International Journal of Data Science and Analytics
https://doi.org/10.1007/s41060-024-00625-7

REGULAR PAPER

Leveraging local data sampling strategies to improve federated learning

Christoph Düsing[1] · Philipp Cimiano[1] · Benjamin Paaßen[1]

Received: 30 April 2024 / Accepted: 6 August 2024

Abstract
Federated learning (FL) facilitates shared training of machine learning models while maintaining data privacy. Unfortunately, it suffers from data imbalance among participating clients, causing the performance of the shared model to drop. To diminish the negative effects of unfavourable data-specific properties, both algorithm- and data-based approaches seek to make FL more resilient against them. In this regard, data-based approaches prove to be more versatile and require less domain knowledge to be applied efficiently. Hence, they seem particularly suitable for widespread application in various FL environments. Although data-based approaches such as local data sampling have been applied to FL in the past, previous research did not provide a systematic analysis of the potential and limitations of individual data sampling strategies to improve FL. To this end, we (1) identify relevant local data sampling strategies applicable to FL systems, (2) identify data-specific properties that negatively affect FL system performance, and (3) provide a benchmark of local data sampling strategies regarding their effect on model performance, convergence, and training time in synthetic, real-world, and large-scale FL environments. Moreover, we propose and rigorously test a novel method for data sampling in FL that locally optimizes the choice of sampling strategy prior to FL participation. Our results show that FL can greatly benefit from applying local data sampling in terms of performance and convergence rate, especially when data imbalance is high or the number of clients and samples is low. Furthermore, our proposed sampling strategy offers the best trade-off between model performance and training time.

Keywords Federated learning · Data sampling · Data imbalance

1 Introduction

Ever since its introduction by Google [1], federated learning (FL) has become an increasingly popular approach towards privacy-preserving distributed machine learning (ML) [2, 3]. As such, it enables multiple participants (usually referred to as *clients*) to jointly train a shared model without granting third parties access to their respective private data [3–5]. While FL facilitates privacy-preserving learning, it comes at the cost of data quality assurance, as no party can directly access and thus evaluate the data held by individual clients [6]. Among others, a central dimension of data quality in FL is data imbalance. It measures how similar a client's feature and label distributions are to the overall cohort's distributions. Unfortunately, FL is prone to suffer from data imbalance, causing the performance of the shared model to decrease significantly when imbalance is high [7–9].

Previous work addresses the issue of data imbalance either on an algorithm- or data-level [10, 11]. The former refers to approaches tailored towards dealing with data imbalance by adjusting model parameters or architectures [10] (e.g. [12, 13]). The latter seeks to mitigate data imbalance by manipulating the data held by individual clients [10, 11]. This is usually achieved by deploying data sampling strategies on each client's local data prior to participating in training [11, 14–16]. Both approaches have demonstrated their suitability to compensate for the drop in performance due to data imbalance to some extent [10, 14]. However, algorithm-based approaches require more domain knowledge and larger amounts of data to be employed efficiently [10, 17]. In turn, it seems reasonable to consider local data sampling when

✉ Christoph Düsing
cduesing@techfak.uni-bielefeld.de

Philipp Cimiano
cimiano@techfak.uni-bielefeld.de

Benjamin Paaßen
bpaassen@techfak.uni-bielefeld.de

[1] CITEC, Bielefeld University, Inspiration 1, 33619 Bielefeld, Germany

Published online: 29 August 2024

facing data imbalance in FL [18]. In spite of data sampling having been applied previously in FL to account for data imbalance, [14–16, 18–25], to the best of our knowledge, no prior research investigated the potential and limitations of local data sampling strategies in FL systematically. This lack of comprehensive evidence on data-based approaches forces FL operators to rely on trial-and-error to determine if data sampling is beneficial for them and what data sampling strategy to choose. This holds the potential to cause significant computational overhead during training, ultimately increasing the cost of FL application.

This motivates our work towards providing an in-depth analysis of the potential of local data sampling strategies to improve FL models. Accordingly, this work tries to answer the following research questions:

RQ1: To what extent can local data sampling improve FL performance, convergence, and training time in face of unfavourable data-specific properties?

RQ2: What are the unique advantages and disadvantages of different sampling strategies over another?

RQ3: Can locally optimizing the choice of data sampling strategy improve FL further?

To answer these research questions, we identify relevant data sampling strategies as well as data-specific properties known to influence the performance of FL. Afterwards, we control the severity of these data properties in simulated FL settings, apply the previously identified sampling strategies to them, and measure the change in performance, convergence rate, and training time. From the findings among four different, widely acknowledged datasets for FL research, we finally draw conclusions regarding the efficient utilization of local data sampling strategies in FL.

In summary, the contributions of our work are as follows:

1. We systematically benchmark existing local data sampling strategies for FL with respect to model performance, rate of convergence, and training time in face of three different notions of imbalanced or unfavourable local data;
2. Beyond that we propose a novel sampling strategy named *Optimized* that locally optimizes the choice of data sampling approach prior to FL participation and highlight its advantages over existing strategies both theoretically and empirically;
3. Afterwards, we investigate how different data-specific properties affect the choice of the optimal sampling strategy at each client's side and provide guidance for future decision-makers of FL applications;
4. Finally, our work sheds light on the distinct advantages of different local data sampling strategies and how they can be employed efficiently when facing data imbalance, particularly small cohorts, or clients with limited data in FL.

In turn, our work is of great value for researchers and practitioners concerned with the application and improvement of FL. Furthermore, it creates promising avenues for future research linking FL literature to the recently emerging *data-centric artificial intelligence* (AI) domain [26, 27].

For this purpose, we elaborate on preliminaries and related work in Sects. 2 and 3. Next, we outline the proposed methodology before conducting a theoretical analysis on the improvements through our proposed local sampling strategy in Sect. 5. This is followed by presenting the results of our empirical evaluation in Sect. 6. Then, we confirm our previous findings using real-world and large-scale FL data as well as state-of-the-art (SOTA) ML models in Sect. 7. Finally, we conclude the article and discuss limitations as well as future research directions in Sect. 8.

2 Preliminaries

FL is a novel approach towards distributed ML. Unlike traditional central ML that requires data from all sources to be collected and stored at a single side, FL allows data to remain at the respective owner's side [28].

By design, FL therefore maintains data privacy of all involved clients and is in turn a perfect fit for a wide range of domains with particularly sensitive data [3, 29, 30], e.g. healthcare [31, 32] or mobile and edge devices [33]. The most common approach towards FL, i.e. *FedAvg* [4], consist of three main parts, namely a *ML model*, a *central server*, and participating *clients* [28].

- A *ML model* "is a set of algorithms and their parameters that are arranged in a particular structure to calculate predictions based on the collected data" [28, p.3]. Here, it is essential that the model architecture is decided upon prior to starting the federated training. Typically, ML models in FL are weight-based neural networks of any kind, including convolutional neural networks (CNNs) and many others [28].
- The *central server* is responsible for orchestrating and coordinating the federated training as well as the participating clients [3, 4]. To do so, it sends out the shared model to those clients selected for the upcoming round of training and receives their updates afterwards [28]. Finally, the central server aggregates all clients' updates to receive a new global model [4].
- *Clients* are all entities that hold a set of private data and that are willing to participate in the FL scheme. Depending on the setting, this could either be a personal device such as a smartphone [28] or a dedicated server held by larger organizations, e.g. healthcare providers [31, 32].

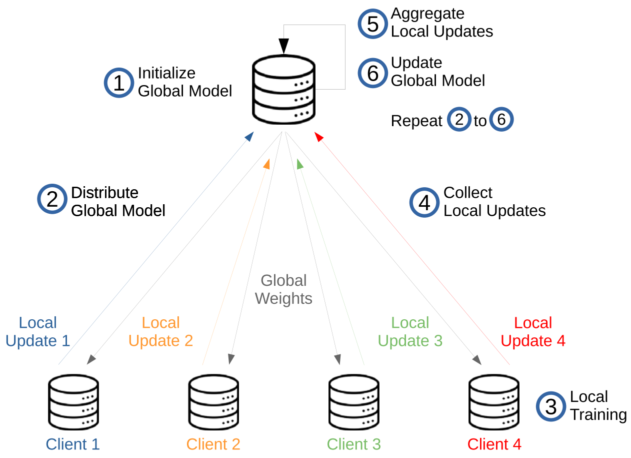


Fig. 1 Steps of FL

The FL training procedure itself is depicted in Fig. 1 and starts with initializing the global model. To train the ML model among all participating clients afterwards, the central server issues several rounds of federated training [4, 34]. During each round of training, a global model held by the central server is sent to the clients (Fig. 1, step 2) which fit it to their respective local data (step 3) [3, 4]. Subsequently, all local model updates are sent back to the server in a privacy-preserving manner (step 4) [29], usually by utilizing secure aggregation [35–37]. Lastly, the central server aggregates all received model updates (step 5) to compute a new global model (step 6) [4]. Once finished, a global model is received that incorporates knowledge from all data contained in the cohort [9].

3 Related work

As data imbalance is a major cause of poor FL performance [5, 8], numerous scholars worked towards making FL more resilient against it [21, 38]. Here, data imbalance refers to client-specific characteristics of data that differ from the cohort in terms of labels, features, or quantity held [8]. Their proposed solutions usually fall into one of two categories: algorithm- and data-based approaches [10, 11]. Following this classification, algorithm-based approaches are characterized by improving the training procedure or tuning the model parameters [39, 40]. Data-based approaches, on the other hand, address data imbalance by balancing the data held by individual clients [19] or the entire cohort [41].

3.1 Algorithm-based approaches

A well-established field of FL research is concerned with the improvements of learning algorithms, e.g. by implementing cost-sensitive learners [42–44], model pruning [45], or distribution regularization [46]. These approaches aim to alleviate performance deterioration caused by data imbalance through algorithmic improvements and large-scale hyperparameter tuning [11, 17].

In terms of cost-sensitive learning, various cost functions have been proposed. In their work, Zhou et al. [42] propose to re-weight model updates with respect to clients' class distributions, whereas other works rely on penalty-terms during loss computation [43]. In this vein, Zhan et al. integrate a calibrated cross-entropy loss into local updates by measuring clients' pairwise label margins [47]. Instead of modifying the loss function used during training, model pruning [45] and parameter activation [48] can dynamically adjust model depth and parameters during training with respect to individual clients and entire cohorts. This allows clients with limited resources (in terms of data and computational resources) to participate in the training procedure [45], but requires the central server to define the scope of these adjustments a priori. Alternatively, distribution-regularized FL aims to project different local data distributions of clients to a common space with minimal distance among the projected distributions using kernel functions [46]. While this approach outperforms non-regularized FL in various data settings, it adds to the complexity of the FL training procedure.

Finally, recent studies address the issue of partially class-disjoint data, which causes their respective model updates to be biased [49]. To this end, Li et al. [50] suggest to extend existing FL strategies with a restricted softmax function to limit the local updates of weights associated with missing classes. The FL strategy FedGELA achieves globally unbiased models from locally biased models by employing a simplex equiangular tight frame [51], which corrects the classification angle of each class on a global level [49].

3.2 Data-based approaches

According to Li et al. [52], learning algorithms are only one of many aspects when applying FL systems. Careful consideration of the data and proper data life cycles are essential parts of successful FL application [52], too. Unlike their algorithm-based counterparts, data-based approaches modify the data held by clients to diminish the effects of data imbalance on FL and require neither costly hyperparameter tuning nor deep domain knowledge [10, 17]. To do so, data-based approaches are either applied locally, i.e. at each client's side, or globally, i.e. among the cohort's combined data.

