive score. *BMC Palliative Care* 15 (12 2016). https://doi.org/10.1186/s12904-016-0094-7
[29] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2018. Inference Attacks Against Collaborative Learning. *CoRR* abs/1805.04049 (2018). arXiv:1805.04049 http://arxiv.org/abs/1805.04049
[30] Ilya Mironov, Kunal Talwar, and Li Zhang. 2019. Rényi Differential Privacy of the Sampled Gaussian Mechanism. *CoRR* abs/1908.10530 (2019). arXiv:1908.10530 http://arxiv.org/abs/1908.10530
[31] Ajinkya More. 2016. Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint arXiv:1608.06048* (2016).
[32] Sarang Narkhede. 2018. Understanding AUC - ROC Curve. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.
[33] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *IEEE Symposium on Security and Privacy, 2019*. 739–753. https://doi.org/10.1109/SP.2019.00065
[34] Travis E Oliphant. 2006. *A guide to NumPy*. Vol. 1. Trelgol Publishing USA.
[35] Stephen R. Pfohl, Andrew M. Dai, and Katherine Heller. 2019. Federated and Differentially Private Learning for Electronic Health Records. arXiv:cs.LG/1911.05861
[36] Alvin Rajkomar and al. 2018. Scalable and accurate deep learning with electronic health records. *npj Digital Medicine* 1, 1 (2018), 18. https://doi.org/10.1038/s41746-018-0029-1 url, An earlier version appeared in eprint arXiv:1801.07860.
[37] Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 2014. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In *INTERSPEECH 2014*. 1058–1062. http://www.isca-speech.org/archive/interspeech_2014/i14_1058.html
[38] Reza Shokri and Vitaly Shmatikov. 2015. Privacy-Preserving Deep Learning. In *ACM SIGSAC Conference on Computer and Communications Security, 2015*. 1310–1321. https://doi.org/10.1145/2810103.2813687
[39] Stacey Truex and al. 2018. A Hybrid Approach to Privacy-Preserving Federated Learning. *CoRR* abs/1812.03224 (2018). arXiv:1812.03224 http://arxiv.org/abs/1812.03224
[40] Hongyi Wang, Scott Sievert, Shengchao Liu, Zachary B. Charles, Dimitris S. Papailiopoulos, and Stephen Wright. 2018. ATOMO: Communication-efficient Learning via Atomic Sparsification. In *NeurIPS*.

[41] Gary M Weiss, Kate McCarthy, and Bibi Zabar. 2007. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? *Dmin* 7, 35-41 (2007), 24.
[42] Wei Wen and al. 2017. TernGrad: Ternary Gradients to Reduce Communication in Distributed Deep Learning. *CoRR* abs/1705.07878 (2017). arXiv:1705.07878 http://arxiv.org/abs/1705.07878
[43] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2020. iDLG: Improved Deep Leakage from Gradients. *arXiv preprint arXiv:2001.02610* (2020).
[44] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep Leakage from Gradients. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 14747–14756. http://papers.nips.cc/paper/9617-deep-leakage-from-gradients

# A APPENDIX

## A.1 Convergence Proofs

The convergence proof of FL-SIGN can be found in [9], whereas the proof of FL-SIGN-DP is a simple adaptation of Theorem 2 from [9]. Here we outline only the main deviations from the proof of that theorem.

**Assumptions:**

(1) *Lower bound:* For all $x$ and some constant $f^*$, $f(x) \geq f^*$, where $f$ denotes the loss/objective function.
(2) *Smoothness:* Let $g(x)$ denote the gradient of the objective function $f$ evaluated at $x$. Then, for all $x, y$ and some non-negative constant $\mathbf{L} = (L_1, L_2, \ldots, L_n)$,
$$|f(y) - [f(x) + g(x)^\mathsf{T}(y - x)]| \leq 1/2 \sum_i L_i (y_i - x_i)^2$$
(3) *Variance bound:* Upon receiving query $x \in \mathbb{R}^n$, the stochastic gradient oracle gives us an independent, unbiased estimate $\hat{g}$ that has bounded variance per coordinate: $\mathbb{E}[\hat{g}(x)] = g(x)$, $\mathbb{E}[(\hat{g}(x)_i - g(x)_i)^2] \leq \tau_i^2$ for a vector of non-negative constants $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_n)$.
(4) *Unimodal, symmetric gradient noise:* At any given point $x$, each component of the stochastic gradient vector $\hat{g}(x)$ has unimodal distribution that is also symmetric about the mean.

Note that adding extra Gaussian noise to each gradient component for the purpose of differential privacy will not violate Assumption 4.

Theorem A.1. *If* $|\mathbb{B}| = T_{\text{cl}}$, $T_{\text{gd}} = 1$, *and* $\gamma = \sqrt{\frac{f_0 - f_*}{||\mathbf{L}||_1 T_{\text{cl}}}}$, *then*
$$\frac{1}{T_{\text{cl}}} \sum_{t=0}^{T_{\text{cl}}-1} \mathbb{E}||g_t||_1 \leq \frac{2}{\sqrt{T_{\text{cl}}}} \left( \frac{||\boldsymbol{\tau}||_1 + nS\sigma}{\sqrt{CN}} + \sqrt{||\mathbf{L}||_1 (f_0 - f^*)} \right)$$

Proof. The primary focus of the proof is to bound the probability that a client computes the sign of a parameter update correctly. Let $M = CN$. As in [9], let $Z_i \in [0, M]$ denote the number of correct sign bits received by the aggregator for parameter $i$, and $p$ denotes the probability that a honest client computes the correct bit. Let $\omega = p - \frac{1}{2}$. According to Theorem 2 in [9],
$$\mathrm{P}\left[Z_i \leq M/2\right] \leq \frac{\sqrt{\mathbb{E}[(\tilde{g}_i - g_i)^2]}}{\sqrt{M}|g_i|}$$
where $\tilde{g}$ is the noisy stochastic gradient. Observe that $\tilde{g}$ has two sources of randomness; (1) the stochasticity of the sampling mechanism which is modelled by the stochastic gradient oracle (see Assumption 3), and (2) the Gaussian noise that is introduced in order to guarantee DP. Importantly, these are independent sources of randomness. Therefore, the probability that a vote fails for the $i^{th}$ parameter is bounded as
$$\begin{aligned}
\mathrm{P}\left[Z_i \leq M/2\right] &\leq \frac{\sqrt{\mathbb{E}[(\tilde{g}_i - g_i)^2]}}{\sqrt{M}|g_i|} \\
&\leq \frac{\sqrt{\tau_i^2 + S^2\sigma^2}}{\sqrt{M}|g_i|} \qquad \text{(by independence)} \\
&\leq \frac{\tau_i + S\sigma}{\sqrt{M}|g_i|}
\end{aligned}$$
where the second inequality follows from Assumption 3 and the fact that the variance of the Gaussian noise is $S^2\sigma$. The rest of the derivation is identical to the proof of Theorem 2 in [9].

□