CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	DP-MTFL: differentially private multi-tier federated learning for IoT applications
Authors	Soleimani, Ramin;Pesch, Dirk
Publication date	2024-07-24
Original Citation	Soleimani, R. and Pesch, D. (2024) 'DP-MTFL: differentially private multi-tier federated learning for iot applications', 2024 IEEE International Conference on Smart Computing (SMARTCOMP), Osaka, Japan, pp. 158–165. https://doi.org/10.1109/SMARTCOMP61445.2024.00042
Type of publication	Conference item
Link to publisher's version	https://ieeexplore.ieee.org/xpl/conhome/10595579/proceeding - 10.1109/SMARTCOMP61445.2024.00042
Rights	© 2024, IEEE. For the purpose of Open Access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission. - http://creativecommons.org/licenses/by/4.0/
Download date	2026-11-07 13:51:00
Item downloaded from	https://hdl.handle.net/10468/18269



UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

DP-MTFL: Differentially Private Multi-Tier Federated Learning for IoT applications

Ramin Soleimani
School of Computer Science and IT
University College Cork
Cork, Ireland
ramin.soleimani@cs.ucc.ie

Dirk Pesch
School of Computer Science and IT
University College Cork
Cork, Ireland
dirk.pesch@ucc.ie

***Abstract*—Differentially Private Federated Learning (DP-FL) is a privacy-preserving machine learning paradigm. Building on a standard DP-FL approach, we introduce and implement a novel Differentially Private Multi-Tier Federated Learning approach specifically tailored for IoT applications, specifically short-term load forecasting. Our method integrates a Sampled Gaussian Mechanism for differential privacy with a hierarchical federated learning approach, where local federations participate in learning while adhering to approximate differential privacy with respect to a global server. We specifically study the optimal number of local rounds on global model convergence. Our findings demonstrate that non-DP models with fewer local rounds exhibit slightly superior performance compared to DP-enabled models. However, integrating DP by introducing additional noise during training with larger local rounds enhances the generalization of global models, suggesting that the sampled Gaussian mechanism functions as a form of regularization. In the evaluation of our method, we utilise an energy consumption dataset from the UK Power Networks Low Carbon London project. Our results show that our approach achieves privacy preserving objectives while obtaining the optimal number of local rounds that minimise the prediction error.**

***Index Terms*—Federated Learning, Internet of Things, Differential Privacy, Sampled Gaussian Mechanism**

I. Introduction

IoT-enabled devices are used in a wide range of applications from smart homes and buildings to industrial automation systems to health and well-being applications. Many of these applications generate data that individuals and organisations wish to keep private. In this context, privacy and data breaches have become a significant concern [2]. Federated Learning (FL) is a decentralized machine learning approach that enables model training in real-time across multiple edge devices (such as IoT devices) without exchanging data gathered by these devices. A principal advantage of FL is the decoupling of model training from the need for direct access to all the training data [1]. IoT applications derive substantial benefits from the FL paradigm, notably in the realms of reduced communication overhead, edge intelligence, data diversity, privacy preservation, and security enhancement. FL represents a robust approach for privacy preservation when contrasted with conventional centralized model learning. Nevertheless, FL also has limitations within privacy-sensitive contexts. Attacks on neural networks [3] underscore the implicit risk and show the neural network's vulnerability to inference attacks. There are also recent works showing the possibility of information extraction on training data using only the trainable parameters of a model [9]–[11].

Differential privacy (DP) serves as a valuable supplementary measure to enhance the privacy robustness of federated learning. DP is a mathematical definition of privacy proposed by Dwork et al. [4]. DP offers verifiable safeguards against adversaries with diverse side information and computational capabilities. It enables precise measurement of privacy loss, ensuring a transparent assessment of data protection. Furthermore, DP exhibits a graceful composition property, proving effective even in scenarios involving multiple accesses to the same data. Over the last decade, significant efforts have been made to design fundamental algorithms and tools that satisfy the principles of DP and to integrate them with learning algorithms and statistical inference. Examples include the work of Abadi et al. [5], where authors developed new algorithmic techniques for learning and a refined analysis of privacy costs within the framework of differential privacy. McMahan et al. [6] and Wang et al. [7] added user-level privacy protection to a federated averaging algorithm. Mironov et al. [8] provide a privacy loss analysis and related numerical tools to search for possible solutions regarding noise scale and sample ratio for the sub-sampled Gaussian mechanism, which has recently become a well-established differential privacy mechanism integrating differential privacy with machine-learning algorithms.

Decentralization of FL through multi-tier FL, also known as hierarchical federated learning (HFL), has recently emerged to address several challenges facing the original two-layer client-server based FL [12]–[14]. Scalability, communication bottlenecks, network heterogeneity, privacy leakage, and a single point of failure are some of the problems of traditional two-layer FL, which can be addressed by alternative configurations or topologies of FL. However, implementing multi-tier FL with DP is not trivial. For example, the subsampled Gaussian mechanism, requires a large number of clients to yield a practical solution [6]. This requirement imposes limits

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6222 (ADVANCE CRT). For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

on designing multi-tier FL with DP, where local federations may not have enough users, which is a common problem of some mid-size IoT applications.