Here, the former aims to improve the learning procedure by balancing the datasets held by each client [19]. In this vein, Duan et al. [19] were the first to propose a framework to avoid accuracy degradation of the federated model by re-balancing the train data of each client locally. Since then, several works have improved data-based approaches further, usually by oversampling minority classes of clients (e.g. [10, 16]), undersampling their majority classes (e.g. [16]), or performing a combination of both (e.g. [14, 20]) in order to re-calibrate their data distributions [15, 21].

On the other hand, global approaches are initiated by the central server and aim to balance the combined data of all clients. Therefore, early approaches suggest sharing a global dataset consisting of small batches of data provided by each client that can be combined with the local datasets during training [41, 53]. Given the impracticability of sharing a global dataset in FL [54], more recent studies apply global data augmentation in a privacy-preserving manner [54, 55]. Therefore, they either rely on synthesizing reliable samples from each client [55] or avoid local training bias using globally shared pseudo-data [54].

While data-based approaches improve the predictive performance of FL models [10, 16], little effort has been conducted to identify prerequisites of their successful application. In this vein, Jorge et al. [14] demonstrated that the number of clients involved in the training procedure affects the margin of performance improvements. Moreover, no significant differences were measured between oversampling or undersampling client data when applied to healthcare data [16]. However, no consensus has yet been reached regarding the potential of local data sampling strategies to improve FL or the advantages of different sampling strategies over alternatives.

4 Methodology

In this work, we opt to consider data-based approaches for FL improvement over algorithm-based alternatives, because—as stated earlier—they require less domain knowledge, technical expertise, and data to be available in order to be applied properly, while achieving similar or even superior performance [10]. In this regard, our work also follows the recently emerging trend towards data-centric AI [26, 27]. In short, this line of research is concerned with improving AI through improvements made to the data it is trained upon rather than the models used during training. Ultimately, local data sampling in FL aligns well with the goals of data-centric AI, as both are supposed to reduce the complexity of AI or FL development and deployment. Furthermore, we stick to benchmarking local data sampling approaches for FL instead of global data sampling, considering their ease of application [10], guarantees regarding data privacy [54], and widespread application (e.g. [10, 14, 16]).

Our approach towards analysing local data sampling strategies for FL relies on identifying relevant data sampling strategies as well as data-specific properties affecting the performance of FL negatively. Once we recognized sampling strategies and data properties, we iteratively increase the severity of these properties in simulated FL environments. Afterwards, we apply data sampling to these FL environments to mitigate the negative impact imposed by unfavourable data configurations. During the process, we measure the improvement in terms of performance achieved due to each sampling strategy.

4.1 Existing local data sampling strategies

As we apply data sampling locally at each client's side, traditional data sampling strategies, i.e. those not specifically designed for the FL setting, can be considered. Here, a vast variety of local sampling strategies is available that either fall into the category of undersampling, oversampling or hybrid sampling [56]. In order to apply them in a federated manner, each client has to run one of these sampling strategies locally prior to FL participation. Finally, balancing each client's local data causes the combined cohort's data to be balanced, too.

In the following, we outline and explain all local data sampling strategies benchmarked during later analyses. Moreover, we provide a visualization of these sampling strategies when applied to FL in Fig. 2. Besides, it is worth mentioning that we do not apply generative approaches for local data oversampling such as presented by Lee et al. [57], as local clients can be limited in terms of data available and might therefore be unable to train the required generative models in the first place [32].

- **None** It serves as baseline for the data sampling strategies. In this setting, no local data sampling is applied and the FL model is trained without re-balancing clients' data.
- **Constrained** In this setting, no data sampling is applied either. Accordingly, it serves as another baseline for the subsequent sampling strategies. However, it differs from *None*, as clients that lack data on one or more of the classes are excluded from the training entirely. We do so, as applying oversampling or undersampling locally requires the respective client to hold at least a single sample per class that can then be oversampled. We include this sampling strategy, as it serves as ablation study for the following strategies to ensure that potential improvements are not just due to the exclusion of aforementioned clients but can be attributed to applying the respective data sampling.
- **Undersampling** We employ random undersampling of the majority classes at each client to match the size of their respective minority classes. Undersampling is known to improve traditional ML [58] but is also beneficial for FL [16].
- **Oversampling** It focuses on adding minority class samples to match the size of the majority class [56, 59]. In our study, we choose SMOTE [60] over alternatives such

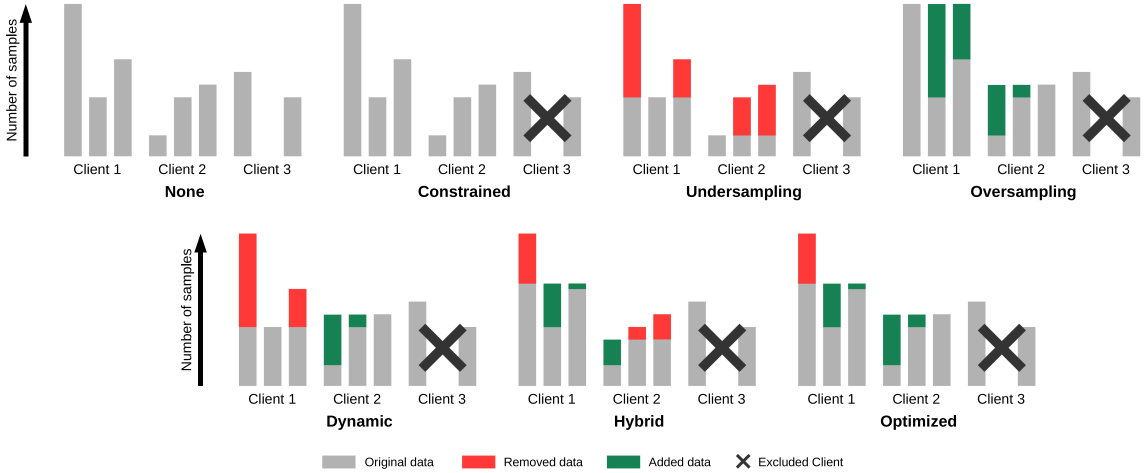


Fig. 2 Local data sampling strategies for a multi-class classification task

as random oversampling due to its demonstrated superiority in terms of achievable model performance [60]. Here, we follow previous works that deploy oversampling using SMOTE in FL [16]. Accordingly, each client applies oversampling using SMOTE locally prior to joining the federated training procedure.

- **Dynamic** As oversampling is known to outperform undersampling in face of particularly small local datasets [56], we apply oversampling to clients with datasets smaller than the average client. In case of sufficiently large datasets, undersampling avoids overfitting better [58] and is consequently applied to clients with above-average data counts.
- **Hybrid** Hybrid data sampling combines undersampling the majority classes and oversampling minority classes in order to balance the dataset [14]. From the variety of available techniques (e.g. [61, 62]), we implement SMOTETomek [61] due to its proven suitability for FL environments [14].

4.2 Optimized local data sampling

Besides benchmarking existing local data sampling strategies applicable to FL, a main objective of this work is to introduce a novel data sampling strategy to further improve federated model performance in unfavourable data settings. Therefore, we propose the following strategy named *Optimized* that locally optimizes the choice of data sampling approach prior to FL participation. In the following, we elaborate upon the rationale behind our proposed local sampling strategy and provide details on its application to FL cohorts.

- **Optimized** Düsing and Cimiano [9] observed that clients not benefiting from FL participation in terms of performance are likely to contribute more to the success of the cohort. The rationale here is that clients with small benefit are those capable of training decently performing local models on their own, as benefit is defined as improvements in terms of performance achieved using the federated model over a locally trained model [9, 63, 64]. Hence, their model updates contributed to the cohort are of particularly high value. This inspires us to consider a novel setting of local data sampling, i.e. a locally optimized sampling strategy.
  In order to do so, each client within the cohort is tasked to run the local sampling strategy selection prior to participating in the training. Therefore, they locally train a model of the same architecture as the one used later on during federated training for each of the three sampling strategies oversampling, undersampling and hybrid sampling.[1] Afterwards, each client chooses the sampling strategy yielding the best performance in terms of F1-score on their local data partition for the subsequent FL participation. Thereby, we optimize each client's local model performance, which—according to Düsing and Cimiano [9]—will improve the federated model's performance the most.

Per default, we apply SMOTE, SMOTETomek, and random undersampling such that, afterwards, each client holds the same amount of samples for each class present in the data. Thus, the ratio r of minority class size n_{Minority} to the majority class size n_{Minority} defined as $r = n_{\text{Minority}}/n_{\text{Majority}}$ is $r = 1$ throughout all analyses.

[1] Note that dynamic sampling does not need to be considered here, as it ultimately chooses between applying oversampling or undersampling at the respective client.

4.3 Data-specific properties affecting FL performance

The literature on FL identifies various data-specific properties affecting the performance of FL. Previously, we mentioned the detrimental effect of data imbalance on the predictive performance of the shared model [10, 13], which is arguably the most prominent property to consider. However, both cohort size and data counts affect performance, too [8, 14]. In order to analyse the potential of local data sampling, we consider the following three factors influencing FL performance:

- **Data imbalance** Data imbalance in FL refers to deviant client data dissimilar to the remaining data held by the cohort [8]. The most common type of data imbalance is label imbalance, where clients differ in terms of label distribution and quantity [3, 8, 13, 50]. This was repeatedly shown to have detrimental effects on performance, with the degree of imbalance influencing the decrease of accuracy [10, 11, 13, 17, 21, 47, 50, 65]. To account for data imbalance, Cheng et al. [10] identified data-based approaches as a promising solution to reduce its impact on FL in various settings.
- **Cohort size** The number of clients participating in the FL affects both its communication overhead [66] as well as its performance [14, 67]. In this vein, Jorge et al. [14] demonstrated that reducing the number of participating clients decreases the accuracy of a FL model. More importantly, however, they also identified that data sampling using SMOTETomek can reduce this negative impact to some extent [14]. Based on these results, we hypothesize that the number of clients participating in the cohort needs to be controlled for during our analyses.
- **Samples per client** ML, and deep learning in particular, benefits significantly from the availability of big data [68]. FL is no exception in this regard. Here, previous studies show that clients with small datasets contribute less to the success of a cohort than those with large datasets [9]. Similarly, Li and colleagues [8] identify large imbalance in data quantity among clients to negatively influence performance. Hence, reducing the amount of data available per client risks harming the shared model's performance. Accordingly, we consider the number of samples per client a relevant data property in terms of our subsequent analyses.

5 Theoretical analysis

Inspired by similar studies [47, 49, 69, 70], we conduct a theoretical analysis of the improved convergence through our proposed *optimized* sampling strategy. In simple terms, our argument is that a local convergence of the loss at every client implies a global convergence of the loss, provided that the loss is smooth and the variance of the weights decreases over time.

The global objective function in standard FL such as *FedAvg* is defined as:

$$F(w) = \sum_{j=1}^{N} \frac{n_j}{N} F_j(w), \tag{1}$$

where N is the overall number of samples, n_j the number of samples on client j, and $F_j(w)$ the local objective function of client j.

We assume that the weights w are learned step by step via some iterative optimizer. More specifically, we assume that each client j obtains local weights w_j^t via some learning rule, and these are aggregated to global weights w^t via Eq. 2 [4].

$$w^{(t)} = \sum_{j=1}^{N} \frac{n_j}{N} w_j^{(t)}. \tag{2}$$

For our convergence analysis, we now make three key assumptions. First, we assume that the local objective functions $F_j(w)$ are L-smooth, meaning that

$$\left\| F_j(u) - F_j(v) \right\| \leq L \left\| u - v \right\|, \quad \forall u, v, \in \mathbb{R}^d \tag{3}$$

This assumption is fulfilled for typical differentiable models and is common in FL theory (e.g. [71, 72]).

