PrivateFL: Accurate, Differentially Private Federated Learning via Personalized Data Transformation

Yuchen Yang, Bo Hui, and Haolin Yuan, *The Johns Hopkins University;*
Neil Gong, *Duke University;* Yinzhi Cao, *The Johns Hopkins University*

https://www.usenix.org/conference/usenixsecurity23/presentation/yang-yuchen

This artifact appendix is included in the Artifact Appendices to the Proceedings of the 32nd USENIX Security Symposium and appends to the paper of the same name that appears in the Proceedings of the 32nd USENIX Security Symposium.

August 9–11, 2023 • Anaheim, CA, USA

978-1-939133-37-3

USENIX'23 Artifact Appendix: PRIVATEFL: Accurate, Differentially Private Federated Learning via Personalized Data Transformation

Yuchen Yang*, Bo Hui*, Haolin Yuan*, Neil Gong†, and Yinzhi Cao
The Johns Hopkins University, †Duke University

A Artifact Appendix

A.1 Abstract

This artifact includes the source code, dataset, setup, and instructions to reproduce the results of PRIVATEFL reported in Section 6, i.e., the evaluation section. This artifact supports our claim that PRIVATEFL can improve the accuracy when applied to FL with DP, and can further improve the accuracy as an add-on to the existing DP-improving method. Our artifacts are available at this open-source repository (https://github.com/BHui97/PrivateFL). Our artifacts require a Linux machine with 64GB of RAM and a GPU with 24 GB of graphics memory.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

None.

A.2.2 How to access

We provide the following accesse to our artifacts:

- GitHub repository: https://github.com/BHui97/PrivateFL. You can clone the source code from the main branches and set up the environment following the instruction in README.md file.

A.2.3 Hardware dependencies

We recommend using a 64bit Linux machine with the following requirements:

- GPU: 24G. We test the artifacts on NVIDIA GeForce RTX 3090
- RAM: 64G
- Storage: 16GB

A.2.4 Software dependencies

- Ubuntu 18.04
- Python 3.8+ and Python libraries listed in requirements.txt
- NVIDIA Driver and CUDA for GPU computation. We test the artifacts on NVIDIA Driver Version 510.108.03 and CUDA Version 11.6

A.2.5 Benchmarks

We list the datasets and models the experiments require with this artifact reported in our paper.

- Datasets:
  - CIFAR-10, MNIST, FashionMNIST, CIFAR-100, EMNIST: we use the library `torchvision.datasets`. Details can be found in `dataset.py` of our GitHub repo.
  - CH-MNIST: Dataset can be downloaded from https://github.com/BHui97/PrivateFL/tree/main/data/CHMNIST
  - Purchase: Dataset can be download from https://github.com/BHui97/PrivateFL/tree/main/data/purchase, please unzip the file before using.
- Models:
  - AlexNet, ResNet, 3-layer DNN, 4-layer DNN: details can be found in `modelUtil.py` of our GitHub repo.
  - CLIP: we use pretrained weights from the library `CLIP`, details can be found in `transfer/extract_cifar.py`.
  - SimCLR: model can be downloaded via https://pl-bolts-weights.s3.us-east-2.amazonaws.com/simclr/bolts_simclr_imagenet/simclr_imagenet.ckpt

A.3 Set-up

A.3.1 Installation

Source code. Start with the source code and set up the environment with the `README.md`, first install the source and enter it:

- `git clone https://github.com/BHui97/PrivateFL.git`
- `cd PrivateFL/script`

```
Client: 0 ACC: 0.9017160534858704, episilon: 1.9973331713299505
Client: 1 ACC: 0.8433049321174622, episilon: 1.9932507283841159
Round: 1
Acc: 0.5007001400280056
Epsilon: 1.9973331713299505
Use time: 43.09s
==> Test Finished!
```

Figure 1: Expected output of the basic test.

Conda (optional). This step is optional if you don't have an environment/package management tool installed. We use Miniconda to create a virtual environment with Python 3.8. You can install it using the following script for a Ubuntu 18.04 machine or refer to the official document[1]:

- `bash install_conda.sh`
- Note: remember to CLOSE and then RE-OPEN your terminal after running the above script.

Python dependencies. Then run the following script to install the required Python dependencies:

- `bash setup.sh`

Datasets. Datasets except Purchase will be automatically downloaded, we provide the Purchase dataset under `data/purchase/dataset_purchase.zip`. Please remember to unzip it before running the test.

Models. Models except ResNext will be automatically downloaded. Please download the ResNext weight `model_best.pth.tar` manually from this link, and put it under the `transfer/model/` folder.

A.3.2 Basic Test

Run the following script to verify the installation and test the functionality:

- `func_test_1.sh`

The expected successful output follows the structure shown in Figure 1. The specific number may be different for different tests.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): PRIVATEFL *can improve the accuracy for FL with DP. This is proven by the experiments (E1), whose results are reported in Section 6.1 (Figure 5) of our paper.*

[1] https://docs.conda.io/projects/conda/en/latest/user-guide/install/linux.html

(C2): PRIVATEFL *can further improve the accuracy for FL with DP as an add-on to the existing DP-improving method. This is proven by the experiments (E2), whose results are reported in Section 6.2 (Table 6) of our paper.*

(C3): PRIVATEFL *consistently improves accuracy when the clients' local training data has different heterogeneity. This is proven by the experiments (E3), whose results are reported in Section 6.4 (Table 9) of our paper.*

(C4): PRIVATEFL *consistently improves the accuracy of FL under DP when the system has a different number of clients. This is proven by the experiments (E4), whose results are reported in Section 6.5 (Table 10) of our paper.*

A.4.2 Experiments

(E1): *[PrivateFL with different epsilon] [Six datasets] [20 human-minutes + 30 compute-hours (from 0.5 to 12 compute-hours for different datasets)]: This experiment tests the accuracy of PrivateFL with different epsilon, i.e., 2 to 8, on six datasets. Each of the datasets can be tested by running `bash script/E1_[dataset].sh`*

How to: The accuracy evaluation on each dataset will be tested via a data-specific script, and the evaluated accuracy for different epsilons will be automatically summarised and shown at the end of the script execution. You can collect the accuracies for each dataset and compare them with Figure 5 in our paper.