In this paper, we present DP-MTFL, a novel privacy-preserving scheme based on Differential Privacy (DP) tailored for Multi-Tier Federated Learning (MTFL). In our approach, we study varying the number of clients in local federations, with each federation being of equal size. Furthermore, our approach builds upon a refined privacy loss analysis of the Sampled Gaussian Mechanism introduced by Mironov et al. [8], where we compute differential privacy parameters. We allocate the Gaussian noise scale among local federations to ensure privacy guarantees for client datasets within the trained global model. In the combined approach, we study the impact of the number of local training rounds on global model performance, to identify the optimum number of local rounds to improve the performance of the globally trained model parameters. Reducing the number of local rounds significantly improves privacy as it results in sharing less information with the global server, limiting the auxiliary information, which can be the source of a differencing attack [16].

We evaluate our approach with the problem of Short-Term Load Forecasting (STLF). STLF is an important task in energy and power systems management. The goal of short-term load forecasting is to predict the future electrical load demand over a relatively short period, typically ranging from a few hours to a few days ahead. Accurate load forecasts are essential for efficient and reliable operation of power systems, helping utilities and grid operators make informed decisions in terms of energy production and demand-side management, especially when the grid is powered by a large share of renewable energy. STLF models rely on energy consumption data, which may contain sensitive information about individuals or businesses. Ensuring the privacy of this data is paramount to prevent misuse. Accessing users' energy consumption data for model training may constitute a privacy breach and fail to comply with the General Data Protection Regulation (GDPR) in Europe or the California Consumer Privacy Act (CCPA), which govern the collection, use, and protection of personal data. Additionally, the risk of linkage attacks, capable of matching "anonymized" records with non-anonymized records in different datasets, can undermine the effectiveness of anonymization techniques in safeguarding user data privacy.

Our choice of the dataset to train our models helped to gain practical insight into the training process in a real-world IoT application. The dataset was released including users' geographical clusters, which maps well to multi-tier FL, where edge servers (local servers) communicate with local smart meters. This avoids artificially clustering clients for local federations, which can be regarded as a privacy violation. Furthermore, utilizing a real-life dataset, which represents genuine clusters (here geographical clusters), significantly strengthens the credibility and robustness of our research outcomes. We conducted all evaluations presented in this paper using Colab Pro+, a cloud-based platform provided by Google that offers enhanced capabilities for machine learning tasks.

Our paper presents several noteworthy contributions to the field, chiefly a novel method that seamlessly incorporates the Sampled Gaussian Mechanism into a hierarchical federated learning framework. Through empirical analysis, we provide valuable insights into the impact of the optimal number of local training rounds on global model performance, convergence and generalization. A second contribution is our insights into the privacy advantages afforded by this approach within the context of approximate differential privacy. We discuss these contributions and their importance in advancing differentially private federated learning.

The paper is organised as follows: in Section 2, we provide a brief overview of federated learning and differential privacy. Subsequently, we introduce our research methodology, elucidating the mathematical rationale behind our proposed method. Section 3 presents the experimental setup, outlining the dataset and implementation details. Section 4 presents and discusses experimental results and we conclude in Section 5, where we summarize our key contributions.

II. Multi-tier federated learning with differential privacy

A. Multi-tier federated learning system

In a multi-tier federated learning system, three primary entities coexist, a global server, local servers, and clients. The training process unfolds through two stages, initialization and training iterations. In the initialization phase, the global server assumes the responsibility of initializing the model parameters and disseminating them across the entire system. During the training iteration, local servers conduct one or more rounds of federated learning. The global server is tasked with aggregating updates from local servers and sending them back to the respective local servers for distribution to clients. Through this iterative process, clients not only learn from other clients within their own local federation but also derive benefits from data present in other local federations.

B. Differential privacy paradiagm

We assume that the global server is less trusted within our multi-tier federated learning system. Consequently, we integrate Differential Privacy (DP) into the system to ensure client data remains differentially private, with respect to the global server and subsequently to other local servers updated by it. This approach enhances the protection of our system not only against attacks originating from the global server but also against potential threats from any other local servers.

We employ local rounds of training in the local federation to restrict the global server's access to auxiliary information. This limitation serves to mitigate the risk of inference attacks from the global server, as it is provided with less information. We consider the Sampled Gaussian Mechanism, a well-established DP mechanism in DP-enabled learning algorithms due to its advantageous composition properties. Sampled Gaussian Mechanism is an approximate DP mechanism, as there is always a possibility of privacy violation, which is proportional to the DP failure parameter δ.

C. Preliminaries

1) Federated Averaging (FedAvg): The Federated Averaging (FedAvg) algorithm [1] is a widely used approach in federated learning for aggregating model updates from different local models. It aims to collaboratively train a global model while keeping the raw data on local devices. We formulate one round of learning iteration in FedAvg as:

$$\theta^{t+1} \leftarrow \frac{\sum_{k \in S_t} n_k \theta_k^t}{m_t}$$

with

$$\begin{aligned} \theta^{t+1} &: \text{Updated server model at time } t+1 \\ S_t &: \text{Selected set of clients at time } t \\ n_k &: \text{Number of samples held by client } k \\ m_t &: m_t \leftarrow \sum_{k \in S_t} n_k \\ \theta_k^t &: \text{Local model of client } k \text{ at time } t \end{aligned}$$

2) Differential Privacy: Differential privacy (DP) is widely used for constraining and measuring the extent of privacy exposure during learning. DP thwarts an adversary's ability to confidently draw conclusions about whether a particular user's data was utilized in training a model, even given access to arbitrary side information. The formal definition of DP relies on the concept of neighbouring datasets: here, we designate a pair of datasets $d, d' \subset \mathcal{D}$ as neighbours if d' can be derived from d by adding or removing a single element:

Definition II.1 (Differential Privacy)**.** A (randomized) algorithm $M : D \to R$ with input domain D and output range R is (ε, δ)-differentially private if for all pairs of neighbouring datasets $d, d' \subset \mathbb{D}$, and every measurable $S \subseteq R$, we have

$$\Pr(M(d) \in S) \leq e^{\varepsilon} \cdot \Pr(M(d') \in S) + \delta$$

where probabilities are with respect to the coin flips of M.