Second, we assume that the variance of the weights decreases over time, meaning that

$$\left\| w^{(t)} - w_j^{(t)} \right\|^2 \leq D_t, \quad \forall t, j \tag{4}$$

where $D_1, D_2, \ldots$ is an eventually decreasing sequence of bounding constants on the difference between the weights of any single client and the aggregated weights via (2). This assumption is motivated by the fact that the weights of all clients are initialized at the same point (hence $D_1 = 0$) and the local loss of every client approximates the global loss. Hence, any reasonable iterative optimizer will drive the weights to similar local optima, thus decreasing their distance over time. Also, refer to Fig. 3 for an illustration.

To make this notion of local optima precise, we finally assume, in line with similar studies [71], that the expected difference between the local loss of client j and the optimal loss achievable for client j is bounded as follows.

$$\mathbb{E}\left[F_j(w_j^{(t)}) - F_j(w^*) \right] \leq C_t \tag{5}$$

where w^* are the loss-minimizing weights for client j, and where $C_1, C_2, \ldots$ is an eventually decreasing sequence of

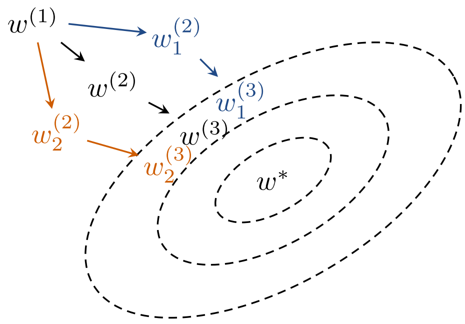


Fig. 3 Illustration of the assumed learning dynamics for our theoretical convergence analysis. The weights $w_j^{(t)}$ of all clients start at the same point and iteratively approach similar local optima, thus decreasing the distance to the aggregate $w^{(t)}$

bounding constants. Note that our analysis is agnostic regarding the precise shape of the bound sequence $C_1, C_2, \ldots$. However, we do assume that the bounds are tighter for our optimized strategy because the sampling strategy is chosen such that a smaller local loss is likely. Also, note that tighter bounds in (5) also make it more likely that a similar local optimum is achieved, hence improving the bounds in (4). In other words, our assumptions also support each other.

Using our assumptions, we can now show that local loss convergence (which we support via our optimized sampling strategy) is sufficient to imply convergence in the federated loss (1). Please refer to Appendix A for details on the proof of the theorem.

Theorem 1 *If assumptions 3, 4, and 5 hold, we obtain:*

$$\mathbb{E}\left[F(w^{(t)}) - F(w^*)\right] \leq C_t + L \cdot D_t, \quad \forall t \tag{6}$$

To summarize, we believe that our procedure ensures tighter bounds for assumptions 4 and 5, thus also improving global loss convergence. We also note that our bound in Theorem 1 is rather conservative: Inspecting Fig. 3, we see that there may be many cases where the aggregated weights achieve even a *better* loss than the single client weights. Still, we provide a worst-case upper bound.

6 Empirical evaluation

6.1 Experimental setup

In order to foster reproducibility, we outline all relevant information on data and models used during our analyses.

Datasets We select four publicly available datasets for the following analyses, all of which were frequently applied in FL research (e.g. [8, 9, 55, 73]). More precisely, we opt for *Covtype*[2], *Diabetes*[2], *Postures*[2], and *MNIST* [74]. *Covtype* is known to be a particularly challenging task in FL when facing data imbalance [8], making it a perfect fit for our analyses. Moreover, Covtype, Diabetes, and Postures contain tabular data, whereas MNIST contains image data on handwritten digits. Finally, our choice of dataset requires binary as well as multi-class classification and significantly differs in the overall data quantity. Accordingly, we argue that our choice of datasets is sufficiently diverse to derive meaningful conclusions from our findings. Table 1 provides some descriptive statistics about the datasets used.

Table 1 Dataset statistics

Name	Samples	Features	Classes
Covtype	581,012	54	2
Diabetes	101,767	37	2
Postures	78,095	15	5
MNIST	70,000	28*28	10

Data setting In what follows, we outline the procedure to simulate a FL setting using the previously mentioned datasets.[3] First of all, we split each dataset D into 80% train (D^{Train}) and 20% test (D^{Test}) data. The test set D^{Test} will be held out from all subsequent steps and serves only to evaluate the performance of the FL models.

Per default, we distribute data samples from D^{Train} among 100 simulated clients ($m = 100$) in accordance with a Dirichlet distribution and its concentration parameter α set to 2. Dirichlet distributions are commonly used to obtain prior distributions in Bayesian statistics and are suitable for simulating real-world data distributions [8]. It was first used to simulate FL settings described by Yurochkin et al. [75] and found widespread application since then. In short, the Dirichlet distribution controls for the severity of label imbalance among the clients, where small α values imply high imbalance [9]. More precisely, our work follows Li et al. [8] and we sample $p_k \sim \text{Dir}_N(\alpha)$ and allocate a $p_{k,j}$ proportion of instances of class k to client j, where $\text{Dir}(\cdot)$ denotes the Dirichlet distribution and α the distribution's concentration parameter [8]. Setting $\alpha = 2$ causes a moderate degree of imbalance within the cohort and ensures that the learning task does not become too trivial due to homogeneous data. Avoiding homogeneity of data is also important, as applying data sampling to perfectly balanced datasets does not hold any value. Afterwards, we perform modifications

[2] Received from the UCI Machine Learning Repository.

[3] Note that simulating FL environments using a centrally available dataset is necessary due to the very limited amount of available real-world FL datasets and in order to isolate the effect of the respective data-specific properties on performance and convergence. Therefore, simulating FL environments is a common approach in existing FL literature [30, 34].

to the data setting in order to test for the impact of the above-mentioned data-specific properties on FL models (as described in Sect. 4.3). These modifications are described in more detail hereafter.

As conducted in similar studies (e.g. [8, 9]), we simulate *data imbalance* by enforcing label imbalance among clients. This is achieved by controlling the aforementioned concentration parameter α. In order to simulate different magnitudes of data imbalance, we decrease α step-wise starting with $\alpha = \inf$ (perfectly homogeneous data) until $\alpha = 0.1$ (severely imbalanced data) is reached.

In order to simulate smaller *cohort sizes*, we randomly remove clients from the default cohort (with $m = 100$ and $\alpha = 2$) until the threshold of $m_{\max}$ clients is met and train the FL model among the remaining clients. Decreasing $m_{\max}$ iteratively during benchmarking reduces not only the cohort size but also the overall number of data samples contained in the cohort.

Finally, we simulate decreasing numbers of *samples per client* by randomly removing as many samples from each client's local data as required in order to meet the threshold $n_{\max}$ defined for each simulation. Thus, we limit the amount of data each client and, in turn, the cohort can learn from. Again, we initially rely on the default cohort configuration with $m = 100$ and $\alpha = 2$ for this simulation.

For each setting, FL is performed with all the aforementioned local sampling strategies, namely *None*, *Constrained*, *Undersampling*, *Oversampling*, *Dynamic*, *Hybrid*, and our proposed sampling strategy named *Optimized*. After each iteration, the performance of the respective FL model is evaluated by measuring the mean micro-averaged F1-score achieved on D^{Test} using fivefold cross-validation in all of our experiments.

FL Setting For all three tabular datasets, we deploy neural networks consisting of three linear layers. Their input layer is of size f (where f is the number of features present in the respective dataset), the hidden layers are of sizes $0.75f$, $0.5f$, and $0.25f$, and the output layer has the size of classes contained in the data. Further, we apply ReLU-activation for all but the output layer and apply dropout of 0.2.

For *MNIST*, we utilize a CNN consisting of two convolutional layers, each with a kernel size of 5. After applying 2×2 max pooling, the network stacks two linear layers (again with ReLU-activation), where the output size of the final layer equals the number of classes within the dataset.

Both model architectures are inspired by Li et al. [8], who successfully applied them to various FL settings in order to empirically investigate the effects of data imbalance on FL performance.

As FL aggregation strategy, we apply *FedAvg* [4], the de facto standard due to its popularity and widespread application [8]. Training is applied for 200 rounds, during each of which all clients perform 3 epochs of local training.

Note that it is not the scope of this work to apply models achieving SOTA performance, but to demonstrate improvements through data sampling for arbitrary models. During the later demonstration on real-world data, however, we apply all aforementioned local sampling strategies to *SOTA models* to confirm the validity of our results on significantly more complex models.

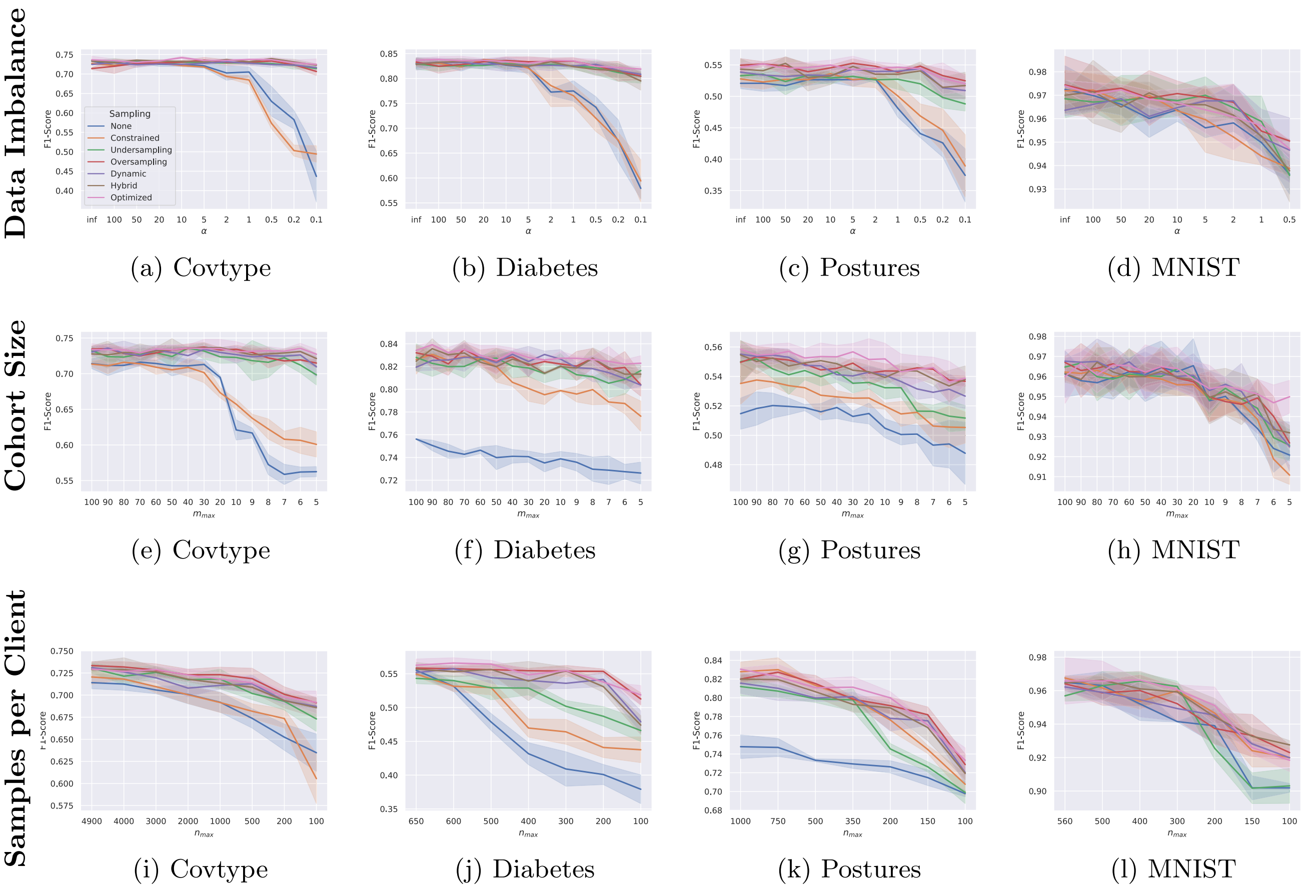
6.2 Impact of data sampling on model performance

In order to measure the effects of local data sampling on FL in face of the above-mentioned data-specific properties affecting FL performance, we follow the methodology outlined in Sect. 4 and iteratively modify the severity of them in simulated cohorts, apply each sampling strategy, and measure FL performance in terms of F1-score afterwards. Figure 4 outlines the performance measured for each setting, sampling strategy, and dataset.

