Trainwreck Attack Tool Manual

Overview

The Trainwreck Attack Tool is a framework for implementing and evaluating train-time data poisoning attacks against image classification models. It provides a comprehensive implementation of the Trainwreck attack methodology along with several baseline attack methods for comparison. This manual explains how to set up and use the tool to perform attacks and evaluate their effectiveness.

System Requirements

- Python 3.8+
- CUDA-compatible GPU (recommended: NVIDIA Titan XP or better)
- At least 16GB RAM
- At least 50GB free disk space (for ImageNet-100 dataset)

Installation

Step 1: Clone the Repository

```
git clone https://github.com/Ljeianvmeu/Trainwreck_ImageNet-100.git
cd Trainwreck_ImageNet-100
```

Step 2: Install Dependencies

```
pip install -r requirements.txt
```

The main dependencies include:

- PyTorch (with CUDA support)
- torchvision
- NumPy
- matplotlib
- tqdm
- scikit-learn
- cleverhans (for adversarial attacks)
- PIL (Pillow)

Dataset Preparation

The tool supports multiple datasets, with particular optimization for ImageNet-100.

ImageNet-100 Setup

The tool uses the ImageNet-100 dataset, which is a 100-class subset of the full ImageNet dataset. The subset follows the class selection defined in the CMC GitHub repository.

To prepare the dataset:

1. Download the full ImageNet dataset (ILSVRC2012)
2. Use the class list from `https://github.com/HobbitLong/CMC/blob/master/imagenet100.txt` to extract the relevant 100 classes
3. Create a new directory structure with these 100 classes
4. Compress the resulting dataset into a zip file
5. Place the zip file in the project root directory

The tool will automatically:

- Extract the dataset if needed
- Organize it into train/val folders preserving the class structure
- Create symbolic links to save disk space

Downloading Pre-trained Weights

Before running the tool, you need to download the pre-trained model weights:

1. Download the model weights (weights.zip) from shared link: https://drive.google.com/drive/folders/1uUIW788I3c-k_y0r2elVxN9kygxk9NZf?usp=sharing
2. Extract the downloaded weights to the `models/weights` folder
3. Verify the weights were properly extracted by checking that the following files in this format exist:

```
models/weights/imagenet100-clean-surrogate-resnet50-30epochs.pth
```

These weights are essential for the surrogate models to function properly and for attack generation.

Core Functionalities

Running a Complete Attack Evaluation

To run a full attack cycle with all attack methods:

```
python main.py
```

This will:

1. Prepare the dataset
2. Extract features using a ViT-L-16 model
3. Train or load a surrogate model
4. Construct attacked datasets using all attack methods

5. Train target models on each attacked dataset
6. Compare and visualize results

Output Files

The tool generates several output files:

- `results/attack_results.txt`: Summary of attack effectiveness
- `results/attack_comparison.png`: Bar chart comparing attack methods
- `results/attack_samples/`: Directory containing visualizations of attacked samples
- `results/raw/`: Raw training metrics for each model
- `attack_data/instructions/`: Attack instruction files
- `attack_data/*/poisoned_data/`: Directories containing poisoned images

Troubleshooting

Common Issues

1. **CUDA Out of Memory**
    - Reduce batch size in main.py
    - Try using a GPU with more memory
2. **Missing Dataset**
    - Ensure the ImageNet-100 dataset is properly structured
    - Check if paths in commons.py point to the correct location
3. **Failed Attack Visualization**
    - Ensure attack instructions exist and contain valid indices
    - Check if poisoned data directories exist and contain files

Logs

The tool provides detailed timestamped logs during execution. If you encounter issues, check the console output for error messages.

References

This tool implements and extends the Trainwreck attack as described in the project JanZahalka/trainwreck: A train-time, black-box damaging adversarial attack on image classifiers. and paper [2311.14772] Trainwreck: A damaging adversarial attack on image classifiers. For more information, refer to the accompanying links.