3) Sampled Gaussian Mechanism: The Gaussian Mechanism (GM) is designed to satisfy the principles of differential privacy when applying a function over the privacy-sensitive dataset. Gaussian noise is incorporated into this mechanism by adding it to the output of the function. The noise is sampled from a Gaussian distribution with zero mean and a carefully chosen standard deviation σ. The Gaussian mechanism exhibits favourable composition properties. In the context of machine learning, where multiple iterations are involved, the ability to compose mechanisms without significantly degrading privacy is crucial.

Definition II.2 (Gaussian Mechanism)**.** Let $f : \mathbb{N}^{|X|} \to \mathbb{R}^d$ be an arbitrary d-dimensional function, and define its ℓ_2 sensitivity to be $\Delta_2 f = \max_{\text{adjacent } x,y} \|f(x) - f(y)\|_2$. The Gaussian Mechanism with parameter σ adds noise scaled to $\mathcal{N}(0, \sigma^2)$ to each of the d components of the output. The σ parameter in the Gaussian Mechanism is calibrated based on the sensitivity of the function(ℓ_2) being privatized and the desired level of privacy (ε, δ).

The Sampled Gaussian Mechanism (SGM) leverages the privacy amplification theorem [17], [23]. According to this theorem, if each step is (ε, δ)-differentially private with respect to the sampled set, it will be $(q\varepsilon, q\delta)$-differentially private with respect to the full database, where $q = \frac{L}{N}$ is the sampling ratio.

Definition II.3 (Sampled Gaussian Mechanism)**.** Let f be a function mapping subsets of S to $\mathbb{R}^d$. We define the Sampled Gaussian Mechanism parameterized with the sampling rate $0 < q \leq 1$ and the noise $\sigma > 0$ as

$$\text{SG}_q^{\sigma}(S)\Delta = f\left(\{x : x \in S \text{ with prob. } q\}\right) + \mathcal{N}\left(0, \sigma^2 I_d\right)$$

where each element of S is sampled independently at random with probability q without replacement, and $\mathcal{N}(0, \sigma^2 I_d)$ is spherical d-dimensional Gaussian noise with per-coordinate variance σ^2.

4) Differentially private federated averaging algorithm (DPFedAvg): The Sampled Gaussian Mechanism is widely recognized as a popular choice in the literature for integrating Differential Privacy (DP) with FedAvg, as discussed in works such as [5]–[8], [18]. Its popularity stems from its advantageous composition properties and its inherent ability to leverage the privacy amplification theorem. Let m be the number of sampled users in a round and Q_t be a set of the sampled users, then for iteration $t \in [T]$, the updated server model w^{t+1} at time $t+1$ is estimated as

$$w^{t+1} \leftarrow \frac{1}{m}\left(\sum_{k \in Q_t} w_k^t + \mathcal{N}(0, I\sigma^2)\right) \tag{1}$$

We utilize `dp_accounting.calibrate_dp_mechanism`, a class of the TensorFlow Privacy library, which is used to implement differential privacy mechanisms in machine learning models, to explore the optimal number of clients per round. The privacy estimation is facilitated by the `RdpAccountant` class, which leverages the `dp_accounting.DpEvent` class. These components draw on methodologies proposed by Wang et. al. [7] and Mironov et. al. [8] and are available in the `dp_accounting` library in Python.

D. Multi-tier federated learning with differential privacy

In our simulated multi-tier federated learning system, a global server holds the updated parameters at step i, denoted as θ^i. These global parameters are computed by averaging the parameters θ_j^i from m local servers, where j denotes the number assigned to each local server in our entire system. We have N clients (users) distributed across m local federations, with n_j indicating the number of clients in the j-th local federation. For T local updates in a client we compute the Sampled Gaussian Mechanism (SGM) parameters: sampling ratio q and noise scale σ^2 for N number of clients within the whole system and the approximate privacy parameters (ε, δ) using a refined version of the moments accountant algorithm

designed by Mironov et al. [8]. To satisfy the SGM principles the following equation (2) should hold in our system:

$$\tilde{\theta}^{i+1} = \frac{1}{Nq}(\sum_{i=1}^{m}\sum_{j=1}^{n_i}\theta_j^i + \mathcal{N}(0, I\sigma^2)) \tag{2}$$

where I is the identity matrix.

