USENIX Security '25 Artifact Appendix - CertTA: Certified Robustness Made Practical for Learning-Based Traffic Analysis

Jinzhu Yan[1] Zhuotao Liu[1,2,✉] Yuyang Xie[1] Shiyu Liang[3] Lin Liu[4] Ke Xu[1,2]
[1] Tsinghua University [2] Zhongguancun Laboratory
[3] Shanghai Jiao Tong University [4] National University of Defense Technology

A Artifact Appendix

A.1 Abstract

This appendix contains research artifacts associated with our USENIX Security'25 paper CertTA, which presents the first solution that provides certifiable robustness against multi-modal adversarial attacks in traffic analysis models. We implement a prototype of CertTA and extensively evaluate it against three categories of multi-modal adversarial attacks across six traffic analysis models and two datasets. Our experimental results demonstrate that CertTA provides significantly stronger robustness guarantees than the state-of-the-art approaches when confronting adversarial attacks.

In this appendix, we present a repository containing two traffic datasets, three attack methodologies to generate adversarial flows, six supervised traffic analysis models, CertTA and three baseline certification methods to construct certified traffic analysis models, three unsupervised anomaly detectors and a novel integrated system that synergistically combines certified traffic analysis models with anomaly detectors. In order to facilitate the reproduction of our key experimental results and foster future research, we also provide step-by-step instructions for running our CertTA prototype and a step-by-step demo of key experiments in the repository.

A.2 Description & Requirements

Our artifacts are organized into the following directories:

- **dataset/** contains the CICDOH20 and TIISSRC23 datasets, including the processed json files and original PCAP files of flow samples.
- **model/** contains the implementations of six supervised traffic analysis systems (*i.e.,* kFP, Kitsune (supervised), Whisper (supervised), DFNet, YaTC and TrafficFormer) and three unsupervised anomaly detection systems (*i.e.,* KMeans, Kitsune, Whisper).
- **certification/** contains the implementations of CertTA's multi-modal smoothing mechanism and the functions for solving CertTA's robustness region against multi-modal adversarial perturbations.
- **attack/** contains the implementations of three multi-modal adversarial attacks (*i.e.,* Blanket, Amoeba, Prism).
- **BARS/** contains the implementations of a baseline certification method BARS.
- **evaluation/** contains the source codes for training and evaluating certified traffic analysis models. Our prototype supports both CertTA and baseline certification methods (*i.e.,* VRS, BARS and RS-Del) for building certified traffic analysis models.
- **integration/** contains the source codes for building and evaluating the integrated system of anomaly detectors and certified traffic analysis models.

A.2.1 Security, Privacy, and Ethical Concerns

Our artifacts are non-destructive, and there is no security or privacy risk when running the CertTA prototype. The datasets used in our evaluations are publicly available, and all third-party artifacts are based on open-source implementations. We strictly followed all terms of use, and no private or sensitive data were accessed or disclosed.

A.2.2 How to Access

Our repository is archived on Zenodo[1], with a GitHub[2] mirror for ease of access. The research artifacts, including the source code of our CertTA prototype and the experimental artifacts (*e.g.,* the datasets, the detailed implementations of traffic analysis models, adversarial attack methodologies and baseline approaches), can be accessed via these public repositories under an open-source license.

A.2.3 Hardware Dependencies

As a reference, our experiments are conducted on a Supermicro SYS-740GP-TNRT server with two Intel(R) Xeon(R) Gold 6348 CPUs (2 × 28 cores), 512GB RAM, one NVIDIA

✉ Corresponding author.

[1] Available at https://doi.org/10.5281/zenodo.15580292
[2] Available at https://github.com/InspiringGroup-Lab/CertTA

A100 GPU and two NVIDIA GeForce RTX 4090 GPUs. Since we have not verified the compatibility of our artifacts in a CPU-only hardware environment, we recommend conducting experiments on a machine equipped with GPUs.

A.2.4 Software Dependencies

As a reference, our artifacts have been successfully tested in Ubuntu 20.04 server with Python 3.8.18. Based on the Anaconda distribution, we provide step-by-step instructions about software environment setup in our repository. All required Python packages are listed in the "environment.yml" file.

A.2.5 Benchmarks

The CICDOH20 and TIISSRC23 traffic datasets, the pre-trained checkpoints of traffic analysis models YaTC and TrafficFormer are required by the experiments with our artifacts. These datasets and model checkpoints are completely provided in our Zenodo repository, while the PCAP files of the two traffic datasets and the pre-trained checkpoint of the TrafficFormer model are not provided in our Github repository due to the repository size limit.

A.3 Set-up

A.3.1 Installation

1. Download the zip file of our Zenodo repository and unzip the complete artifacts.
2. Ensure that you have *conda* installed on your system. If you do not have *conda*, you can install it as part of the Anaconda distribution or Miniconda.
3. Open a terminal or command prompt.
4. Create a new *conda* environment with the name of your choice (*e.g.,* CertTA) and install all the required packages listed in the "environment.yml" file:

```
conda create -n CertTA -f environment.yml
```

5. Once the environment is created, activate it by running:

```
conda activate CertTA
```

6. Switch to the root directory of the repository (*i.e.,* the *CertTA_public* directory).

A.3.2 Basic Test

To check that all required software components are used and functioning fine, run the command:

```
python evaluation/setup_check.py
```

The evaluation environment is properly initialized if "Setup Successful!" appears at the end of the output.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): CertTA provides much stronger robustness guarantees against multi-modal adversarial attacks than the SOTA approaches (*i.e.,* VRS, BARS and RS-Del), while imposing very minimal performance reductions on clean traffic. This is proven by the experiment (E1) described in Section 5.2 whose results are illustrated in Table 5, Figure 6 and Figure 7.

