USENIX

THE ADVANCED COMPUTING
SYSTEMS ASSOCIATION

# CertTA: Certified Robustness Made Practical for Learning-Based Traffic Analysis

Jinzhu Yan, *Tsinghua University;* Zhuotao Liu, *Tsinghua University and Zhongguancun Laboratory;* Yuyang Xie, *Tsinghua University;* Shiyu Liang, *Shanghai Jiao Tong University;* Lin Liu, *National University of Defense Technology;* Ke Xu, *Tsinghua University and Zhongguancun Laboratory*

https://www.usenix.org/conference/usenixsecurity25/presentation/yan-jinzhu

This paper is included in the Proceedings of the 34th USENIX Security Symposium.

August 13–15, 2025 • Seattle, WA, USA

978-1-939133-52-6

Open access to the Proceedings of the 34th USENIX Security Symposium is sponsored by USENIX.

# 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*

## Abstract

Learning-based traffic analysis models exhibit significant vulnerabilities to adversarial attacks. Attackers can compromise these models by generating adversarial network flows with precisely optimized perturbations. These perturbations typically take two forms: additive modifications, which include packet length padding and timing delays, and discrete alterations, such as dummy packet insertion. In response to these threats, certified robustness has emerged as a promising methodology for ensuring reliable model performance in the presence of adversarially manipulated network traffic.

However, current approaches inadequately address the multi-modal nature of adversarial perturbations in network traffic, resulting in limited robustness guarantees against sophisticated attacks. To overcome this limitation, we introduce CertTA, the first solution providing certifiable robustness against multi-modal adversarial attacks in traffic analysis models. CertTA incorporates a novel multi-modal smoothing mechanism that explicitly accounts for attack-induced perturbations during the generation of smoothing samples, based on which CertTA rigorously derives robustness regions that are meaningful against these attacks. 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. Further, CertTA is universally applicable across diverse model architectures and flow representations.

## 1 Introduction

Network traffic analysis is crucial to understanding network activities and detecting cyberspace attacks. While learning-based traffic analysis models achieve superior accuracies in many traffic analysis tasks, they often struggle to maintain robustness against adversarial attacks. For instance, by properly introducing *adversarial perturbations* to the original network traffic (such as inserting dummy packets, padding the packet lengths or delaying packets), the researchers can easily undermine the performance of traffic analysis models [15, 16, 22, 26, 29, 33, 40, 43].

To address this problem, our community explored a useful paradigm, named *certified robustness*, which provably guarantees that the learning-based models are secure against certain adversarial attacks. Randomized smoothing [5] is the prevalent methodology to achieve certified robustness. Randomized smoothing transforms a base classifier $f$ into a certifiably robust model $g$ as follows. Given an input sample $\boldsymbol{x}$, it first generates a set of smoothing samples $\{\boldsymbol{s}\}$ in the vicinity of $\boldsymbol{x}$ by applying randomized perturbations over $\boldsymbol{x}$. It then feeds these smoothing samples into $f$ to collect a set of inference results $f(\{\boldsymbol{s}\})$. Finally, it outputs $g(\boldsymbol{x})$ as the majority class from $f(\{\boldsymbol{s}\})$ and mathematically derives a *robustness region* based on the probability distribution of $f(\{\boldsymbol{s}\})$. $g$ is certifiably robust because given any adversarial input $\tilde{\boldsymbol{x}}$ within the robustness region of $\boldsymbol{x}$, $g(\tilde{\boldsymbol{x}})$ is provably equivalent to $g(\boldsymbol{x})$, implying that the adversity of $\tilde{\boldsymbol{x}}$ cannot disrupt the model prediction. Based on this fundamental methodology, our community has extensively studied certified robustness in the field of Computer Vision (CV) [6, 18, 20, 42, 47] and Natural Language Processing (NLP) [13, 49, 52, 54].