To implement DP at the local servers level in practice, we designed a system where local servers add noise proportional to the number of users in their possession. Additionally, each client participates in the local federation aggregation with probability q. In our method the global parameters at time $i+1$, (θ^{i+1}) is estimated as per equation (3):

$$\tilde{\theta}^{i+1} = \frac{1}{m}((\sum_{i=1}^{m}\frac{1}{n_i q}\sum_{j=1}^{n_i}\theta_j^i + \mathcal{N}(0, I\frac{n_i}{N}\sigma^2))) \tag{3}$$

we show below that under the constraint that n_i is an integer and equal to $\frac{N}{m}$ our method satisfies the SGM mechanism principles:

Proof.

$$\begin{aligned}\tilde{\theta}^{i+1} &= \frac{1}{m}\left(\sum_{i=1}^{m}\left(\frac{1}{n_i q}\sum_{j=1}^{n_i}\theta_j^i\right) + \sum_{i=1}^{m}\left(\frac{1}{n_i q}N(0, I\frac{n_i}{N}\sigma^2)\right)\right)\\ &= \frac{1}{Nq}\left(\sum_{i=1}^{m}\sum_{j=1}^{n_i}\theta_j^i + \sum_{i=1}^{m}N(0, I\frac{n_i}{N}\sigma^2)\right)\end{aligned}$$

Considering $\sum_{i=1}^{m} n_i = N$ and that the sum of m independent Normal random variables is normal distributed[1] [19], we have $\sum_{i=1}^{m} N(0, I\frac{n_i}{N}\sigma^2) = \mathcal{N}(0, I\sigma^2)$. This means equation (2) equals equation (3). □

III. Experimental Implementation

A. Dataset

To empirically validate our approach, we analyze the performance of our algorithm using a publicly available dataset [20]. This dataset comprises energy consumption readings from 5,567 London households, who participated in the UK Power Networks-led Low Carbon London project between November 2011 and February 2014. The dataset is particularly well-suited for our work, being derived from a real-world cluster of users. This characteristic makes it an ideal choice for evaluating a Multi-tier Federated Learning system.

We construct our dataset by considering 1,800 users clustered into 18 groups, based on the types of meters they use (Dynamic Time of Use - dToU or Static Time of Use - sToU), along with their ACORN[2] classification details [21]. Each group consists of 100 users, resulting in each local federation having 100 clients with the same type of meters and similar geo-demographic segmentation.

It is important to emphasize that we intentionally refrained from cleaning or pre-processing the dataset to remove outliers for example to remain close to real-life scenarios, where our algorithm would be implemented on the full dataset as available at local clients. This fidelity is crucial for accurately representing the challenges and complexities encountered in practical applications.

B. Experimental setup

1) Pre-processing: To mitigate the influence of outliers, we used standardization as a pre-processing technique in our baseline model pipeline. By standardizing all features to a consistent scale and minimizing the influence of outliers, these methods contribute to ensuring the model's effective acquisition of patterns and its ability to make precise predictions. Furthermore, standardization can lead to faster convergence during the training process, particularly when using optimization algorithms like gradient descent. It helps prevent issues such as vanishing or exploding gradients. We use the StandardScaler function, which is part of the scikit-learn library in Python.

$$x_{\text{standardized}} = \frac{x - \mu}{\sigma}$$

where:

$x_{\text{standardized}}$ is the standardized value of the feature.

x is the original value of the feature.

μ is the mean of the feature.

σ is the standard deviation of the feature.

2) Baseline model: In our federated learning approach for Short-Term Load Forecasting (STLF), the baseline model is designed by combining the strengths of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks called ConvLSTM. CNN layers excel in feature extraction from multidimensional data, making them suitable for early stage time-series forecasting tasks. LSTMs capture temporal dependencies and patterns over sequential data, making them useful in handling long-range dependencies, which is crucial for forecasting tasks where historical context plays a significant role. ConvLSTM combines the spatial pattern recognition capabilities of convolutional layers with the ability to capture long-term dependencies in sequential data provided by LSTM layers. This architecture is well-suited for STLF tasks where both spatial and temporal relationships in the input time-series data are important. To transfer the high-level features learned by previous network layers into the final output (forecast load), we add two dense layers at the

[1]If $X_1, X_2, .., X_n$ are mutually independent normal random variables with means $\mu_1, \mu_2, ..., \mu_n$ and variances $\sigma_1^2, \sigma_2^2, ..., \sigma_n^2$, then the linear combination:

$$Y = \sum_{i=1}^{n} c_i X_i$$

follows the normal distribution:

$$\mathcal{N}(\sum_{i=1}^{n} c_i\mu_i, \sum_{i=1}^{n} c_i^2\sigma_i^2)$$

[2]Acorn analyses data from hundreds of different sources and segments UK postcodes by common characteristics

end of our architecture. The choice of the neural network architecture shown in Figure 1, encompassing the number of layers and nodes, was meticulously fine-tuned through an iterative experimentation process. Various configurations were tested, and the final architecture was selected based on its superior performance observed during these experiments. The total number of trainable parameters in the final model is 234847. We implemented our deep learning baseline model using the `Keras` framework, an open-source high-level neural networks API written in Python.

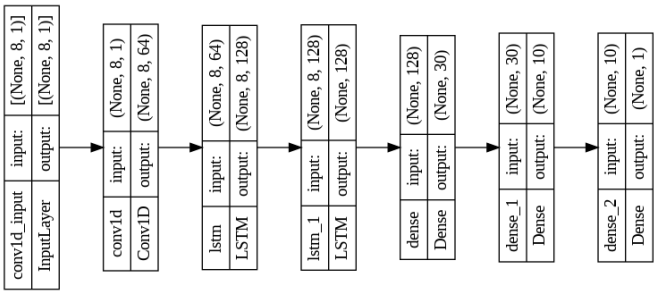


Fig. 1. Baseline Model

3) Hyperparameter tuning: We employed Bayesian optimization techniques to tune the hyperparameters of our baseline model. Bayesian optimization is a probabilistic model-based optimization method designed to discover the optimal set of hyperparameters for a given machine learning model. We constructed a probabilistic model of the objective function and leveraged it to propose new sets of hyperparameters for exploration. In our baseline model, we focused on tuning three key hyperparameters: the loss function (mean absolute error and mean squared error), the input data window size, and the learning rate. Bayesian Optimization played a crucial role in optimizing these hyperparameters to enhance the performance of our baseline model. We ended up with `WINDOW_SIZE = 8`, `learning_rate = 1e-7`, and mean squared error as our choice of the loss function.

