ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

A Artifact Appendix

A.1 Abstract

Obligatory. Briefly describe your artifact including minimal hardware and software requirements, how it supports your paper, how it can be validated, and what is the expected result. At submission time, it will also be used to select appropriate reviewers. It will also help readers understand what was evaluated and how.

This artifact includes the source code for the experiments in the paper. The artifact is built upon Python and its libraries (e.g., Pytorch) and requires the access to GPUs for accelerating the model training. The required Python libraries are listed in the source code. The artifact is tested on Linux with NVIDIA V100 GPUs. The artifact will validate the attack performance observed in the paper. By running the code, the artifact will output original models, pruned models, and print out the results (i.e., attack accuracy) of membership inference attacks and defenses on the models.

A.2 Artifact check-list (meta-information)

Obligatory. Fill in whatever is applicable with some keywords and remove unrelated items.

- **Algorithm:** The proposed MIA attack and defense is proposed and included in the source code.
- **Model:** The ResNet18, DenseNet121, VGG16, FC models are included.
- **Data set:** The access to the CIFAR10, CIFAR100, CHMNIST, SVHN, Location, Texas, Purchase datasets is included.
- **Hardware:** GPU is required to accelerate model training.
- **Metrics:** The prediction accuracy and attack accuracy are reported.
- **Output:** The model prediction accuracy and attack accuracy will be output.
- **Experiments:** The guide to reproduce the experiments is provided in README file.
- **How much disk space required (approximately)?:** For each dataset and neural network architecture, we need around 10GB-100GB disk space to store original models, pruned models, pruned models with defense, and the corresponding shadow models. To run all the experiments, around 2TB disk space is required to store all the models. To reduce the disk space requirement, we can delete the models that have been evaluated, since the models trained on different datasets and neural network architectures are independent.
- **How much time is needed to prepare workflow (approximately)?:** Less than 1 hour is needed to install all the Python libraries.
- **How much time is needed to complete experiments (approximately)?:** It takes around 2-3 hours to evaluate the attacks and defenses on a single experimental setting using an NVIDIA V100 GPU. The entire experiment settings include 7 datasets, 4 neural network architectures, 4 pruning approaches, and 5 sparsity levels, in total 255 pruned models.
- **Publicly available (explicitly provide evolving version reference)?:** The code is available at github.com/Machine-Learning-Security-Lab/mia_prune.
- **Code licenses (if publicly available)?:** The code is under MIT License.
- **Data licenses (if publicly available)?:** All datasets are publicly available.

A.3 Description

Obligatory. For inapplicable subsections (e.g., the "How to access" subsection when not applying for the "Artifacts Available" badge), please specify 'N/A'.

A.3.1 How to access

Clone repository from Github. Final stale URL: github.com/Machine-Learning-Security-Lab/mia_prune/tree/v1.0.0.

A.3.2 Hardware dependencies

GPU is required to accelerate the neural network training and membership inference attacks.

A.3.3 Software dependencies

Python 3 is required. The code is tested using Python 3.8. The required Python libraries (e.g., Pytorch) is provided in the requirement.txt file.

A.3.4 Data sets

All the datasets are publicly available. The repository contains all the link to the datasets.

A.3.5 Models

The code is provided to generate machine learning models.

A.3.6 Security, privacy, and ethical concerns

N/A

A.4 Installation

Obligatory. Describe the setup procedures for your artifact targeting novice users (even if you use a VM image or access to a remote machine).

First, install Python 3.8 with a virtual environment. Second, install the required Python libraries in the requirement.txt file. Third, create a folder to store the downloaded datasets. Fourth, create a folder to store the trained and pruned models.

A.5 Experiment workflow

Describe the high-level view of your experimental workflow and how it is implemented, invoked and customized (if needed), i.e. some OS scripts, IPython/Jupyter notebook, portable CK workflow, etc. This subsection is optional as long as the experiment workflow can be easily embedded in the next subsection.

The workflow for MIA attacks is summarized as follow: 1) Train an original neural network. 2) Prune the model and fine-tune the model. 3) Conduct membership inference attacks on the pruned model. 4) Conduct membership inference attacks on the original model.

The workflow for MIA defenses is summarized as follow: 1) Train an original neural network. 2) Based on an original model, prune the model and fine-tune the model with defense. 3) Evaluate the performance of defense by conduct membership inference attacks on the pruned model with defense.

A.6 Evaluation and expected results

Obligatory. Start by listing the main claims in your paper. Next, list your key results and detail how they each support the main claims. Finally, detail all the steps to reproduce each of the key results in your paper by running the artifacts. Describe the expected results and the maximum variation of empirical results (particularly important for performance numbers).

The paper presents the following main claims. 1) Neural network pruning increases the privacy risks of pruned models in terms of membership inference attacks. 2) The proposed SAMIA has advantages in identifying the pruned models' prediction divergence by using finergrained prediction metrics. 3) The proposed PPB protects the fine-tuning process of neural network pruning by reducing the prediction gaps based on their KL-divergence distances.

The key results include: 1) membership inference attack accuracy of the pruned models is usually higher than that of the original models. 2) the proposed SAMIA attack achieves the highest attack accuracy in most cases compared with baseline attacks. 3) the proposed PPB defense is effective in protecting all pruning approaches from attacks and can reduce the attack accuracy.

The steps to reproduce the first key results include: 1) Train an original neural network. 2) Prune the model and fine-tune the model. 3) Conduct SAMIA attacks on the pruned model. 4) Conduct SAMIA attacks on the original model.

The steps to reproduce the second results include: 1) Derive the pruned models in the first key result. 2) Conduct SAMIA attacks and baseline attacks on the pruned models.

The steps to reproduce the third results include: 1) Derive the original models in the first key result. 2) Prune the model and fine-tune the model with PPB defense. 3) Conduct SAMIA attacks on the pruned models.

Detailed examples for running these experiments are provided in the README file.

A.7 Experiment customization

The dataset can be changed by modifying the dataset.py file. The neural network architecture can be changed by modifying the models.py file. The pruning method can be changed by modifying the pruner.py file.

A.8 Notes

A.9 Version

Based on the LaTeX template for Artifact Evaluation V20220119.