The results suggest that applying local data sampling, regardless of the actual method used, improves the performance of the FL model in most cases. The few exceptions are for datasets *Covtype* and *Diabetes*, where local data sampling neither harms nor improves performance significantly when label imbalances are small ($\alpha > 2$).

Data imbalance Considering *data imbalance*, we find that the performances for *None* and *Constrained* drop rapidly for $\alpha < 2$, whereas all other sampling strategies maintain the same level of performance and show only marginal performance deterioration for severe data imbalance. The only exception from this is *MNIST*, where local sampling indeed improves performance of the federated models but suffers from data imbalance similarly to the baseline-settings *None* and *Constrained*. This shows the performance improvements achieved by applying data sampling when facing imbalanced data. Among all sampling strategies, *Oversampling* and our proposed *Optimized* are the two best-performing approaches, in particular when facing high data imbalance.

Cohort size In terms of *cohort size*, we identify similar patterns yet the magnitudes of improvements are smaller. For *Covtype*, *Postures*, and *MNIST*, FL without data sampling (i.e. *None* and *Constrained*) suffers for settings with $m_{\max} < 30$, causing the model performance to drop significantly. For *Diabetes*, there is a much more steady decrease in performance for decreasing cohort sizes and also a large difference between *None* and *Constrained*. For Fig. 4e–g, we uncover that decreasing $m_{\max}$ correlates with larger improvements using local data sampling. For *MNIST*, improvements through data sampling again remain steady throughout the decrease of $m_{\max}$. However, among all data sampling strate-



(a) Covtype (b) Diabetes (c) Postures (d) MNIST

(e) Covtype (f) Diabetes (g) Postures (h) MNIST

(i) Covtype (j) Diabetes (k) Postures (l) MNIST

Fig. 4 Model performance (mean and standard deviation of fivefold cross-validation)

gies, we see that no strategy yields significantly better performance compared to the others for large FL cohorts. However, as cohort sizes decrease, improvements of *Optimized* become more evident. Additionally, results for the *Postures* dataset also reveal that *Undersampling* is less effective compared to its alternatives when cohorts and their overall data counts become increasingly smaller.

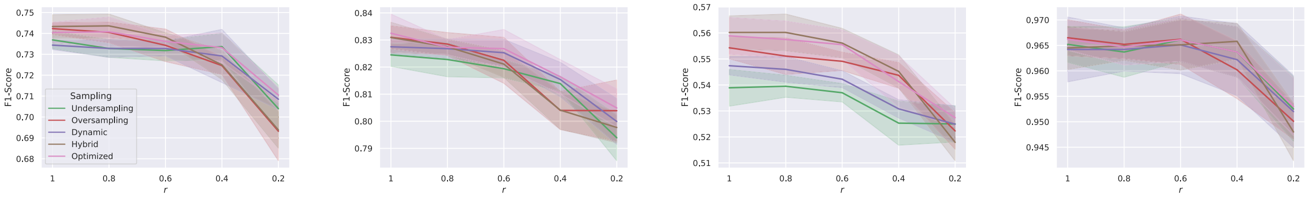
Samples per client Looking at our findings regarding the number of *samples per client* reveals the following findings: first, neither data sampling strategy can compensate for the decrease in sample size to the same extent they do when facing increasing data imbalance or decreasing cohort sizes. Second, *Oversampling* and *Optimized* are again among the best-performing data sampling strategies. Third, *Undersampling* performs significantly worse than all alternatives for small n_{max} and even drops below the performance of FL without data sampling on *Postures* and *MNIST*. This is most likely due to the fact that it even further decreases the amount of data to learn from and the performance of deep learning models hinges on the availability of larger datasets [68].

Overall, we find local data sampling strategies to offer vast improvements in terms of performance compared to our two baselines. In total, *Oversampling* and *Optimized* perform best among all datasets and data specifications, with *Optimized* having a slight edge over the former for increasingly small cohort sizes.

We previously considered the ratio of minority class size and majority class size after data sampling to be $r = 1$, i.e. all classes are sampled to the same size. In Fig. 5, we now investigate how reducing r affects model performance.

The results show that among all datasets, reducing r causes the respective model performance to decrease, regardless of the applied sampling strategy. Considering our findings among all datasets, we argue that while applications with $r = 1$ yield the best performance, reducing r up to $r = 0.6$ does not affect performance much, but reduces the computational overhead during sampling and subsequent training.

Finally, we conclude that applying local data sampling strategies in FL is a good choice when optimal performance is the main objective. Regarding the choice of sampling strategy, we find that unlike previous studies that mainly concluded that the choice of sampling strategy does not significantly affect the final performance (e.g. [16]), certain data-specific properties affect the efficiency of these strategies to different extents. For clients with particularly



(a) Covtype (b) Diabetes (c) Postures (d) MNIST

Fig. 5 Performance impact of class ratio after data sampling

imbalanced or small local datasets, for example (either induced by small cohort sizes or few samples per client), we identify *Oversampling* and *Optimized* as sampling strategies with the highest overall performance. To avoid deteriorating performance, setting $r > 0.6$ is recommended.

6.3 Impact of data sampling on model convergence

Following existing studies that evaluate the effectiveness of FL algorithms [46, 55, 76], we study the impact of local data sampling strategies on model convergence by measuring the minimal rounds required to reach a certain performance target. More precisely, we set the target F1-score for *Covtype* and *Diabetes* to 0.7, and to 0.5 and 0.8 for *Postures* and *MNIST*, respectively. These targets are set in accordance with our previous findings as well as similar studies on FL convergence [54, 55]. In Fig. 6, we plot the number of federated rounds of training required to meet the target performance previously set. (The cut-off at 50 rounds serves to ensure good scaling, as training rarely exceeds 50 rounds.)

Data imbalance The results from our model convergence analysis depicted in Fig. 6 reveal that applying data sampling strategies indeed increases the convergence rate of the respective FL model. In particular, the convergence of FL models without data sampling in place starts to worsen when *data imbalance* increases ($\alpha < 5$). Except for *Postures*, where no data sampling approach yields a model that reaches the target performance for $\alpha = 0.1$, local data sampling significantly decreases the number of rounds required, especially for cohorts with particularly high data imbalance.

Cohort size From our analysis on the *cohort size*, we find that local data sampling can speed up convergence significantly for all cohort sizes. As $m_{\max}$ decreases, improvements in terms of model convergence increase even further. Considering the *Diabetes* dataset as an example, the federated model *None* does not meet the target performance of 0.7 for cohorts of 5 clients. In contrast, using any local data sampling strategy would allow to meet that target within 12 or less rounds of training.

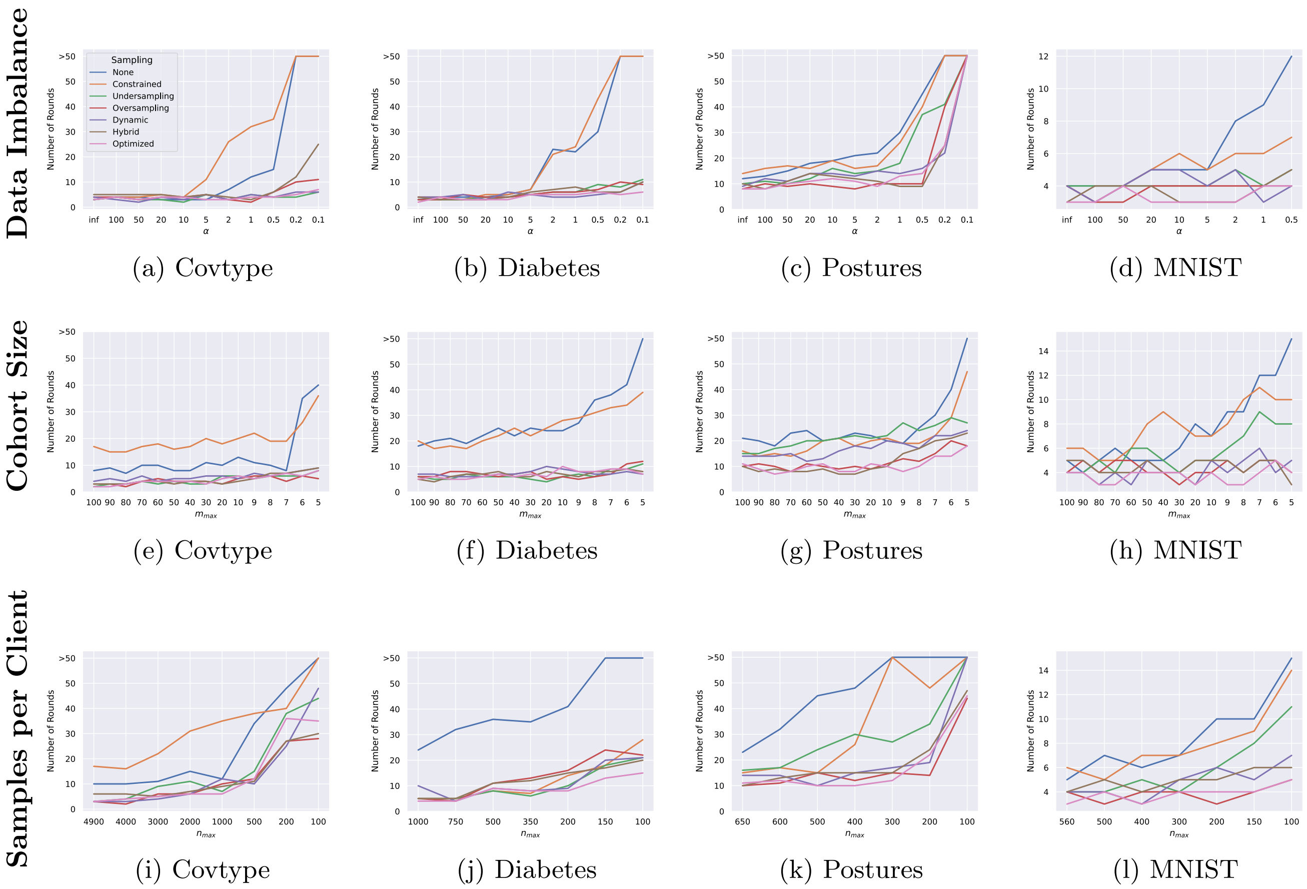
Samples per client Similar to previous finding on model performance, neither sampling strategy can maintain the same magnitude of improvement over the baselines during our analysis of the number of *samples per client*. With few exceptions, however, FL applications with clients that hold limited amounts of data can still improve model convergence using local data sampling. In particular, relying on oversampling or locally optimizing the choice of sampling strategy facilitates fast convergence.

Although there is no sampling strategy significantly better than the other when facing data imbalance, *Oversampling* and *Optimized* are again the overall best strategies to deal with small cohorts and limited local data. Moreover, we find that applying *Undersampling* to multi-class datasets requires more rounds of training than the other sampling strategies, posing a limitation to its applicability in arbitrary environments.

6.4 Impact of data sampling on training time

In addition to previous analyses and inspired by Wang and colleagues [46], we take the effects of local data sampling on the training time into account, as altering the amount of data involved during training correlates with training time per round. To account for this, we measure both the number of seconds taken for each training round per client and the overall training time required to train a federated model to achieve the target performances defined in Sect. 6.3. Unlike previous analyses, we only consider the label imbalance in this regard, because it directly affects the magnitude of imbalance present at each client. This local imbalance then determines the required amount of samples being undersampled or oversampled, which in turn affects the overall training time. Finally, it is worth mentioning that all models were trained on the same hardware, namely a single NVIDIA Tesla V100 (16GB) and batch-size set to 16.