Preparation: Follow Section A.3 to finish the setup. Remember to unzip the purchase dataset before running the script for purchase.

Execution: First navigate to the script folder via `cd script`, then run `bash E1_[dataset].sh`. Please replace the `[dataset]` with one of [mnist, fashion-mnist, emnist, purchase, cifar10, chmnist], e.g, `bash E1_mnist.sh`. Note that cifar10 and chmnist need 12+ hours of training due to the large model size, which may vary from different GPUs. The other datasets could be finished within 1 hour. If you encounter CUDA out of memory, please reduce the value `-physical_bs` in `bash E1_[dataset].sh`.

Results: The results are shown as a table with four columns named `[data, mode, epsilon, accuracy]`. The `data` column shows the current evaluated dataset name; the `mode` column shows different DP methods, e.g, LDP; the `epsilon` column shows different epsilon that is being evaluated, e.g., 2; the `accuracy` column shows the testing accuracy for the combination of previous three columns, e.g., `accuracy = 0.946` for `[data = emnist, mode = CDP, epsilon = 2]`.

(E2): *[PrivateFL + DP-improvement with different epsilon] [Two datasets] [10 human-minutes + 2 compute-hours]: This experiment tests the accuracy of Pri-*

vateFL combined with existing DP-improvement methods. We test different epsilon, i.e., 2 to 8, on two datasets, with three different pretrained encoders. Each of the datasets can be tested by running `bash script/E2_[dataset].sh`

How to: The accuracy evaluation on each dataset will be tested via a data-specific script, and the evaluated accuracy for different epsilons and pretrained encoders will be automatically summarised and shown at the end of the script execution. You can collect the accuracies for each dataset and compare them with Table 5 in our paper.

Preparation: Follow Section A.3 to finish the setup. Remember to download the ResNext model following Section A.3.1 before running the script. To save time, we have uploaded the extracted features from different encoders under folder `transfer/feature`. If you want to extract it yourself, delete all folders, e.g., folder named `cifar10_2cpc_100client_clip_64`, under `transfer/feature`.

Execution: First navigate to the script folder via `cd script`, then run `bash E2_[dataset].sh`. Please replace the `[dataset]` with one of [cifar10, cifar100], e.g, `bash E2_cifar10.sh`

Results: The results are shown as a table with four columns named `[data, mode, model, epsilon, accuracy]`. The `model` column shows the pretrained encoder used to extract the feature, e.g., clip; the `data, mode, epsilon` columns are similar to E1; the `accuracy` column shows the testing accuracy for the combination of the previous four columns, e.g., `accuracy = 0.594` for `[data = cifar100, mode = CDP, model = clip, epsilon = 2]`.

(E3): *[PrivateFL with the different data heterogeneity] [Two datasets] [10 human-minutes + 2 compute-hours]: This experiment tests the accuracy of PrivateFL with different data heterogeneity, i.e., 2 to 10 classes per client, on two datasets. Each of the datasets can be tested by running* `bash script/E3_[dataset].sh`

How to: The accuracy evaluation on each dataset will be tested via a data-specific script, and the evaluated accuracy for different data heterogeneity will be automatically summarised and shown at the end of the script execution. You can collect the accuracies for each dataset and compare them with Table 9 in our paper.

Preparation: Follow Section A.3 to finish the setup.

Execution: First navigate to the script folder via `cd script`, then run `bash E3_[dataset].sh`. Please replace the `[dataset]` with one of [mnist, cifar10], e.g, `bash E3_mnist.sh`

Results: The results are shown as a table with four columns named `[data, mode, ncpc, accuracy]`. The `data, mode` columns are similar to E1; the `ncpc` is the number of classes assigned to each client, i.e., 2 (non-iid) to 10 (iid); the `accuracy` column shows the testing accuracy for the combination of the previous three columns, e.g., `accuracy = 0.922` for `[data = mnist, mode = CDP, ncpc = 2]`.

(E4): *[Private FL with the different number of clients] [Two datasets] [10 human-minutes + 5 compute-hours]: This experiment tests the accuracy of PrivateFL with the different number of clients, i.e., 50 to 500, on two datasets. Each of the datasets can be tested by running* `bash script/E4_[dataset].sh`

How to: The accuracy evaluation on each dataset will be tested via a data-specific script, and the evaluated accuracy for different numbers of clients will be automatically summarised and shown at the end of the script execution. You can collect the accuracies for each dataset and compare them with Table 10 in our paper.

Preparation: Follow Section A.3 to finish the setup.

Execution: First navigate to the script folder via `cd script`, then run `bash E4_[dataset].sh`. Please replace the `[dataset]` with one of [mnist, cifar10], e.g, `bash E4_mnist.sh`

Results: The results are shown as a table with four columns named `[data, mode, nc, accuracy]`. The `data, mode` columns are similar to E1; the `nc` is the number of clients, i.e., 50 to 500; the `accuracy` column shows the testing accuracy for the combination of the previous three columns, e.g., `accuracy = 0.892` for `[data = mnist, mode = LDP, nc = 50]`.

A.5 Version

Based on the LaTeX template for Artifact Evaluation V20220926. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2023/.