

A Artifact Appendix

A.1 Abstract

The artifact discovers the vulnerability gap between manual models and automl models against various kinds of attacks (adversarial, poison, backdoor, extraction and membership) in image classification domain. It implements all datasets, models, and attacks used in our paper.

We expect the artifact could support the paper’s claim that automl models are more vulnerable than manual models against various kinds of attacks, which could be explained by their small gradient variance.

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

- **Binary:** on [pypi](#) with any platform.
- **Model:** ResNet and other model pretrained weights are available with `-official` flag to download them automatically at first running.
- **Data set:** CIFAR10, CIFAR100 and ImageNet32. Use `-download` flag to download them automatically at first running. ImageNet32 requires manual set-up at their [website](#) due to legality.
- **Run-time environment:**
At any platform (Windows and Ubuntu tested).
‘Pytorch’ and ‘torchvision’ required. (CUDA recommended)
‘adversarial-robustness-toolbox’ required for extraction attack and membership attack.
- **Hardware:** GPU with CUDA support is recommended.
- **Execution:** Model training and backdoor attack would be time-consuming. It would cost more than half day on a Nvidia Quodro RTX6000.
- **Metrics:** Model accuracy, attack success rate, clean accuracy drop and cross entropy.
- **Output:** console output and saved model files (.pth).
- **Experiments:** OS scripts. Recommend to run scripts 3-5 times to reduce the randomness of experiments.
- **How much disk space required (approximately)?:** less than 5GB.
- **How much time is needed to prepare workflow (approximately)?:** within 1 hour.
- **How much time is needed to complete experiments (approximately)?:** 3-4 days.
- **Publicly available?:** on GitHub.
- **Code licenses (if publicly available)?:** GPL-3.
- **Archived (provide DOI)?:** GitHub commit [ff315234561602203615d11166f8f346b4f29dd4](#).

A.3 Description

A.3.1 How to access

- **GitHub:** `pip install -e .`
- **PYPI:** `pip install autovul`
- **Docker Hub:** `docker pull local0state/autovul`
- **GitHub Packages:** `docker pull ghcr.io/ain-soph/autovul`

A.3.2 Hardware dependencies

Recommend to use GPU with CUDA and CUDNN. Less than 5GB disk space is needed.

A.3.3 Software dependencies

You need to install `python==3.9, pytorch==1.9.x, torchvision==0.10.x` manually.

ART (IBM) is required for extraction attack and membership attack. `pip install adversarial-robustness-toolbox`

A.3.4 Data sets

We use CIFAR10, CIFAR100 and ImageNet32 datasets. Use `-download` flag to download them automatically at first running. ImageNet32 requires manual set-up at their [website](#) due to legality.

A.3.5 Models

ResNet and other model pretrained weights are available with `-official` flag to download automatically at first running.

A.4 Installation

- **GitHub:** `pip install -e .`
- **PYPI:** `pip install autovul`
- **Docker Hub:** `docker pull local0state/autovul`
- **GitHub Packages:** `docker pull ghcr.io/ain-soph/autovul`

(optional) Config Path

You can set the config files to customize data storage location and many other default settings. View `/configs_example` as an example config setting.

We support 3 configs (priority ascend):

- **package (DO NOT MODIFY)**
 - `autovul/base/configs/*.yaml`
 - `autovul/vision/configs/*.yaml`
- **user**
 - `./autovul/configs/base/*.yaml`
 - `./autovul/configs/vision/*.yaml`
- **workspace**
 - `./configs/base/*.yaml`
 - `./configs/vision/*.yaml`

A.5 Experiment workflow

Bash Files

Check the bash files under `/bash` to reproduce our paper results.

Train Models

You need to first run `/bash/train.sh` to get pretrained models.

If you run it for the first time, please run with `-download` flag to download the dataset:

```
bash ./bash/train.sh "--download"
```

Run Attacks

```
/bash/adv_attack.sh
```

```
/bash/poison.sh
```

```
/bash/backdoor.sh
```

```
/bash/extraction.sh
```

```
/bash/membership.sh
```

Run Other Exps

```
/bash/grad_var.sh
```

```
/bash/mitigation_backdoor.sh
```

```
/bash/mitigation_extraction.sh
```

For mitigation experiments, the architecture names in our paper map to:

- **darts-i:** `diy_deep`
- **darts-ii:** `diy_noskip`
- **darts-iii:** `diy_deep_noskip`

These are the 3 options for `-model_arch {arch}` (with `-model darts`)

A.6 Evaluation and expected results

Our paper claims that automl models are more vulnerable than manual models against various kinds of attacks, which could be explained by low gradient variance.

Training

Most models around 96%-97% accuracy on CIFAR10.

Attack

For automl models on CIFAR10,

- **adversarial:** higher success rate around 10% ($\pm 4\%$).
- **poison:** lower accuracy drop around 5% ($\pm 2\%$).
- **backdoor:** higher success rate around 2% ($\pm 1\%$) and lower accuracy drop around 1% ($\pm 1\%$).
- **extraction:** lower inference cross entropy around 0.3 ($\pm 0.1\%$).
- **membership:** higher auc around 0.04 ($\pm 0.01\%$).

Others

- **gradient variance:** automl with lower gradient variance around 2.2 (± 0.5).
- **mitigation architecture:** deep architectures (`darts-i, darts-iii`) have larger cross entropy for extraction attack around 0.5, and higher accuracy drop for poisoning attack around 7% ($\pm 3\%$) with setting of 40% poisoning fraction.

A.7 Experiment customization

Use `-h` or `-help` flag for example python files to check available arguments.

A.8 Notes

A.9 Methodology

Submission, reviewing and badging methodology:

- <https://www.acm.org/publications/policies/artifact-review-badging>
- <http://cTuning.org/ae/submission-20201122.html>
- <http://cTuning.org/ae/reviewing-20201122.html>