Yet, enabling certified robustness in learning-based traffic analysis is fundamentally challenging due to the *multi-modality* of adversarial perturbations. Specifically, existing attacks can simultaneously apply *additive perturbations* (*e.g.,* padding the packet lengths or delaying packets) and *discrete perturbations* (*e.g.,* inserting dummy packets) when generating adversarial network traffic [15, 16, 22, 26, 29, 33, 40, 43]. However, none of the state-of-the-art (SOTA) approaches have considered certified robustness against multi-modal adversity. For instance, BARS [39] only considers additive perturbations applied to the network flow features (*e.g.,* the mean, variance and max of packet lengths) in its randomized smoothing process. As a result, the insertion of a single dummy packet

✉ Corresponding author.

to the network flow may overwhelm the robustness regions derived by BARS. Similarly, RS-Del [13], which considers the discrete perturbations over sequence data (*e.g.,* insertion, substitution and deletion), is ineffective against the additive perturbations in adversarial flows. In § 2, we present concrete quantitative results to demonstrate that the multi-modal adversity can easily undermine the robustness guarantees offered by existing SOTA approaches.

To address this problem, we present CertTA, the first approach that enables certifiably robust traffic analysis models against multi-modal adversarial attacks. CertTA is founded on a critical insight: we explicitly account for the adversarial perturbations introduced by these attacks when designing the perturbation mechanisms in our system's randomized smoothing process. This enables us to derive robustness regions that are meaningful against these attacks. To this end, CertTA proposes a novel certification framework that (i) generates smoothing samples by a multi-modal smoothing mechanism and then (ii) derives robustness regions from the complicated probability distribution of these smoothing samples.

Specifically, the multi-modal smoothing mechanism in CertTA consists of a discrete smoothing mechanism that randomly selects packets from a network flow, and an additive smoothing mechanism that applies Exponential noises to the metadata (*e.g.,* the packet length and inter-arrival time) of the selected packets. When deriving robustness regions, the multi-modality of adversarial attacks introduces several critical challenges. First, discrete perturbations can cause significant variations in the network traffic data. For instance, the insertion of a single dummy packet can result in significant displacement of the packet sequence (a key feature used by many models [8, 25, 35, 38]), or significant changes in various statistical flow features. Second, the robustness regions derived from additive perturbations exhibit non-linear relationships with the number of input dimensions. This problem, known as the "curse of dimensionality" [3, 17, 36], substantially diminishes the effectiveness of the robustness regions when processing extended flow sequences or analyzing multiple flow statistics. These challenges are effectively addressed in CertTA's certification framework.

**Contributions.** The major contribution of this paper is the design, mathematical construction, and evaluation of CertTA, the first approach provides certifiably robustness against multi-modal adversarial attacks in traffic analysis models. We extensively evaluate CertTA over six heterogeneous traffic analysis models against three categories of adversarial attacks. The experimental results show that CertTA exhibits the following advantages over the SOTA approaches.

(i) Generality. CertTA is universally applicable to heterogeneous traffic analysis models trained 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). Meanwhile, the robustness regions derived in CertTA across different models are unified, serving as quantifiable metrics to compare the robustness across different models.

(ii) Stronger Robustness Guarantees. CertTA demonstrates significant performance advantages over the SOTA approaches in terms of certified accuracy, defined as the percentage of adversarial flows guaranteed to be classified correctly. Most notably, in scenarios where existing approaches fail to maintain any certified accuracy, CertTA achieves 99% certified accuracy with two Transformer-based models and exceeds 80% certified accuracy across the four remaining models.

(iii) Synergistic Integration with Anomaly Detection. We propose a novel integration between CertTA and anomaly detection systems, which creates a fundamental dilemma for the attackers: stealth adversarial flows (*i.e.,* with small perturbations) which may bypass the anomaly detector are ineffective against CertTA; and the adversarial flows with significant perturbations which may exceed CertTA's certified robustness regions can be easily captured by the anomaly detector. We demonstrate that the integrated system achieves consistently high Defense Success Rate against adversarial attacks with varying attack intensities.

## 2 Problem Space and Motivation

**Certified Robustness in CV and NLP.** Our community has extensively studied certified robustness in the area of CV [6, 18, 20, 42, 47] and NLP [13, 49, 52, 54]. The typical randomized smoothing approach in CV, referred to as VRS [5], provides an isotropic $\ell_2$-norm robustness radius against additive perturbations on image pixels. The paradigm of certifying an image classifier against pixel-wise additive perturbations in VRS can be straightforwardly adapted to certifying the robustness of traffic analysis models against additive perturbations on statistical flow features. Yet, due to the diverse scales of different flow features (*e.g.,* the percentage of outgoing packets is smaller than 1, while the average packet size is on the order of hundreds), the isotropic $\ell_2$-norm robustness radius is often impractical for certain features with larger scales.