4) Federations implementation: In our multi-tier federated learning simulation, we utilized TensorFlow Federated (TFF) [22]. The `tff.learning.algorithms.build_unweighted_fed_avg()` API was employed to construct our local federations for both differentially private (DP) and non-DP learning algorithms, leveraging the relevant TFF aggregation APIs. To implement a global federation (federation of federations), we designed our customized federated learning algorithm using low-level TFF APIs.

5) Privacy parameters tuning: Introducing noise typically diminishes the utility of the model. However, we have the ability to regulate the noise level in the average update for each round by adjusting two parameters: the standard deviation of the Gaussian noise added to the sum and the number of clients included in the averaging. Our approach involves initially determining the noise threshold that the model can withstand with a modest number of clients per round, maintaining an acceptable loss in model utility. Subsequently, during the training of the final model, we can augment the noise in the sum while simultaneously proportionally increasing the number of clients per round. In our final configuration, we set the noise multiplier to 0.07025 per local federation and 1.2645 for the entire system, with $q = 0.05$, achieving strong approximate differential privacy parameters of $(\varepsilon = 4, \delta = 2 \times 10^{-4})$ with respect to the global server.

6) Computational Platform: Our experiments were conducted on Google Colab Pro+, a cloud-based platform that provides a convenient and accessible environment for running machine learning tasks. We leveraged the GPU acceleration T4 provided by Google Colab. The total runtime of the various settings of our experiments exceeded 600 hours.

IV. Results and Discussion

We have previously emphasized the benefits of integrating local training rounds within the local federation, particularly in fortifying the privacy and security of clients in the presence of a potentially probing global server. This discussion is contextualized within the framework of approximately differentially private learning algorithms, which stand as one of the most widely embraced mechanisms for achieving differential privacy in the literature. Additionally, we hinted at the potential advantages of local rounds in addressing issues related to communication overhead and scalability.

Furthermore, our exploration delved into assessing the impact of local rounds on the overall performance and convergence of the global model across various experimental settings. Surprisingly, our findings revealed a considerable influence of the optimal number of local rounds on both the global model's performance and convergence.

To address sensitivity to scale and enhance robustness to outliers, we employed the Mean Absolute Percentage Error (MAPE) metric to assess the models. In the context of our study, we address a time-series prediction problem related to energy consumption. We partitioned the energy consumption records of each user, measured in kilowatt-hours (kWh) per half-hour, into distinct windows comprising 8 half-hour intervals—determined through our hyperparameter tuning approach. Each of these windows functions as a training instance, with the ensuing energy consumption value (per half-hour) serving as the corresponding label. We kept 10% of the data from each client aside to form an independent evaluation set, allowing a robust assessment of the model's generalization performance.

In our simulated differentially private (DP)-enabled multi-tier federated learning approach, we established 18 local federations, each consisting of 100 clients. Participants within each federation share the same electricity meter type (either `dToU` or `sToU`) and exhibit similar geo-demographic segmentation based on the ACORN[2] classification. The parameters, specifically a `noise_multiplier` of 1.2645 and $q = 0.05$, were derived from the computation of differential privacy parameters. This computation involved 400 iterations of training over end-user data, denoted by T, with $\varepsilon = 4$ and $\delta = 2 \times 10^{-4}$.

We conducted experiments for varying numbers of local rounds, denoted by k, and global rounds, denoted by g. Since we computed the differential privacy parameters for $T = 400$ iterations of training over clients' data, it is essential to note that the equation $T = k \cdot g$ should hold, in other words, g and k must be factors of T.

To demonstrate a compelling incentive for local federations to participate in the multi-tier federation, we trained a model using clients' data from a single cluster (block0) named 'single federation.' The model was trained with a `noise_multiplier` of 0.07025 per local federation in the multi-tier federated learning setting), and $q = 0.05$. Subsequently, we compared its performance with global models trained within the multi-tier federation system and presented the performance scores in the result tables. In our evaluation scheme, we randomly selected 20 users from local federations (blocks) and utilized their held-out datasets to evaluate the performance of models trained with different settings of the number of local and global rounds, as demonstrated in Table I. We computed the average, minimum, and maximum of the MAPE scores of the models for 20 randomly chosen clients in block0 and block17. As shown in Table I, models trained within multi-tier federated learning settings outperform the single federation model, which was trained only by a federation of clients in block0. It means that local federations have enough incentive to take part in the multi-tier federation and learn from the data distributions of other federations.

As evident from the result tables, the empirical optimal values for this application, considering the number of local rounds and global rounds, are $k = 4$ and $g = 100$. With these values, the globally trained model outperforms other global models trained under different pairs of local and global rounds. Our results underscore the significant role these values play in terms of global model convergence and performance. Another noteworthy observation in our results is the impact of the number of local rounds on the model's performance, particularly on data that is challenging to predict. For instance, in Tables I and II (see Max MAPE scores column) the global model trained with $k = 4$ and $g = 100$ (optimal empirical values) outperformed other models significantly. The difference with the second-best performer regarding the worst client to predict in Table I (block0) is 2.69, and in Table II (block17) is 40.59.

