MLforCyberSec Project

Wenwei Zhang(wz2037) Linnan Zhang(lz2400) Tianhao Wang(tw2245)

Link to Github

```
├── data
    └── clean_validation_data.h5 // this is clean data used to evaluate the BadNet and design the backdoo
    └── clean_test_data.h5
    └── sunglasses_poisoned_data.h5
    └── anonymous_1_poisoned_data.h5
    └── Multi-trigger Multi-target
        └── eyebrows_poisoned_data.h5
        └── lipstick_poisoned_data.h5
        └── sunglasses_poisoned_data.h5
├── models
    └── sunglasses_bd_net.h5
    └── sunglasses_bd_weights.h5
    └── multi_trigger_multi_target_bd_net.h5
    └── multi_trigger_multi_target_bd_weights.h5
    └── anonymous_1_bd_net.h5
    └── anonymous_1_bd_weights.h5
    └── anonymous_2_bd_net.h5
    └── anonymous_2_bd_weights.h5
    └── repair_sunglasses_bd_net.h5
    └── repair_anonymous_1_bd_net.h5
    └── repair_anonymous_2_bd_net.h5
    └── repair_multi_trigger_multi_target_bd_net.h5
├── architecture.py
├── sparse-fine-pruning.py
├── Geval_Anonymous_1.py
├── Geval_Anonymous_2.py
├── Geval_Multi_trigger.py
├── Geval_Sunglasses.py
└── eval.py // this is the evaluation script
```

I. Dependencies

1. Python 3.6.9
2. Keras 2.3.1
3. Numpy 1.16.3
4. Matplotlib 2.2.2
5. H5py 2.9.0
6. TensorFlow 2.7.0

II. Validation Data

1. Download the validation and test datasets from here and store them under `data/` directory.

2. The dataset contains images from YouTube Aligned Face Dataset. We retrieve 1283 individuals each containing 9 images in the validation dataset.
3. sunglasses_poisoned_data.h5 contains test images with sunglasses trigger that activates the backdoor for sunglasses_bd_net.h5. Similarly, there are other .h5 files with poisoned data that correspond to different BadNets under models directory.

III. Evaluating the Backdoored Model

1. The DNN architecture used to train the face recognition model is the state-of-the-art DeepID network. This DNN is backdoored with multiple triggers. Each trigger is associated with its own target label.
2. To evaluate the backdoored model, execute `eval.py` by running:
   `python3 eval.py <clean validation data directory> <model directory>` .
   E.g., `python3 eval.py data/clean_validation_data.h5 models/sunglasses_bd_net.h5` . Clean data classification accuracy on the provided validation dataset for sunglasses_bd_net.h5 is 97.87 %.

IV. Repair Network

1. The repaired network for all 4 bad nets are: `models/repair_sunglasses_bd_net.h5` `models/repair_anonymous_1_bd_net.h5` `models/repair_anonymous_2_bd_net.h5` `models/repair_multi_trigger_multi_target_bd_net.h5` .
   They are created by runing command:
   `python3 sparse-fine-pruning.py <model path> <clean data path>`
   e.g. `python3 sparse-fine-pruning.py models/anonymous_1_bd_net.h5 data/clean_validation_data.h5`
2. Each repaired network has a corresponding evaluation script:
   - `models/repair_sunglasses_bd_net.h5` : `Geval_Sunglasses.py`
   - `models/repair_anonymous_1_bd_net.h5` : `Geval_Anonymous_1.py`
   - `models/repair_anonymous_2_bd_net.h5` : `Geval_Anonymous_2.py`
   - `models/repair_multi_trigger_multi_target_bd_net.h5` : `Geval_Multi_trigger.py`

   Each script can be run by:
   - `python3 Geval_*.py`

V. Project Report

Project report can be read here.