Figure 7 summarizes the results of our analysis of the training time per round and client. Furthermore, it outlines the number of clients excluded from training. From the graphs, we find that the training time for *None* and *Constrained* remains unaffected by the change in α, as the total amount of data does not change for this setting. For *Oversampling* and *Hybrid*, training time increases significantly en par with an increased imbalance. At $\alpha = 0.1$, it

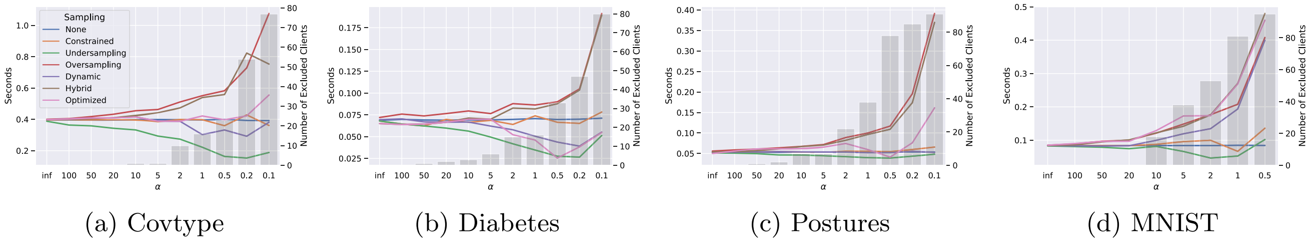


(a) Covtype (b) Diabetes (c) Postures (d) MNIST

(e) Covtype (f) Diabetes (g) Postures (h) MNIST

(i) Covtype (j) Diabetes (k) Postures (l) MNIST

Fig. 6 Model convergence (median of fivefold cross-validation)



(a) Covtype (b) Diabetes (c) Postures (d) MNIST

Fig. 7 Training time per round and client and number of excluded clients

exceeds the baseline *None* 2.5–8 times, depending on the dataset. As argued previously, this is likely due to the increasing discrepancy of minority and majority class size at each client, necessitating larger amounts of data being oversampled. Regarding *Dynamic*, we find that the training time is similar to that of *None* for *Covtype* and *Postures*, smaller than the baselines for *Diabetes*, and significantly higher on *MNIST*. Unsurprisingly, the training time per client decreases when *Undersampling* is in place. This is due to the reduction in local dataset sizes. Finally, the training time per round and client varies among datasets for our proposed sampling strategy named *Optimized*. More precisely, its training time is similar to *None* on *Covtype*, smaller than it on *Diabetes*, and greater on the remaining datasets. As later analyses will reveal, this is due to the fact that the locally optimized choice of sampling strategy differs notably among these datasets.

Plotting the number of excluded clients (i.e. clients not holding samples of every output class) reveals a steep increase in number for $\alpha < 1$. At its peak, the significant increase in global data imbalance causes around 80% of clients to be excluded from the training entirely. Still, as our previous analyses have shown, this reduction in active clients combined with the application of local data sampling increases FL model performance and convergence significantly.

Table 2 Overall training time for $\alpha = 2$

Method (s)	Covtype (s)	Diabetes (s)	Postures (s)	MNIST (s)
None	278	182	123	67
Constrained	934	40	80	28
Undersampling	74	15	53	11
Oversampling	136	47	88	33
Dynamic	105	21	53	25
Hybrid	170	52	89	25
Optimized*	108	23	60	24

*not including time for local optimization of sampling strategy

Besides benchmarking the training time per round and client, we also investigate the overall time required to train a FL model that achieves the target performances per dataset we previously defined. This is motivated by the fact that monitoring training time per round alone does not take our previous findings on the improved model convergence through local data sampling into account. Accordingly, we also benchmark sampling strategies with respect to the overall training time. This combines various previous findings, as the overall training time is defined as number of rounds required to reach the target performance multiplied by training time per round per clients and the number of active (i.e. non-excluded) clients. Note that communication time, i.e. time required to send and receive model updates, is not included, as it heavily depends on external factors such as available bandwidth and latency.

Table 2 outlines the overall training time per dataset and sampling strategy when applied to the default data setting ($m = 100$ and $\alpha = 2$). It shows that *Undersampling* has the lowest overall training time regardless of the dataset, converging 2.5–12 times faster then the respective *None* setting. The improvements are due to the improved convergence rate despite fewer clients participating and the reduced data quantity. More surprisingly, however, all other sampling strategies improve the overall training time, too.

While our results have shown for such data sampling strategies that they increase the training time per round and client, this is mostly due to their improved rate of convergence compared to FL without data sampling. Among these strategies, *Oversampling* requires the largest amount of time, usually followed by *Hybrid* sampling. Furthermore, the table shows that *Dynamic* and *Optimized* achieve very similar overall training times.

Overall, our findings allow to conclude that *Undersampling* is the data sampling strategy of choice when computational time and resources are limited. Moreover, we find all data sampling strategies contained in this work to improve the overall training time of FL models. Considering previous findings from Sect. 6.2, local data sampling does not only improve the training time of FL models but also their performance in terms of F1-score.

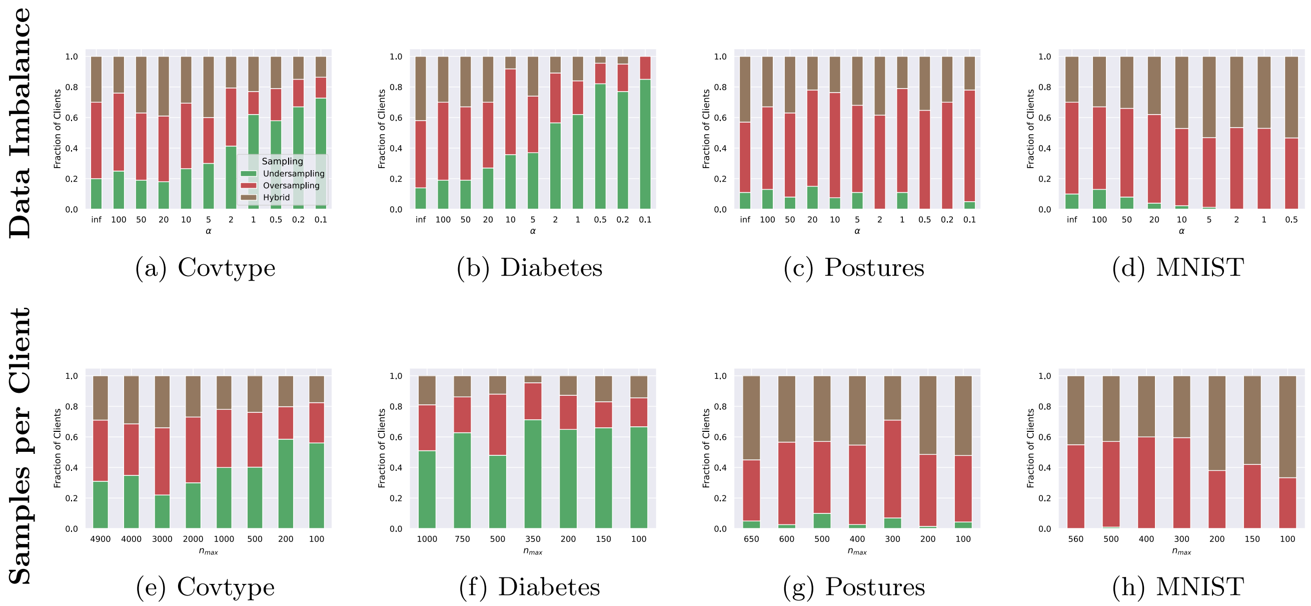
6.5 Optimal choice of sampling strategy

Considering all previous analyses, we argue that our proposed strategy named *Optimized* is the best choice of local data sampling strategies for FL to achieve high performance and fast convergence in a reasonable amount of time. Accordingly, we recommend it for future integration into FL deployment processes of various kinds. In order to reliably deploy locally optimized data sampling, however, we finally have to explore the factors influencing the choice of optimal data sampling strategy at each client's side. Knowledge about the inner workings of the strategy not only increases its reliability but also its trustworthiness and reproducibility.

Therefore, we visualize each client's locally optimized choice of data sampling strategy in Fig. 8. In it, we do not include cohorts with reduced amounts of clients, as the number of clients does not affect each client's local choice at all. Consequently, the figure shows the percentage of clients within each cohort applying either oversampling, undersampling, or hybrid sampling when increasing *data imbalance* or decreasing the amount of *samples per client*.

Figure 8 reveals some distinct differences in terms of local optimization between the two binary classification datasets (*Covtype* and *Diabetes*) and the two multi-class datasets (*Postures* and *MNIST*). The most dominant difference lies in the prevalence of undersampling as optimal sampling strategy. In case of binary classification tasks, the share of clients choosing undersampling instead of over- and hybrid sampling increases with increasing imbalance or decreasing samples per client. In terms of increasing data imbalance, this suggests that locally oversampling large amounts of data (which is necessary as the discrepancy between local minority and majority class size increases) is less effective. For the two multi-class datasets on the other hand, the number of clients opting for undersampling is vanishingly small. Although the share of clients preferring oversampling remains mostly consistent throughout both analyses, the number of clients applying hybrid sampling increases slightly.

Unfortunately, optimizing the choice of data sampling strategy locally requires clients to spend additional time and resources prior to FL application. To save these resources and



(a) Covtype (b) Diabetes (c) Postures (d) MNIST

(e) Covtype (f) Diabetes (g) Postures (h) MNIST

Fig. 8 Choice of sampling strategy in *Optimized*

also to justify or previous decision not to include the time for local optimization into the overall training time reported in Table 2, we lastly aim to provide the coefficients of correlation derived from a logistic regression model between a set of data properties and each client's respective optimal sampling strategy. Based on these coefficients, future decision-makers may identify the sampling strategy that will most likely yield the optimal performance without expensive local optimization. The logistic regression model is fitted on the data reported throughout all our previous experiments. Moreover, the set of independent variables representing each client consists of four variables, namely: the number of samples held by the client, the number of input features for the federated model, the number of output classes, and the local imbalance LI_j that is defined as

$$\mathrm{LI}_j = \frac{n^j_{\text{Majority}}}{n^j_{\text{Minority}}}, \tag{7}$$

where n^j_{Majority} and n^j_{Minority} refer to the majority and minority class sizes of client j.

The correlation coefficients reported in Table 3 confirm our previous findings on the number of classes and their effect on the choice of sampling strategy. Here, the negative coefficient for undersampling and the positive coefficients for hybrid sampling and oversampling prove that these strategies are to be preferred in multi-class settings. Moreover, local imbalance shows a positive correlation with undersampling but negative correlation with the other two options. Accordingly, clients with high discrepancy between majority and minority class size are likely to benefit from applying undersampling. Finally, the number of samples and the number of features have the weakest correlation with either class, suggesting that they are less important for decision making. However, we find that clients with large sample sizes can benefit the most from applying oversampling, whereas tasks with many features show highest positive correlation with hybrid sampling.

Table 3 Logistic regression coefficients per sampling strategy in *Optimized* from all previous experiments

	Undersampling	Hybrid	Oversampling
# Samples	−0.0005*	0.0002*	0.0003*
# Features	0.0096*	0.0253*	−0.0349*
# Classes	−1.0296*	0.7176*	0.3119*
Local Imbalance	0.1211*	−0.0335	−0.0876*

*significant at $p < 0.05$

7 Demonstration on real-world and large-scale data

After finishing the *empirical* evaluation using synthesized FL environments, we finally seek to confirm our previous findings on real-world and large-scale data with SOTA models. Ultimately, this serves to prove that our previous findings are of practical value.