(C2): The synergistic integration between CertTA and anomaly detection systems can create a fundamental dilemma for the attacker, thereby achieving consistently high Defense Success Rate against adversarial attacks with varying attack intensities. This is proven by the experiment (E2) described in Section 5.3 whose results are illustrated in Figure 8.

A.4.2 Experiments

Running all the experiments in our paper - involving two traffic datasets, three adversarial attack methodologies, six supervised traffic analysis systems, CertTA and three baseline certification methods, three unsupervised anomaly detectors and an integrated system - might take several weeks to complete. To facilitate a quick verification of the major claims made in our paper, we provide a step-by-step demo in the repository to reproduce key experiments. In this demo, we use the CICDOH20 dataset and the YaTC model to build CertTA-certified/baseline traffic analysis models and reproduce key experimental results based on these models.

(E1): [Certified Robustness against Adversarial Attacks] [4 human-hours + 6 compute-hours]: Using the CICDOH20 dataset and the YaTC model, we reproduce key experimental results in Table 5 and Figure 6 to verify the major claim (C1).

Step 1: Train CertTA-certified/baseline traffic analysis models. The trained model checkpoints and training logs will be saved. To save time, we provide pre-saved model checkpoints to proceed with subsequent experiment steps.

Step 2: Evaluate the performance on clean traffic. We evaluate the classification performance of CertTA-certified/baseline traffic analysis models on clean traffic. The accuracy/precision/recall/F_1-score of each traffic class and their macro aggregation will be saved, which can be compared with the results in Table 5.

Step 3: Measure the robustness region of the CertTA-certified traffic analysis model. Based on the robustness region offered by CertTA-certified traffic analysis model, we plot the CDF curves of certified accuracy under different robustness radius. The plotted figure will be saved, which can be compared with the sub-figure on the left-most column of Figure 6.

Step 4: Generate adversarial flows. We train the Blanket, Amoeba and Prism attack models to generate adversarial flows. The trained model checkpoints, training logs and generated adversarial flows will be saved. To save time, we provide pre-saved datasets of these adversarial flows to proceed with subsequent experiment steps.

Step 5: Evaluate the certified accuracy against adversarial flows. We evaluate the certified accuracy of CertTA-certified/baseline traffic analysis models against adversarial flows. The certified accuracy of each traffic class and their macro aggregation will be saved, which can be compared with the results in the right-side three columns of Figure 6.

Step 6: Verify the reproduced experimental results.

(E2): [Integration with Anomaly Detection] [3 human-hours + 5 compute-hours]: Using the CICDOH20 dataset and the YaTC model, we reproduce key experimental results in Figure 8 to verify the major claim (C2).

Step 1: Generate adversarial flows of different intensities. We train the Blanket, Amoeba and Prism attack models to generate adversarial flows with different levels of attack intensities. The trained model checkpoints, training logs and generated adversarial flows will be saved. To save time, we provide pre-saved datasets of these adversarial flows to proceed with subsequent experiment steps.

Step 2: Evaluate the non-certified traffic analysis model against adversarial flows. The accuracy of each traffic class and their macro aggregation will be saved, which can be compared with the results in the left two columns of Figure 8.

Step 3: Train the anomaly detector. We train the Kitsune model for anomaly detection against adversarial flows. The trained model checkpoint and training log will be saved.

Step 4: Evaluate the standalone anomaly detector against adversarial flows. The False Positive Rate on clean traffic and the True Positive Rate on adversarial flows will be saved, which can be compared with the results in the middle two columns of Figure 8.

Step 5: Evaluate the standalone certified traffic analysis model against adversarial flows. The accuracy of each traffic class and their macro aggregation will be saved, which can be compared with the results in the right two columns of Figure 8.

Step 6: Evaluate the integrated system against adversarial flows. The Defense Success Rate on each traffic class and their macro aggregation will be saved, which can be compared with the results in the middle two columns and right two columns of Figure 8.

Step 7: Verify the reproduced experimental results.

The commands to be executed in each experiment step and the organization of the experimental results are detailed in the Demo.md file. All intermediate model checkpoints and experimental results are saved to enable easy comparisons with the results found in our paper. Beyond this demo, you can also follow the aforementioned experiment steps to reproduce results using other datasets and models.

A.5 Notes on Reusability

Our artifacts integrate two traffic datasets, six traffic analysis models and three anomaly detectors using different flow representations (*e.g.,* flow statistics, raw flow sequences and raw bytes) and architectures (*e.g.,* traditional machine learning based, deep learning based and Transformer based), three categories of adversarial attacks (*e.g.,* Generative Adversarial Network based, Reinforcement Learning based and explicit modeling based) and four robustness certification methods. Based on the step-by-step instructions in our repository, these artifacts can be reused to facilitate future studies and deployment of traffic analysis systems.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2025/.