In terms of privacy enhancement, our proposed differentially private multi-tier federated learning method demonstrates significant potential to enhance client privacy. Specifically, identifying the optimal observational local rounds of training, here $k = 4$, not only improves global model accuracy but also substantially reduces information sharing with a less trusted global server by a factor of 4, enhancing approximate differential privacy.

The results presented in Table I and Table II indicate that even with an increase in local rounds, the accuracy is not dramatically compromised. Given this observation and considering the tradeoff between increased privacy/security and model accuracy, it is advisable to cautiously explore higher values for local rounds, as suggested by our findings.

TABLE I
COMPARISON OF MAPE SCORES ON HELD-OUT DATA FROM BLOCK 0 USERS FOR GLOBAL MODELS TRAINED WITH DIFFERENT CONFIGURATIONS, CONSIDERING VARYING NUMBERS OF LOCAL AND GLOBAL ROUNDS

Number of g and k	Block0\20 random users		
	Average MAPE	Min MAPE	Max MAPE
$k = 1$ $g = 400$	36.67	20.03	56.52
$k = 2$ $g = 200$	35.72	18.96	53.35
$k = 4$ $g = 100$	35.29	18.59	50.66
$k = 8$ $g = 50$	36.11	19.26	55.23
$k = 16$ $g = 25$	35.78	19.25	53.51
$k = 25$ $g = 16$	36.06	19.74	54.05
$k = 50$ $g = 8$	39.48	21.57	62.74
Single Federation	41.83	21.57	76.26

TABLE II
COMPARISON OF MAPE SCORES ON HELD-OUT DATA FROM BLOCK 20 USERS FOR GLOBAL MODELS TRAINED WITH DIFFERENT CONFIGURATIONS, CONSIDERING VARYING NUMBERS OF LOCAL AND GLOBAL ROUNDS

Number of g and k	Block17\20 random users		
	Average MAPE	Min MAPE	Max MAPE
$k = 1$ $g = 400$	97.20	23.90	308.20
$k = 2$ $g = 200$	87.55	21.27	275.21
$k = 4$ $g = 100$	71.57	20.73	206.40
$k = 8$ $g = 50$	90.18	21.95	280.44
$k = 16$ $g = 25$	84.95	21.76	271.29
$k = 25$ $g = 16$	81.80	22.52	246.99
$k = 50$ $g = 8$	100.78	24.58	311.26
Single Federation	118.83	33.86	353.20

To ensure a representative evaluation, we carefully selected a test set by randomly choosing 20% of users from each of the 18 local federations. These users had previously contributed their trained datasets, and now, their held-out datasets were utilized for evaluation purposes. This meticulous approach adheres to the principle of Independent and Identically Distributed (IID) data, reinforcing the reliability of our evaluation results. Figure 2 visually represents the computed evaluation metric (MAPE) based on this carefully constructed test set for our trained models. As the results show, the global model trained with $k = 4$ and $g = 100$ (optimal empirical values) outperforms other models on the comprehensive test set data.

Figure 3 illustrates the convergence of the global models based on the local rounds for various configurations of the numbers of local and global rounds. The plot indicates that a larger number of local rounds may result in slower and more

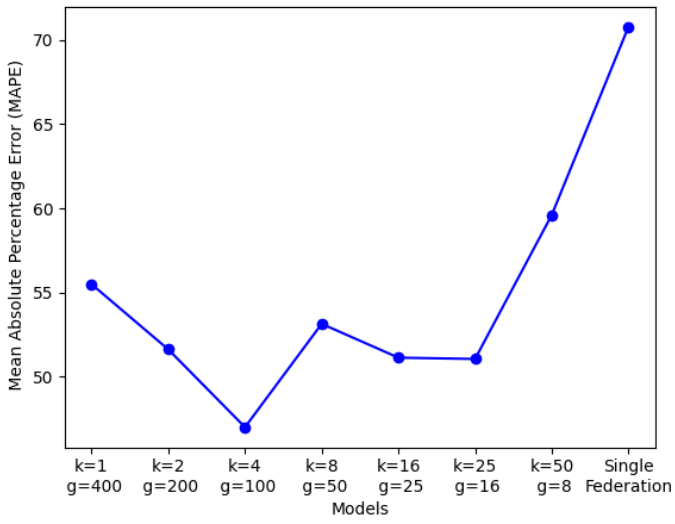


Fig. 2. MAPE evaluation results for our trained models on the test set. The global model with $k = 4$ and $g = 100$ outperforms others

unstable model convergence in our use case.

Figure 4 visualizes the global model performance for $k = 4$ and $g = 100$ on the held-out data of a client indexed 90 from block0. To assess the impact of Gaussian noise on the global model's performance, we conducted experiments within our multi-tier FL scheme, comparing models trained without noise but with the same sampling ratio $q = 0.05$ to those trained with noise (deferentially enabled models). Table III presents a comparison of the performance between DP-enabled and non-DP models on users' held-out data from block 0. As observed in the table, non-DP models with fewer local rounds exhibit slightly better prediction performance compared to DP-enabled models. However, in our specific use case, the results suggest that the additional noise introduced during training contributes to better generalization of unseen data, particularly for a larger number of rounds. It appears that the randomization introduced by the Sampled Gaussian Mechanism (SGM) in settings with larger local rounds acts as a regularization technique, facilitating improved generalization for the global model. This finding underscores the need for further exploration in future research endeavours. It is noteworthy that during privacy parameter tuning, we aimed to introduce noise to the models up to their tolerable limits.

