Instructions for Paper Artifacts

1 OVERVIEW

We use the jupyter notebook to demo the codes and show the attack results on our pretrained models.

The folder contains three separate jupyter files for different datasets.

- **NeuGuard_ALEXNET_cifar10.ipynb** for CIFAR10 dataset
- **NeuGuard_ALEXNET_cifar100.ipynb** for CIFAR100 dataset
- **NeuGuard_Texas100.ipynb** for Texas100 dataset.

2 REQUIREMENTS

CUDA Enabled GPU hardware

- python == 3.7.10
- pytorch == 1.9.0
- cudatoolkit == 10.2

We provide the **environment.yml** file for our running environment.

You can create the environment using the following command.

conda env create -f environment.yml

Detailed instructions please check the conda website.

We specify the running environment here, but other versions of python and pytorch might also work.

3 DATASETS

To run the attack evaluation, you need to download the three datasets first.

For the Texas100 dataset, the default way to load the data takes long time, we save a npz file to speed up the loading. We provide both ways to load the data.

- Name: texas100_data.npz download here. (https://drive.google.com/file/d/1G9-oWyLqiSTDuB2ku6xYY7MVWOur6OOA/view?usp=sharing).
- We load Texas100 data with a randomized order following the file **random_r_texas100_prune**. Please download it before running the code. Using python 2 requires changes to the code that loads this file, and we provide the corresponding code for replacement.

For CIFAR10 and CIFAR100 datasets, they will download automatically if you don't have them.

4 EVALUATION STEPS

For the convenience of running the experiments, we provide the code in the jupyter notebook.

Once you open the jupyter file, you can search for the corresponding cell using the **# keywords** given in each step and run the cells following the instructions.

For each code in the jupyter notebook, we have similar running steps.

(1) Run all the cells above the **# start train** cell for initialization.
(2) Load model in the **# load saved model** cell and the following two cells to check the model accuracy.
(3) Load unsort NSH attack model in the **# load membership inference attack** cell and the following five cells to check the model accuracy. This step runs the unsorted neural network based membership inference attack corresponding to Table 4 in Section 6.1.
(4) Load sort NN attack model in the **# load NN attack model** cell and the following two cells to check the model accuracy. This step runs the sorted neural network based membership inference attack corresponding to Table 4 in Section 6.1.
(5) Run cell in **# load for metric base attack** and following cells to perform the metric based attack. This step run four metric based membership inference attacks corresponding to Table 6 in Section 7.
(6) Run cell in **# load for c&w label-only attack** and following cells to perform the c&w label-only attack. This attack corresponds to Table 8 in Section 7 for CIFAR10 and CIFAR100.

In the code, we predefined the load models with names that correspond to the specific pretrained models provided for inference and performing the attack. We also provide the training code to train our defensive model and perform attacks. You can modify the parameters and settings to train your model.

5 PRETRAINED MODELS

We provide pretrianed models used for the work. Specifically, we upload the models trained using the proposed NeuGuard method for all three datasets, and we include both sorted and unsorted NN based attack models, respectively.

The pretrained models can be downloaded here. (https://drive.google.com/drive/folders/1qjPOpicHpCoKcdmL2Iko5f7P6ho5MrIq?usp=sharing)