This demonstration addresses four limitations of our previous analyses, namely the use of *synthetic FL environments*, *smaller datasets*, *non-SOTA models*, and ignoring the *local optimization time* for our proposed sampling strategy. In the following, we will elaborate upon how these limitations are addressed during the demonstration and provide details on data and models used.

Synthetic FL environments While the use of synthetic data splits was necessary to systematically study the impact of the three data-specific properties and their severity on the value of local sampling strategies, previous studies found that "[s]uch synthetic partition approaches may fall short of modelling the complex statistical heterogeneity of real federated datasets" [77, p.3]. Evaluating novel FL methods on real data splits instead ensures their capability of addressing real-world challenges [77].

Therefore, we use the *FEMNIST* dataset obtained from the FL benchmark corpus LEAF [78] in Sect. 7.1. *FEMNIST* has been created by partitioning the *EMNIST* dataset (an extension to MNIST which also contains letters) [79], according to the writers' identities contained in the data. In turn, *FEMNIST* splits data naturally, thus providing a real-world data partitioning suitable for our demonstration. It contains 3.550 clients with a total of 805.263 samples [78]. However, we decide to limit our demonstration to numerical digits as contained in the MNIST dataset in order to allow for a better comparison of results.

Smaller datasets Moreover, the datasets used in Sect. 6 are not particularly large (the largest one being *Covtype* with about 600,000 samples) and split among at most 100 clients. Therefore, our previous analyses might fall short on capturing data sampling effects unique to large-scale datasets and environments.

To this end, we also demonstrate the effectiveness of local data sampling and our proposed approach on the *KDD Cup 1999* dataset [80] in Sect. 7.2. The dataset contains about 4,900,000 samples used for network intrusion detection based on 41 different features [80]. To simulate the large-scale FL environment, we split the train dataset D^{Train} among 10,000 simulated clients. To simulate a realistic degree of data imbalance, we set $\alpha = 0.2$.

Non-SOTA models Our previous choices of models were inspired by the studies of Li et al. [8] on the impact of data imbalance on FL and in accordance with various previous studies on this topic (e.g. [5, 9]). Despite these models achieving reasonable performance on the datasets used in our study, it remains open, if and to what extent our findings generalize to larger and more complex models used for prediction. To resolve this uncertainty, we train a model following the VGG-16 architecture [81] to classify the images from the *FEMNIST* dataset (see Sect. 7.1). VGG-16 consists of 13 convolutional layers, 5 max pooling layers, and 3 dense layers [81]. For the classification of *KDD Cup 1999* data in Sect. 7.2, we rely on a neural network similar to those outlined in Sect. 6.1. However, we extend them to five hidden layers with 128 neurons each.

Local optimization time We previously argued that monitoring correlations of different data-specific properties with the choice of optimal sampling strategy (as conducted in Sect. 6.5) may allow clients to choose the local sampling strategy that likely provides the best outcome for the cohort without having to perform costly local optimization. In order to validate whether this assumption was justified, we include the sampling strategy named *Optimized_coef* into our following demonstrations. Unlike the previously proposed *Optimized*, *Optimized_coef* relies on the correlation coefficients presented in Table 3. To do so, clients first calculate their respective local imbalance LI_j in accordance with Eq. 7. Subsequently, they determine the optimal sampling strategy s^* from the set of available sampling strategies S that maximizes the following equation:

$$s^* = \arg\max_{s \in S} n_j \beta_n^s + f_j \beta_f^s + c_j \beta_c^s + LI_j \beta_{LI}^s, \tag{8}$$

where n_j, f_j, and c_j refer to the number of samples, features, and classes of client j, whereas β_n^s, β_f^s, β_c^s, and β_{LI}^s are the coefficients of correlation for the number of samples, features, classes, and the local imbalance of sampling strategy s, as presented in Table 3.

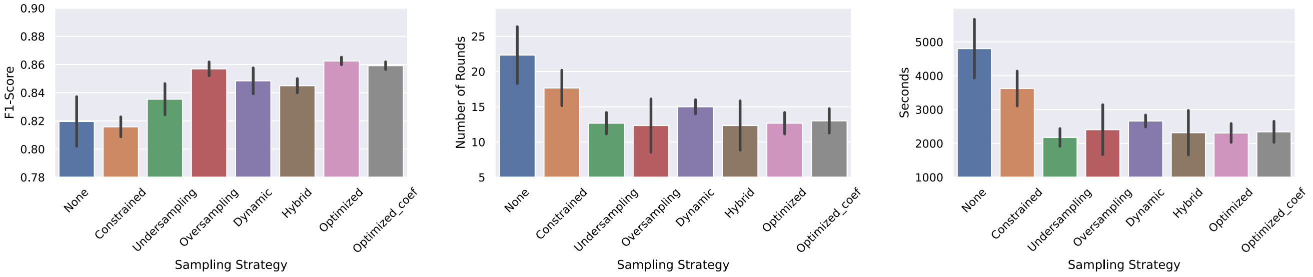
7.1 Real-world data

Figure 9 depicts the results in terms of performance, convergence[4], and overall train time from applying each local sampling strategy to *FEMNIST* before training a VGG-16 model using *FedAvg*.

Figure 9a shows VGG-16's performance for all local sampling strategies. Similar to our previous findings, both baselines perform significantly worse than settings were data sampling was applied. Moreover, oversampling as well as our two proposed sampling strategies *Optimized* and *Optimized_coef* slightly outperform their alternative and yield the overall best F1-scores. Another interesting aspect is that the coefficient-based choice of optimal sampling strategy performs only marginally worse compared to the computationally much more expensive *Optimized* sampling.

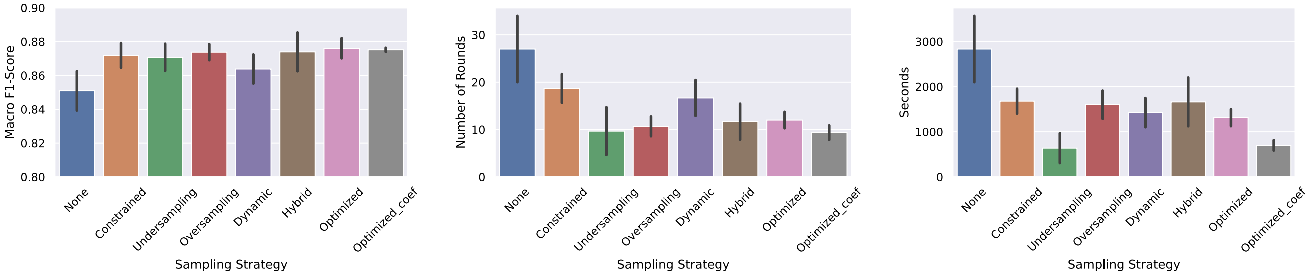
Next, the results on model convergence in Fig. 9b also confirm our findings. They show that applying local data sampling has the potential to reduce the number of rounds required to achieve the target performance by up to 50%. Interestingly—with the exception of dynamic sampling—all sampling strategies require a very similar number of rounds to reach convergence. This includes the coefficient-based sam-

[4] In line with our previous analyses on model convergence for MNIST, we set the target F1-score for *FEMNIST* to 0.8.



(a) Performance (b) Convergence (c) Overall Training Time

Fig. 9 Performance, convergence, and overall training time for the *FEMNIST* dataset using VGG-16 (mean and standard deviation of fivefold cross-validation)



(a) Performance (b) Convergence (c) Overall Training Time

Fig. 10 Performance, convergence, and overall training time for the *KDD Cup 1999* dataset (mean and standard deviation of fivefold cross-validation)

pling strategy *Optimized_coef* that converges in almost the same number of rounds as *Optimized*.

Finally, we visualize the overall training time that again consists of the number of rounds for convergence, the train time per client and round, as well as the number of active clients per round. It shows that, despite some of the sampling strategies increasing the overall sample size contained in the cohort, the speed-up convergence causes the overall training time to decrease significantly compared to the settings without data sampling in place. Unsurprisingly, undersampling is again the sampling strategy with the lowest overall training time, followed by both optimized approaches and hybrid sampling. (However, hybrid sampling has significantly higher standard deviation, making it harder to judge it reliability.)

7.2 Large-scale data

Similar to Fig. 9, Fig. 10 depicts performance, convergence[5], and overall train time for the *KDD Cup 1999* dataset using various local data sampling strategies.

The findings regarding the macro-averaged F1-score demonstrate the improvements through either data sampling strategy compared to the *None* baseline setting. In line with previous findings, *Oversampling*, *Hybrid*, *Optimized*, and the coefficients-based *Optimized_coef* yield the best performance. However, the improvements over the alternatives are less pronounced compared to previous analyses. In particular, it is worth mentioning that the mere exclusion of clients that do not hold samples from all classes (as indicated by the *Constrained* setting) accounts for most of the improvement over the baseline.

Figure 10b reveals that even on large-scale datasets, local data sampling can significantly improve the convergence of federately trained models. In particular, most strategies allow to reduce the number of rounds required by more than half. Here, the convergence of *Undersampling* stands out from previous findings, as we now find it to be among the fastest converging sampling strategies.

With respect to the overall training time, Fig. 10c confirms the superiority of local *Undersampling*. Additionally, it reveals that although *Optimized* has a decent training time, *Optimized_coef* allows for a significantly faster training. This is likely due to the fact that the coefficient-based strategy selection caused more clients to chose undersampling compared to optimizing locally. Accordingly, these findings further support our previous claim that *Optimized_coef* is a

[5] Considering the baseline performance without data sampling, we set the target F1-score for *KDD Cup 1999* to 0.85.

viable alternative to *Optimized*, especially when computational resources are limited.

In general, the results of both demonstrations confirm our previous findings presented in Sect. 6. From this we argue that our findings generalize reasonably well for FL on real-world and large-scale data settings as well as more complex SOTA model architectures.

8 Discussion and conclusion

Despite its great potential and theoretical guarantees to overcome the issue of distributed privacy-preserving ML [2, 30, 82], FL suffers from performance degradation when data is imbalanced [8]. To alleviate these drawbacks and improve FL on imbalanced data, both algorithm- and data-based approaches have been proposed in the past [10, 11]. Although the former are usually evaluated carefully and systematically (e.g. [42]), the latter lack a systematic analysis of their potential to improve FL. In turn, FL initiators have to rely on a best-guess- or trial-and-error-based approach when it comes to applying data sampling strategies in their FL environment. Unfortunately, this might necessitate repeating the FL several times to identify the best-performing approach or it might harm performance.

In this work, we address this evident gap in the existing literature by providing a holistic view on local data sampling strategies applied to FL environments and systematically benchmarking their impact on FL performance. Therefore, we identify the most common local data sampling strategies (i.e. *Undersampling*, *Oversampling*, *Dynamic* sampling, and *Hybrid* sampling) as well as relevant data-specific properties negatively affecting FL systems (i.e. *Data Imbalance*, *Cohort Size*, and number of *Samples per Client*). Moreover, we propose a novel local data sampling strategy named *Optimized* that facilitates local optimization of the sampling strategy selection prior to FL participation. Afterwards, we apply the aforementioned sampling strategies to various FL environments where we control for the severity of each data-specific property, to ultimately answer our three initial research questions.

Our findings with respect to ***RQ1*** (*To what extent can local data sampling improve FL performance, convergence, and train time in face of unfavourable data-specific properties?*) suggest that applying data sampling strategies prior to training a federated model increases both the performance of the model in terms of F1-score as well as its rate of convergence. This is mostly in line with similar studies applied to non-distributed settings [83]. For highly imbalanced or particularly small cohorts, we find that local data sampling nearly completely compensates for the decrease in performance and convergence speed caused by the respective influencing data property. With respect to the sample size of clients, similar but less pronounced patterns can be observed. Additionally, we find that the ratio of minority and majority class size after data sampling affects the effectiveness of local data sampling. Here, we show that reducing the ratio below $r = 0.6$ decreases model performance significantly. With respect to the training time per round as well as the overall training time, our findings prove that although some sampling strategies increase the training time per round and client, either data sampling strategy reduces the overall train time due to the improved convergence.