Similarly, although initially proposed to provide robustness guarantees against discrete perturbations (*e.g.,* insertion, substitution and deletion) applied to sequence data like texts and binary files, RS-Del [13] can be adopted in traffic analysis models by viewing a network flow as a discrete sequence of packets. However, the robustness regions derived from this simple adaptation are inadequate when confronted with additive perturbations in network traffic, such as packet length padding and timing delays.

**Certified Robustness for Traffic Analysis.** BARS [39] represents the leading research on certified robustness in traffic analysis. Specifically, BARS improves upon VRS by taking into account the diverse scales of different flow features. It introduces a distribution transformer to customize the scale and

Table 1: Comparison with SOTA Approaches.

| | Effectiveness against Attacks | | | Supported Learning Models | | | | |
|---|---|---|---|---|---|---|---|---|
| | Additive Perturbations | Discrete Perturbations | Multi-modal Perturbations | Flow Statistics Input | Raw Flow Sequences Input | Raw Bytes Input | Base Model Architecture | Unified Metrics of Robustness |
| VRS [5]⋆ | ✔ | ✘ | ✘ | ✔ | ✔ | ✘ | any | ✘ |
| BARS [39] | ✔ | ✘ | ✘ | ✔ | ✔ | ✘ | DL-based | ✘ |
| RS-Del [13]⋆ | ✘ | ✔ | ✘ | ✔ | ✔ | ✔ | any | ✔ |
| CertTA (Ours) | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | any | ✔ |

⋆ We adapt VRS and RS-Del for traffic analysis models, as discussed in § 2.

shape of the random noise added to each dimension of the feature vector. Consequently, BARS can provide anisotropic robustness radius for different dimensions and achieve stronger robustness guarantees than previous work. However, BARS still suffers from several critical drawbacks:

(i) Lack of Generality. The applicability of BARS is limited by several factors. First, the noise-adding process in BARS is not applicable to recent Transformer-based traffic analysis models [24, 31, 55, 56] that directly take raw packet bytes as input. This is because the raw packet bytes are discrete structured data and not numerically continuous. Furthermore, the noise-shaping process in BARS depends on the gradient descent algorithm of Deep Learning (DL) based models. As a result, traffic analysis models that are based on traditional Machine Learning (ML) techniques, such as Random Forest [11] and Support Vector Machines [30], or include traditional ML modules as part of their analysis pipeline [8], are also not compatible with BARS. Finally, the $\ell_2$-norm robustness radius provided by BARS is model-specific, contingent upon the particular flow features employed during model training. This specificity results in non-comparable robustness measures across different models, hindering the establishment of a unified robustness benchmark.

(ii) Ineffectiveness against Discrete Perturbations. The $\ell_2$-norm robustness radius against additive perturbations is fragile when countered with discrete perturbations. For traffic analysis models that use flow statistics as input [7, 11, 27, 30, 57], inserting a dummy packet into the flow can cause significant changes in certain statistical features (*e.g.,* the maximum packet length). For models that take raw flow sequences (*e.g.,* the packet length sequence) as input [8, 25, 35, 38, 46] or models that take raw bytes as input [24, 31, 55, 56], the introduction of dummy packets creates input dimensional misalignment, which readily exceeds the $\ell_2$-norm robustness radius established by BARS.

(iii) Curse of Dimensionality. Mathematically, the $\ell_2$-norm robustness radius in BARS exhibits non-linear relationships with the number of input dimensions. This is a fundamental limitation of the randomized smoothing methodology [3, 17, 36]. When processing extended flow sequences or analyzing multiple flow statistics, the effectiveness of the $\ell_2$-norm robustness radius diminishes substantially.