V. Conclusion

In this paper, we introduced a novel Differentially Private Multi-Tier Federated Learning (DP-MTFL) approach suitable for IoT applications. Our methodology integrates the Sampled Gaussian Mechanism, an approximate differential privacy mechanism, into hierarchical federated learning. We presented empirical insights into the role played by the optimal number of observational local rounds in shaping global model convergence, generalization, and, notably, its privacy implications within the context of approximate differential privacy.

TABLE III
Comparison of MAPE Scores on Held-Out Data from Block 0 Users for Global Models Trained with and without Differential Privacy (DP) for Varying Numbers of Local and Global Rounds

Number of g and k	Block0\20 random users	
	Average MAPE DP enabled	Average MAPE None DP
$k = 1$ $g = 400$	36.67	36.93
$k = 2$ $g = 200$	35.72	35.23
$k = 4$ $g = 100$	35.29	36.46
$k = 8$ $g = 50$	36.11	36.31
$k = 16$ $g = 25$	35.78	35.03
$k = 25$ $g = 16$	36.06	72.18
$k = 50$ $g = 8$	39.48	56.02

Our approach addresses privacy concerns for user data kept within local servers for small and medium-sized user populations, which may not meet practical requirements for the approximate Sampled Gaussian Mechanism in terms of the number of users. By enabling local servers to engage in learning with other local federations while adhering to approximate differential privacy concerning the global server, our proposed method mitigates privacy risks associated with the failure parameter delta within approximate differential privacy. Additionally, it boosts global model performance by assessing the empirical optimum number of local rounds.

The utilization of a real-life dataset with authentic clusters (groups of properties in the same area) enhances the robustness of our results. Notably, we refrained from employing clustering methods, addressing potential privacy concerns in real-life scenarios. Our research contributes to advancing privacy-preserving federated learning methods, particularly in the realm of IoT applications, and offers valuable insights for future developments in this domain.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Aguera y Arcas,"Communication-Efficient Learning of Deep Networks from Decentralized Data", Proceedings of the 20th International Conference on Artificial Intelligence and Statistics PMLR 54, pp. 1273-1282, 2017.

[2] L. Tawalbeh, F. Muheidat, M. Tawalbeh, and M. Quwaider, "IoT Privacy and Security: Challenges and Solutions," Appl. Sci., vol. 10, no. 12, p. 4102, 2020. DOI: 10.3390/app10124102.

[3] R. Shokri, M. Stronati, C. Song and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2017, pp. 3-18, doi: 10.1109/SP.2017.41.

[4] Dwork, C. (2006). Differential Privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds) Automata, Languages and Programming. ICALP 2006. Lecture Notes in Computer Science, vol 4052. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11787006_1

[5] M. Abadi, A. Chu, I. Goodfellow, H. Brendan McMahan, I. Mironov, K. Talwar, L. Zhang. "Deep Learning with Differential Privacy", Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16). Association for Computing Machinery, New York, NY, USA, pp. 308–318. 2016. https://doi.org/10.1145/2976749.2978318

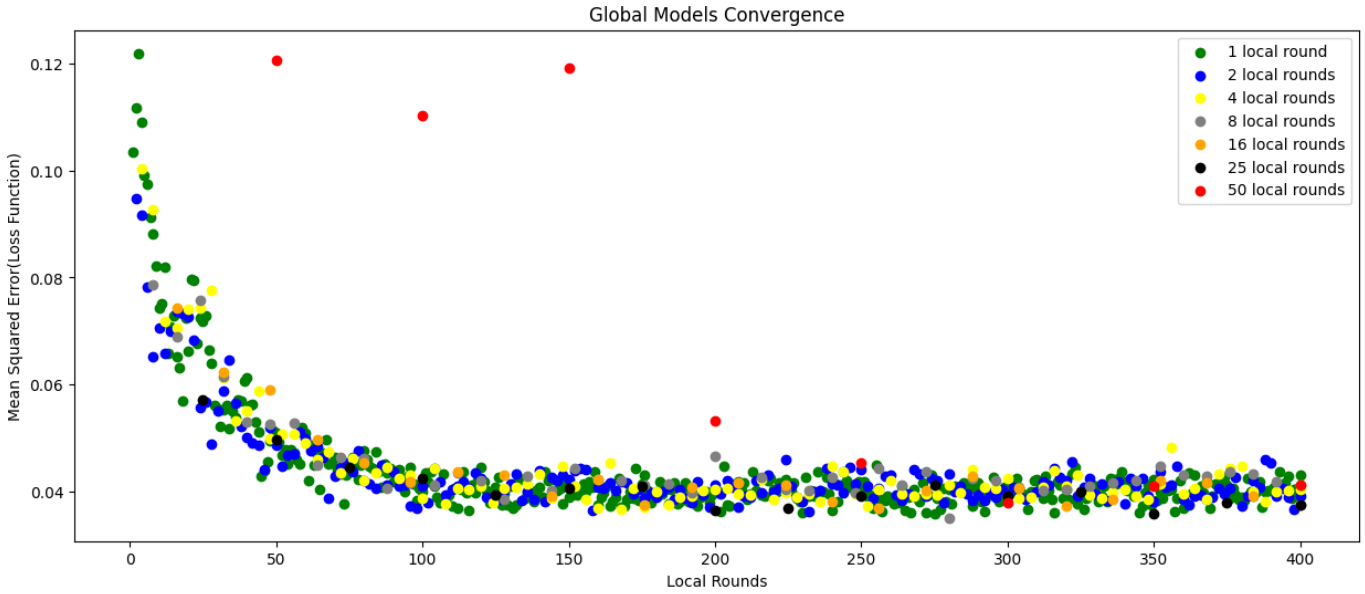


Fig. 3. Global model convergence based on local rounds across different configurations of local and global rounds