To answer ***RQ2*** (*What are the unique advantages and disadvantages of different sampling strategies over another?*), we find *Oversampling* and *Optimized* yielding the highest overall performance and exhibit the fastest convergence among different datasets and data distributions. However, the choice of sampling strategy affects both the overall and the training time per round. Our findings suggest that although some sampling strategies increase the training time per round and client, either data sampling strategy reduces the overall train time due to the improved convergence. Among all strategies, *Undersampling* requires the least training time, followed shortly by *Dynamic* and *Optimized*.

Finally, regarding ***RQ3*** (*Can locally optimizing the choice of data sampling strategy improve FL further?*), we conclude from our findings that locally optimizing the choice of data sampling strategy is a good choice for various FL applications where high performance or fast convergence is the main objective. Yet, local optimization is computationally expensive and might hence be inapplicable when time and resources are limited. Our results thus shed light on the impact of local optimization, showing, among other things, that the application of undersampling is a promising option in many binary classification settings but less viable for multi-class classification. During demonstration, we show that relying on the coefficients of correlation allows clients to avoid the computationally expensive local optimization without having to compromise on performance, convergence, or training time.

Limitations Currently, the scope of our analyses is limited to three data-specific properties influencing FL performance. However, additional model- or client-specific factors such as model complexity [84–87], model heterogeneity [67, 88, 89], and client dropout rates [2, 90–92] were found to affect FL performance. Furthermore, additional nuances of data imbalance such as partially class-disjoint data, feature imbalance, and quantity imbalance have detrimental effects on FL [8, 9, 49]. In turn, the validity of our findings is yet unknown for these settings.

Moreover, we rely on Cheng et al. [10] who proved empirically that data-based approaches perform similarly, if not superior, compared to algorithm-based approaches and did not compare performance with baselines such as *Gradient Harmonizing Mechanism Classification* [93] ourselves.

Future work In various real-world applications of FL, clients have access to vast amounts of unlabelled data, which currently cannot be facilitated during the supervised training of federated models. And although high labelling costs may prevent clients from utilizing this data during training [94], integrating such data to augment the train data through federated semi-supervised learning [95, 96] seems to be a promising extension to this work.

Moreover, FL might benefit from considering more sophisticated sampling strategies such as oversampling using generative models [57], triplet-based oversampling [73], or similarity-based undersampling [97]. Finally, we seek to complement our work by considering different types of data and models, e.g. recurrent neural networks for time-series data.

Appendix A Proof of Theorem 1

This section proves global loss convergence based on assumptions (3),(4), and (5).

First, we prove an auxiliary bound for the local losses. Using assumption (3), we obtain

$$\left|F_j(w_j^{(t)}) - F_j(w^{(t)})\right| \leq L \cdot \left\|w_j^{(t)} - w^{(t)}\right\|$$

Applying assumption (4) to the right-hand-side and rearranging then yield the bound

$$F_j\left(w^{(t)}\right) \leq F_j\left(w_j^{(t)}\right) + L \cdot D_t. \tag{A1}$$

Now, consider the expected loss difference for the cohort, which by definition equals:

$$\begin{aligned}\mathbb{E}&\left[F\left(w^{(t)}\right) - F\left(w^*\right)\right]\\ &= \mathbb{E}\left[\sum_{j=1}^{N} \frac{n_j}{N} \cdot \left(F_j\left(w^{(t)}\right) - F_j\left(w^*\right)\right)\right]\\ &= \sum_{j=1}^{N} \frac{n_j}{N} \cdot \mathbb{E}\left[F_j\left(w^{(t)}\right) - F_j\left(w^*\right)\right]\end{aligned}$$

Using Equation (A1), we obtain:

$$\begin{aligned}\sum_{j=1}^{N} &\frac{n_j}{N} \cdot \mathbb{E}\left[F_j\left(w^{(t)}\right) - F_j\left(w^*\right)\right]\\ &\leq \sum_{j=1}^{N} \frac{n_j}{N} \cdot \mathbb{E}\left[F_j\left(w_j^{(t)}\right) + L \cdot D_t - F_j\left(w^*\right)\right]\\ &\leq L \cdot D_t + \sum_{j=1}^{N} \frac{n_j}{N} \cdot \mathbb{E}\left[F_j\left(w_j^{(t)}\right) - F_j\left(w^*\right)\right]\end{aligned}$$

because $\sum_{j=1}^{N} \frac{n_j}{N} = 1$.

Finally, we apply assumption (5), yielding:

$$L \cdot D_t + \sum_{j=1}^{N} \frac{n_j}{N} \cdot \mathbb{E}\left[F_j\left(w_j^{(t)}\right) - F_j\left(w^*\right)\right] \leq L \cdot D_t + C_t,$$

which concludes the proof. □

Author Contributions C.D. wrote the main manuscript and prepared the figures. P.C. provided supervision throughout the whole study and contributed to its design. B.P. significantly contributed to the theoretical analysis and provided proofs. All authors revised and reviewed the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. This research is partially funded by the Ministry of Culture and Science of North Rhine-Westphalia under the grant no NW21-059A SAIL.

Data availability The datasets used in this study are available at the following repositories: Covtype, Diabetes, Postures, and KDD Cup 1999 are available at the UCI Machine Learning Repository (https://archive.ics.uci.edu/datasets). The MNIST dataset was received through PyTorch (https://pytorch.org/vision/main/generated/torchvision.datasets.MNIST.html), whereas the FEMNIST dataset is available through LEAF (https://leaf.cmu.edu/).

Declarations

Conflict of interests The authors declare no competing interests.

References

1. McMahan, B., Ramage, D.: Federated learning: collaborative machine learning without centralized training data. Google Research Blog (2017)
2. Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., Zhou, Y.: A hybrid approach to privacy-preserving federated learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, pp. 1–11 (2019)
3. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cum-

mings, R., et al.: Advances and open problems in federated learning. Found. Trends Mach. Learn. **14**(1—-2), 1–210 (2021)

4. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
5. Du, Z., Sun, J., Li, A., Chen, P.-Y., Zhang, J., Li, H.H., Chen, Y.: Rethinking normalization methods in federated learning. In: Proceedings of the 3rd International Workshop on Distributed Machine Learning, pp. 16–22 (2022)
6. Liu, G., Ma, X., Yang, Y., Wang, C., Liu, J.: Federaser: enabling efficient client-level data removal from federated learning models. In: 2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS), pp. 1–10. IEEE (2021)
7. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-iid data. Preprint at arXiv:1806.00582 (2018)
8. Li, Q., Diao, Y., Chen, Q., He, B.: Federated learning on non-iid data silos: an experimental study. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 965–978. IEEE (2022)
9. Düsing, C., Cimiano, P.: Towards predicting client benefit and contribution in federated learning from data imbalance. In: Proceedings of the 3rd International Workshop on Distributed Machine Learning, pp. 23–29 (2022)
10. Cheng, X., Shi, F., Liu, Y., Zhou, J., Liu, X., Huang, L.: A class-imbalanced heterogeneous federated learning model for detecting icing on wind turbine blades. IEEE Trans. Ind. Inf. **18**(12), 8487–8497 (2022)
11. Lu, S., Gao, Z., Xu, Q., Jiang, C., Zhang, A., Wang, X.: Class-imbalance privacy-preserving federated learning for decentralized fault diagnosis with biometric authentication. IEEE Tran. Ind. Inf. **18**(12), 9101–9111 (2022)
12. Wang, H., Kaplan, Z., Niu, D., Li, B.: Optimizing federated learning on non-iid data with reinforcement learning. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, pp. 1698–1707. IEEE (2020)
13. Chakraborty, D., Ghosh, A.: Improving the robustness of federated learning for severely imbalanced datasets. Preprint at arXiv:2204.13414 (2022)
14. Jorge, J., Barros, P., Yokoyama, R., Guidoni, D., Ramos, H., Fonseca, N., Villas, L.: Applying federated learning in the detection of freezing of gait in parkinson's disease. In: 2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC), pp. 195–200. IEEE (2022)
15. Shingi, G.: A federated learning based approach for loan defaults prediction. In: 2020 International Conference on Data Mining Workshops (ICDMW), pp. 362–368. IEEE (2020)
16. Islam, H., Mosa, A., *et al.*: A federated mining approach on predicting diabetes-related complications: Demonstration using real-world clinical data. In: AMIA Annual Symposium Proceedings, vol. 2021, p. 556. American Medical Informatics Association (2021)
17. Deng, Y., Zhou, Y., Liu, G., Wang, J.H., Shui, Y.: Enhancing federated learning by one-shot transferring of intermediate features from clients. In: 2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA), pp. 1–11. IEEE (2023)
18. Younis, R., Fisichella, M.: Fly-smote: re-balancing the non-iid iot edge devices data in federated learning system. IEEE Access **10**, 65092–65102 (2022)
19. Duan, M., Liu, D., Chen, X., Tan, Y., Ren, J., Qiao, L., Liang, L.: Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In: 2019 IEEE 37th International Conference on Computer Design (ICCD), pp. 246–254. IEEE (2019)
20. Chen, J., Guo, Q., Fu, Z., Shang, Q., Ma, H., Wu, D.: Campus network intrusion detection based on federated learning. In: 2022 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2022)
21. Shuai, X., Shen, Y., Jiang, S., Zhao, Z., Yan, Z., Xing, G.: Balancefl: Addressing class imbalance in long-tail federated learning. In: 2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), pp. 271–284. IEEE (2022)
22. Cai, L., Lin, D., Zhang, J., Yu, S.: Dynamic sample selection for federated learning with heterogeneous data in fog computing. In: ICC 2020-2020 IEEE International Conference on Communications (ICC), pp. 1–6. IEEE (2020)
23. Wang, D., Shen, L., Luo, Y., Hu, H., Su, K., Wen, Y., Tao, D.: Fedabc: Targeting fair competition in personalized federated learning. Preprint at arXiv:2302.07450 (2023)
24. Wang, H., Muñoz-González, L., Eklund, D., Raza, S.: Non-iid data re-balancing at iot edge with peer-to-peer federated learning for anomaly detection. In: Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks, pp. 153–163 (2021)
25. Yang, W., Zhang, Y., Ye, K., Li, L., Xu, C.-Z.: Ffd: A federated learning based method for credit card fraud detection. In: Big Data–BigData 2019: 8th International Congress, Held as Part of the Services Conference Federation, SCF 2019, San Diego, CA, USA, June 25–30, 2019, Proceedings 8, pp. 18–32. Springer (2019)
26. Zha, D., Bhat, Z.P., Lai, K.-H., Yang, F., Hu, X.: Data-centric ai: Perspectives and challenges. In: Proceedings of the 2023 SIAM International Conference on Data Mining (SDM), pp. 945–948. SIAM (2023)
27. Whang, S.E., Roh, Y., Song, H., Lee, J.-G.: Data collection and quality challenges in deep learning: a data-centric ai perspective. VLDB J. **32**(4), 791–813 (2023)
28. Harasic, M., Keese, F.-S., Mattern, D., Paschke, A.: Recent advances and future challenges in federated recommender systems. Int. J. Data Sci. Anal. **17**, 1–21 (2023)
29. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol. **10**(2), 1–19 (2019)
30. Varlamis, I., Sardianos, C., Chronis, C., Dimitrakopoulos, G., Himeur, Y., Alsalemi, A., Bensaali, F., Amira, A.: Using big data and federated learning for generating energy efficiency recommendations. Int. J. Data Sci. Anal. **16**(3), 353–369 (2023)
31. Lincy, M., Kowshalya, A.M.: Early detection of type-2 diabetes using federated learning. Int. J. Sci. Res. Sci. Eng. Technol. **12**, 257–267 (2020)
32. Düsing, C., Cimiano, P.: Federated learning to improve counterfactual explanations for sepsis treatment prediction. In: International Conference on Artificial Intelligence in Medicine, pp. 86–96. Springer (2023)
33. Gebremeskel, G.B.: Leveraging big data analytics for intelligent transportation systems: optimize the internet of vehicles data structure and modeling. Int. J. Data Sci. Anal., 1–16 (2023)
34. Wang, X., Meng, S., Chen, Y., Liu, Q., Yuan, R., Li, Q.: Federated deep recommendation system based on multi-view feature embedding. In: 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA), pp. 1–9. IEEE (2022)
35. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K.: Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1175–1191 (2017)
36. Wu, Y., Dong, S., Zhou, Y., Zhao, Y., Fu, F., Yang, T., Niu, C., Wu, F., Cui, B.: Kvsagg: Secure aggregation of distributed key-value sets. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE). IEEE (2023)
37. Li, K.H., Gusmão, P.P.B., Beutel, D.J., Lane, N.D.: Secure aggregation for federated learning in flower. In: Proceedings of the 2nd