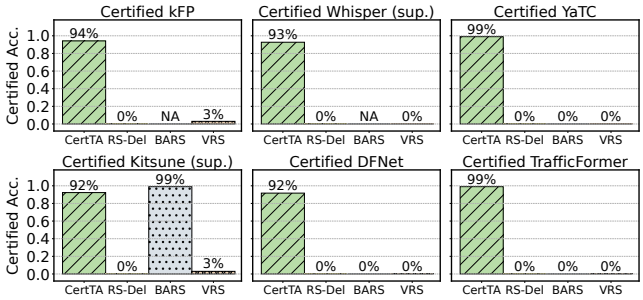


Figure 1: The robustness certified by SOTA approaches is fragile against multi-modal adversity in traffic analysis.

**Quantitative Results.** We quantify the aforementioned drawbacks of these SOTA approaches using the following experiment. Specifically, we train six heterogeneous traffic analysis models kFP [11], Kitsune [27], Whisper [8], DFNet [38], YaTC [55] and TrafficFormer [56] on the CICDOH20 [28] dataset to classify tunneling traffic that utilize DNS over HTTPS[1]. Based on these six models, we apply CertTA and three SOTA approaches to create certifiably robust traffic classifiers. Subsequently, we generate adversarial flows from the CICDOH20 dataset using the multi-modal adversarial attack methodology specified in Blanket [29]. Finally, we evaluate these certifiably robust traffic classifiers against the adversarial flows. In Figure 1, we report the certified accuracies offered by different approaches, quantified as the percentage of adversarial flows that are certified to be classified correctly. For traffic analysis models that use raw flow sequences as input (*i.e.,* Whisper (supervised) and DFNet) or use raw bytes as input (*i.e.,* YaTC and TrafficFormer), the certified accuracies offered by RS-Del, BARS and VRS are all zero. For traffic analysis models that use flow statistics as input (*i.e.,* kFP and Kitsune (supervised)), only BARS achieves 99% certified accuracy in Kitsune, while in other cases the certified accuracies offered by all SOTA approaches are nearly zero. In contrast, CertTA achieves over 92% certified accuracy across all six traffic analysis models.

**Design Goal.** As summarized in Table 1, CertTA is designed

[1] Kitsune and Whisper are unsupervised models designed for anomaly detection. We extend these two models to supervised versions for multi-class classification.

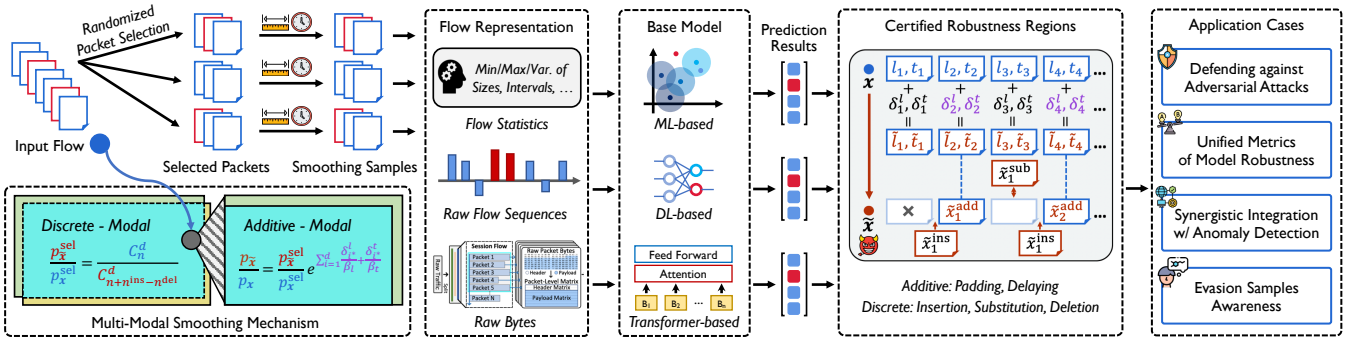


Figure 2: The workflow of CertTA.

to advance SOTA in both effectiveness and generality. Aware of the multi-modal adversity in traffic analysis, CertTA proposes the first certification framework that addresses both additive and discrete perturbations applied to network traffic. This framework offers several key advantages. First, compared to prior art, CertTA provides significantly stronger performance guarantees against existing adversarial attacks. Furthermore, CertTA is universally applicable to enable certifiably robust traffic analysis models with arbitrary architectures and flow representations. Finally, CertTA offers unified metrics to compare the robustness across various heterogeneous traffic analysis models.

