Jigsaw Puzzle: Selective Backdoor Attack to Subvert Malware Classifier

Overview

We provide the code for most of the main experiments. This includes JP attack in the feature space, problem space, and how to run MNTD defense against baseline backdoor, JP attack (feature space), and JP attack (problem space). We also include the JP transfer attack with access to limited training data and mismatched model.

Get Started

After download the code, please uncompress the `data/apg/apg.zip`, you will get three files: `apg-X.json`, `apg-y.json`, and `apg-meta.json`.

Environment set up

You need CUDA 9.0 available. We use **Python 3.6.8, Tensorflow 1.12.0, Keras 2.2.5, and PyTorch 1.1.0** for most of the code. Please download Anaconda here: https://www.anaconda.com/products/distribution and then run the following command in shell to install Python and related dependencies.

```
conda create -n backdoor python==3.6.8
conda activate backdoor
pip install scipy==1.3.0
pip install numpy==1.16.4
pip install --ignore-installed tensorflow-gpu==1.12.0
pip install keras==2.2.5
pip install -r requirements.txt
conda install ujson==1.35 # ujson cannot be installed via pip
```

Train a clean classifier

1. Train an SVM clean classifier. It will select 10000 features using L2 regularization. Also, we will get the top benign features from SVM weights, which will be used for baseline backdoor attack.

```
./run_backdoor_svm.sh
```

2. Train an MLP clean classifier as the defenders' model, also serves as the initial model of the alternate optimization. Use the first section of the code and comment all the other parts.

```
./run_backdoor_mlp.sh
```

JP attack (feature-space)

1. Run the following script. You may need to change your CUDA device IDs accordingly. We use 8 GPU cards to run 10 families.

```
./run_jp_feature_space.sh
```

JP attack (problem-space)

1. Based on the selected 10000 features, first we collect all available gadgets for each feature. We are able to find gadgets for 2171 features. Please refer `gadget_collect/README.md` for details. We obtained the code from the original authors of "Intriguing properties of adversarial ML attacks in the problem space" [65] and extend to all feature types based their implementation (we keep most of the instructions and license from the original README).

2. The output is `data/apg/realizable_features.txt` which contains the feature indices and names of those 2171 realizable features, and `data/apg/fea-side-effect-fea-mapping-depth10-2171fea.p` contains the mapping of each feature and their corresponding side-effect features.

3. Obtain the trigger using these 2171 realizable features (you may change the family and GPU device IDs accordingly)

```
./run_jp_realizable.sh
```

4. After we got the trigger, we run the following to add corresponding side-effect features and perform the problem space attack:

```
./run_jp_problem_space_after_realizable.sh
```

MNTD against JP attack (feature space and problem space)

Please run different code blocks inside `run_mntd.sh` (and **comment all other code blocks**) to train the MNTD models and evaluate on backdoored models.

- Step 1. Train clean shadow models (training and validation set)
  - using `mntd_train_basic_benign.py`
- Step 2. Train backdoored shadow models (training and validation set)
  - using `mntd_train_basic_jumbo.py`
- Step 3. Train the target clean models (testing set)
  - using `mntd_train_target_benign.py`
- Step 4. Train the target backdoored models (testing set)
  - testing set 1 (baseline backdoor): `mntd_train_basic_trojaned.py`
  - testing set 2 (JP attack, feature space or problem space): `mntd_train_target_jp.py`
  - testing set 3 (JP attack, problem space): `mntd_train_target_jp.py`
- Step 5. Train the meta classifier, with query tuning. The trained meta classifier will be used to evaluate different kinds of backdoored models, either feature space or problem space.
  - need to run after step 1, 2, 3, 4
  - using `mntd_run_meta.py`
- Step 6. Train the meta classifier, without query tuning, similar as with query tuning.
  - need to run after step 1, 2, 3, 4
  - using `mntd_run_meta.py`
- Step 7. evaluate the pre-trained meta-classifier in problem space, with and without query tuning.
  - need to run after step 5 and 6

    - using `mntd_run_meta.py`
- Step 8. Train another meta classifier to evaluate the baseline backdoor attack
    - need to run after step 1, 2, 3, 4
    - using `mntd_run_meta.py`

JP transfer attack (limited data and mismatched model)

1. Run the following script. It works for both feature space and problem space.

```
run_backdoor_transfer.sh
```