A Artifact Appendix

A.1 Abstract

This appendix offers concise instructions for utilizing our artifact. For comprehensive guidance, please consult the **README.md** file located in the artifact within the root directory of the compressed package.

A.1.1 How to access

We put the artifacts on Zenodo (the link is https://doi.org/10.5281/zenodo.13293403) and uploaded them to Github at the same time (the link is https://github.com/demo123attack/Artifact-CCS24 to facilitate online preview. **Please note that the version on Github does not contain the complete artifacts because uploading large files (models/best) on Github is restricted. So please be sure to use the model on Zenodo!** The reason why we chose to release the public version on Github is to facilitate continuous maintenance and collaboration.

A.2 Environment Setup

This implementation has been successfully tested on **Windows 11** with **Python 3.7.16** and **TensorFlow 2.9.3**, utilizing an **NVIDIA GeForce RTX 3070 Ti Laptop GPU 8GB**, a 12th Gen Intel(R) Core(TM) i7-12700H processor, and 16GB of RAM. For optimal performance, we recommend using GPU acceleration (e.g., **NVIDIA GeForce RTX 3070 Ti Laptop GPU 8GB**). If you choose to use GPU acceleration, please install the necessary components on Windows by following the *"requirements_windows_gpu.txt"* file. While we suggest using a Windows environment and a Python 3.7 conda environment for compatibility, these are not mandatory. For detailed instructions on setting up your environment, please refer to the ‘README.md‘ file included in the artifact.

A.2.1 Notes for AEC Review

To facilitate AEC review, we summarize the materials and instructions related to the three badges as follows:

1. *Artifacts Available.* This artifact have been uploaded to Zenodo and made public on GitHub.

2. *Artifacts Evaluated.* To train our classifier, please run "code/train_test.py", which which utilizes 5-fold cross-validation on the Ground truth dataset and saves our models in the "models/results/" folder.

   ```
   python train_test.py --train
   ```

   To evaluate the effectiveness of our classifier, please run "code/train_test.py", the results are displayed in "experiment/best-model.png".

```
python train_test.py --test
```

3. *Results Reproduced.* Please see the above section in this README (”Experiment Results”).