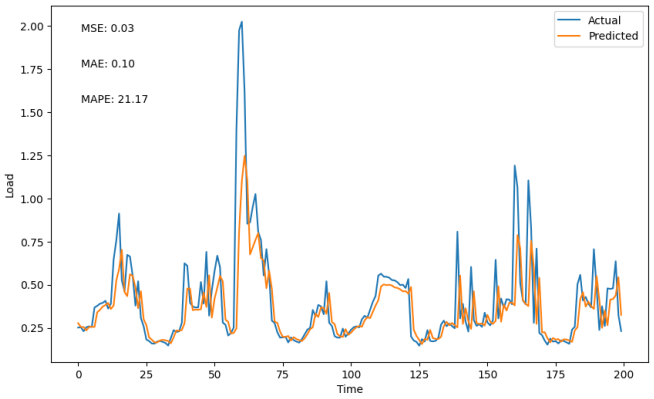


Fig. 4. Performance of the global model with $k = 4$ and $g = 100$ on held-out data from client 90 in block 0

[6] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning Differentially Private Recurrent Language Models," openreview.net, Feb. 24, 2018. https://openreview.net/forum?id=BJ0hF1Z0b

[7] Y.-X. Wang, B. Balle, and S. P. Kasiviswanathan, "Subsampled Renyi Differential Privacy and Analytical Moments Accountant," proceedings.mlr.press, Apr. 11, 2019. https://proceedings.mlr.press/v89/wang19b.html (accessed Jan. 29, 2024).

[8] I. Mironov, K. Talwar, and L. Zhang, "Rényi Differential Privacy of the Sampled Gaussian Mechanism," arXiv.org, Aug. 27, 2019. https://arxiv.org/abs/1908.10530.

[9] M. Fredrikson, S. Jha, and T. Ristenpart, "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures," Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15, 2015, doi: https://doi.org/10.1145/2810103.2813677.

[10] X. Wu, M. Fredrikson, S. Jha and J. F. Naughton, "A Methodology for Formalizing Model-Inversion Attacks," 2016 IEEE 29th Computer Security Foundations Symposium (CSF), Lisbon, Portugal, 2016, pp. 355-370, doi: 10.1109/CSF.2016.32.

[11] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks," arXiv:1802.08232 [cs], Jul. 2019, Available: https://arxiv.org/abs/1802.08232

[12] A. Wainakh, A. S. Guinea, T. Grube and M. Mühlhäuser,"Enhancing Privacy via Hierarchical Federated Learning," 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS and PW), Genoa, Italy, 2020, pp. 344-347, doi: 10.1109 / EuroSPW51379.2020.00053.

[13] M. S. H. Abad, E. Ozfatura, D. GUndUz and O. Ercetin, "Hierarchical Federated Learning ACROSS Heterogeneous Cellular Networks," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 8866-8870, doi: 10.1109/ICASSP40776.2020.9054634.

[14] L. Shi, J. Shu, W. Zhang and Y. Liu, "HFL-DP: Hierarchical Federated Learning with Differential Privacy," 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 2021, pp. 1-7, doi: 10.1109/GLOBECOM46510.2021.9685644.

[15] W. Y. B. Lim, J. S. Ng,Z. Xiong,J. Jin,Y. Zhang, D. Niyato,C. Leung, "Decentralized Edge Intelligence: A Dynamic Resource Allocation Framework for Hierarchical Federated Learning," in IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 3, pp. 536-550, 1 March 2022, doi: 10.1109/TPDS.2021.3096076.

[16] C. Ma et al., "On safeguarding privacy and security in the framework of federated learning", IEEE network, vol. 34, no. 4, pp. 242-248, 2020.

[17] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What Can We Learn Privately?," arXiv.org, Feb. 18, 2010. https://arxiv.org/abs/0803.0924

[18] G. Andrew, O. Thakkar, H. Mcmahan, and S. Ramaswamy, "Differentially Private Learning with Adaptive Clipping." Accessed: Feb. 11, 2024. [Online]. Available: https://arxiv.org/pdf/1905.03871.pdf

[19] "26.1 - Sums of Independent Normal Random Variables — STAT 414," PennState: Statistics Online Courses. https://online.stat.psu.edu/stat414/lesson/26/26.1

[20] "Smart meters in London," www.kaggle.com. https://www.kaggle.com/datasets/jeanmidev/smart-meters-in-london

[21] Acorn, "Acorn - The smarter consumer classification — CACI," Caci.co.uk, 2023. https://acorn.caci.co.uk/

[22] "TensorFlow Federated," TensorFlow. https://www.tensorflow.org/federated

[23] A. Beimel, H. Brenner, S. P. Kasiviswanathan, and K. Nissim, "Bounds on the sample complexity for private learning and private data release," Machine Learning, vol. 94, no. 3, pp. 401–437, Sep. 2013, doi: https://doi.org/10.1007/s10994-013-5404-1