ACM International Workshop on Distributed Machine Learning, pp. 8–14 (2021)

38. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proc. Mach. Learn. Syst. **2**, 429–450 (2020)
39. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
40. Cui, Y., Jia, M., Lin, T.-Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9268–9277 (2019)
41. Tuor, T., Wang, S., Ko, B.J., Liu, C., Leung, K.K.: Overcoming noisy and irrelevant data in federated learning. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 5020–5027. IEEE (2021)
42. Zhou, F., Yang, Y., Wang, C., Hu, X.: Federated learning based fault diagnosis driven by intra-client imbalance degree. Entropy **25**(4), 606 (2023)
43. Zhou, Z.-H., Liu, X.-Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Trans. Knowl. Data Eng. **18**(1), 63–77 (2005)
44. Khan, S.H., Hayat, M., Bennamoun, M., Sohel, F.A., Togneri, R.: Cost-sensitive learning of deep feature representations from imbalanced data. IEEE Trans. Neural Netw. Learn. Syst. **29**(8), 3573–3587 (2017)
45. Kim, M., Yu, S., Kim, S., Moon, S.-M.: Depthfl: Depthwise federated learning for heterogeneous clients. In: The Eleventh International Conference on Learning Representations (2022)
46. Wang, Y., Tong, Y., Zhou, Z., Zhang, R., Pan, S.J., Fan, L., Yang, Q.: Distribution-regularized federated learning on non-iid data. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE), pp. 2113–2125. IEEE (2023)
47. Zhang, J., Li, Z., Li, B., Xu, J., Wu, S., Ding, S., Wu, C.: Federated learning with label distribution skew via logits calibration. In: International Conference on Machine Learning, pp. 26311–26329. PMLR (2022)
48. Gu, Z., Zhang, K., Bai, G., Chen, L., Zhao, L., Yang, C.: Dynamic activation of clients and parameters for federated learning over heterogeneous graphs. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE). IEEE (2023)
49. Fan, Z., Yao, J., Han, B., Zhang, Y., Wang, Y., et al.: Federated learning with bilateral curation for partially class-disjoint data. Adv. Neural Inf. Process. Syst. **36** (2024)
50. Li, X.-C., Zhan, D.-C.: Fedrs: Federated learning with restricted softmax for label distribution non-iid data. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 995–1005 (2021)
51. Fang, C., He, H., Long, Q., Su, W.J.: Exploring deep neural networks via layer-peeled model: minority collapse in imbalanced training. Proc. Natl. Acad. Sci. **118**(43), 2103091118 (2021)
52. Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X., He, B.: A survey on federated learning systems: vision, hype and reality for data privacy and protection. IEEE Trans. Knowl. Data Eng. **35**, 3347 (2021)
53. Yoshida, N., Nishio, T., Morikura, M., Yamamoto, K., Yonetani, R.: Hybrid-fl for wireless networks: Cooperative learning mechanism using non-iid data. In: ICC 2020-2020 IEEE International Conference On Communications (ICC), pp. 1–7. IEEE (2020)
54. Guo, Y., Tang, X., Lin, T.: Fedbr: Improving federated learning on heterogeneous data via local learning bias reduction. In: International Conference on Machine Learning, pp. 41354–41381. PMLR (2023)
55. Ou, J., Shen, Y., Wang, F., Liu, Q., Zhang, X., Lv, H.: Aggenhance: aggregation enhancement by class interior points in federated learning with non-iid data. ACM Trans. Intell. Syst. Technol. **13**(6), 1–25 (2022)
56. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor. Newsl. **6**(1), 20–29 (2004)
57. Lee, J., Park, K.: Gan-based imbalanced data intrusion detection system. Person. Ubiquitous Comput. **25**, 121–128 (2021)
58. Domingos, P.: Metacost: A general method for making classifiers cost-sensitive. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 155–164 (1999)
59. Menardi, G., Torelli, N.: Training and assessing classification rules with imbalanced data. Data Min. knowl. Discov. **28**, 92–122 (2014)
60. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. J. Artif. Intelli. Res. **16**, 321 (2002)
61. Zeng, M., Zou, B., Wei, F., Liu, X., Wang, L.: Effective prediction of three common diseases by combining smote with tomek links technique for imbalanced medical data. In: 2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS), pp. 225–228. IEEE (2016)
62. Choirunnisa, S., Lianto, J.: Hybrid method of undersampling and oversampling for handling imbalanced data. In: 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 276–280. IEEE (2018)
63. Li, X.-C., Zhan, D.-C., Shao, Y., Li, B., Song, S.: Fedphp: Federated personalization with inherited private models. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 587–602. Springer (2021)
64. Yu, T., Bagdasaryan, E., Shmatikov, V.: Salvaging federated learning by local adaptation. Preprint at arXiv:2002.04758 (2020)
65. Zhang, J., Li, C., Qi, J., He, J.: A survey on class imbalance in federated learning. Preprint at arXiv:2303.11673 (2023)
66. Konečǹy, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: strategies for improving communication efficiency. Preprint at arXiv:1610.05492 (2016)
67. Jiang, Z., Xu, Y., Xu, H., Wang, Z., Qiao, C., Zhao, Y.: Fedmp: Federated learning through adaptive model pruning in heterogeneous edge computing. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 767–779. IEEE (2022)
68. Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald, R., Muharemagic, E.: Deep learning applications and challenges in big data analytics. J. Big Data **2**(1), 1–21 (2015)
69. Cen, S., Zhang, H., Chi, Y., Chen, W., Liu, T.-Y.: Convergence of distributed stochastic variance reduced methods without sampling extra data. IEEE Trans. Signal Process. **68**, 3976–3989 (2020)
70. Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z.: On the convergence of fedavg on non-iid data. In: International Conference on Learning Representations (2020)
71. Haddadpour, F., Mahdavi, M.: On the convergence of local descent methods in federated learning. Preprint at arXiv:1910.14425 (2019)
72. Rostami, M., Kia, S.S.: Federated learning using variance reduced stochastic gradient for probabilistically activated agents. In: 2023 American Control Conference (ACC), pp. 861–866. IEEE (2023)
73. Xiao, C., Wang, S.: Triplets oversampling for class imbalanced federated datasets. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 368–383. Springer (2023)
74. Deng, L.: The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Process. Mag. **29**(6), 141–142 (2012)
75. Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N., Khazaeni, Y.: Bayesian nonparametric federated learning of neural networks. In: International Conference on Machine Learning, pp. 7252–7261. PMLR (2019)

76. Zhang, J., Li, A., Tang, M., Sun, J., Chen, X., Zhang, F., Chen, C., Chen, Y., Li, H.: Fed-cbs: A heterogeneity-aware client sampling mechanism for federated learning via class-imbalance reduction. In: International Conference on Machine Learning, pp. 41354–41381. PMLR (2023)
77. Terrail, J., Ayed, S.-S., Cyffers, E., Grimberg, F., He, C., Loeb, R., Mangold, P., Marchand, T., Marfoq, O., Mushtaq, E., et al.: Flamby: datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. Adv. Neural Inf. Process. Syst. **35**, 5315–5334 (2022)
78. Caldas, S., Duddu, S.M.K., Wu, P., Li, T., Konečnỳ, J., McMahan, H.B., Smith, V., Talwalkar, A.: Leaf: A benchmark for federated settings. Preprint at arXiv:1812.01097 (2018)
79. Cohen, G., Afshar, S., Tapson, J., Van Schaik, A.: Emnist: Extending mnist to handwritten letters. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 2921–2926. IEEE (2017)
80. Stolfo, S., Fan, W., Lee, W., Prodromidis, A., Chan, P.: KDD Cup 1999 data. https://archive.ics.uci.edu/dataset/130/kdd+cup+1999+data (1999)
81. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Preprint at arXiv:1409.1556 (2014)
82. Rastogi, V., Nath, S.: Differentially private aggregation of distributed time-series with transformation and encryption. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, pp. 735–746 (2010)
83. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. Intell. Data Anal. **6**(5), 429–449 (2002)
84. Zhan, Y., Li, P., Qu, Z., Zeng, D., Guo, S.: A learning-based incentive mechanism for federated learning. IEEE Internet Things J. **7**(7), 6360–6368 (2020)
85. Zhu, H., Zhang, H., Jin, Y.: From federated learning to federated neural architecture search: a survey. Complex Intell. Syst. **7**, 639–657 (2021)
86. Diao, E., Ding, J., Tarokh, V.: Heterofl: Computation and communication efficient federated learning for heterogeneous clients. Preprint at arXiv:2010.01264 (2020)
87. Lu, X., Liao, Y., Liu, C., Lio, P., Hui, P.: Heterogeneous model fusion federated learning mechanism based on model mapping. IEEE Internet Things J. **9**(8), 6058–6068 (2021)
88. Chen, M., Xu, Y., Xu, H., Huang, L.: Enhancing decentralized federated learning for non-iid data on heterogeneous devices. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE), pp. 2289–2302. IEEE (2023)
89. Gong, Y., Li, Y., Freris, N.M.: Fedadmm: A robust federated deep learning framework with adaptivity to system heterogeneity. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 2575–2587. IEEE (2022)
90. Wang, H., Xu, J.: Friends to help: Saving federated learning from client dropout. Preprint at arXiv:2205.13222 (2022)
91. Beutel, D.J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., Sani, L., Li, K.H., Parcollet, T., Gusmão, P.P.B., et al.: Flower: A friendly federated learning research framework. Preprint at arXiv:2007.14390 (2020)
92. Xu, R., Baracaldo, N., Zhou, Y., Anwar, A., Ludwig, H.: Hybridalpha: An efficient approach for privacy-preserving federated learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, pp. 13–23 (2019)
93. Li, B., Liu, Y., Wang, X.: Gradient harmonized single-stage detector. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 8577–8584 (2019)
94. Wang, L., Xu, Y., Xu, H., Liu, J., Wang, Z., Huang, L.: Enhancing federated learning with in-cloud unlabeled data. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 136–149. IEEE (2022)
95. Jeong, W., Yoon, J., Yang, E., Hwang, S.J.: Federated semi-supervised learning with inter-client consistency & disjoint learning. Preprint at arXiv:2006.12097 (2020)
96. Itahara, S., Nishio, T., Koda, Y., Morikura, M., Yamamoto, K.: Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. IEEE Trans. Mob. Comput. **22**(1), 191–205 (2021)
97. Mani, I., Zhang, I.: knn approach to unbalanced data distributions: a case study involving information extraction. In: Proceedings of Workshop on Learning from Imbalanced Datasets, vol. 126, pp. 1–7. ICML (2003)