**Assumptions and Threat Model.** Given an input flow $\boldsymbol{x}$, the adversary's objective is to generate an adversarial flow $\tilde{\boldsymbol{x}}$ that successfully deceives a learning-based traffic analysis model. The adversary may employ various traffic manipulation techniques, including packet insertion, substitution, deletion, packet length padding, and timing delays, either individually or in combination. Beyond random perturbations, the adversary can leverage sophisticated attack methodologies (such as [15, 16, 22, 26, 29, 33, 40, 43]) to construct these adversarial flows.

When constructing adversarial flows based on specific perturbations, it is imperative to comply with the feasibility constraints posed by network protocols [32]. For additive perturbations (*i.e.,* packet length padding and timing delays), the attacker is limited to increasing the existing length or transmission time of a packet, rather than reducing its original attributes. As presented in § 4.2, CertTA's certification quantifies the additive perturbations using the $\ell_1$-norm of the *incremental* lengths and times introduced into traffic packets. Concerning discrete perturbations, namely packet insertion, substitution, and deletion, existing attack methodologies [22, 26, 29] demonstrate the feasibility of packet insertion-based adversarial attacks. However, future adversarial attacks that rely on packet substitution and deletion could induce certain side effects, such as packet loss and retransmission. Consequently, the attackers may need to concurrently implement packet insertion to ensure adherence to feasibility constraints when executing packet deletion or substitution-based adversarial attacks. CertTA's mathematical constructions regarding discrete perturbations are exclusively based on the number of inserted, deleted, and substituted packets within a flow, without considering why these packets are induced. Therefore, CertTA is applicable regardless of the methodologies adopted when implementing the attacks. An important caveat, however, is that when the implementation of a specific deletion or substitution-based adversarial attacks is available in the future, we suggest refining CertTA's mathematical constructions based on such knowledge accordingly to further enhance its certification effectiveness against this attack.

## 3 Workflow of CertTA

A brief workflow of CertTA is shown in Figure 2. Given a base traffic analysis model $f$, CertTA constructs a certifiably robust model $g$ as follows. (i) Given an input flow $\boldsymbol{x}$ with $n$ packets, CertTA generates a set of smoothing samples $\{\boldsymbol{s}\}$, where each smoothing sample $\boldsymbol{s}$ is created by randomly selecting $d$ packets from $\boldsymbol{x}$ and adding Exponential noises to the length and inter-arrival time of each selected packet, respectively. (ii) For each smoothing sample $\boldsymbol{s}$, CertTA processes it into flow representations required by the base traffic analysis model $f$ and feeds these representations into $f$ to obtain a prediction result $f(\boldsymbol{s})$. (iii) CertTA obtains $g(\boldsymbol{x})$ (*i.e.,* the output of the certifiably robust model $g$ given input flow $\boldsymbol{x}$) by taking the majority class $y_A$ from $f(\{\boldsymbol{s}\})$ (*i.e.,* the prediction results of all smoothing samples). (iv) CertTA calculates the percentage of $y_A$ in $f(\{\boldsymbol{s}\})$ and derives the certified robustness region against both additive and discrete perturbations. Given any adversarial flow $\tilde{\boldsymbol{x}}$ encompassed by the robustness region of $\boldsymbol{x}$, $g(\tilde{\boldsymbol{x}})$ is certifiably equivalent to $g(\boldsymbol{x})$. In the following section, we present the mathematical construction underpinning the above workflow.

## 4 Robustness Certification by CertTA

CertTA proposes a multi-modal randomized smoothing mechanism to generate smoothing samples and derives the robust-

Table 2: Notations.

| Notation | Description |
|---|---|
| $\mathcal{X}, \mathcal{Y}$ | Set of traffic flows, classes |
| $\mathcal{Z}, S$ | Set of random variables, smoothing samples in $\mathcal{X}$ |
| $\boldsymbol{x}, \tilde{\boldsymbol{x}}$ | Input flow: original, adversarial |
| $f, g$ | Traffic classifier: base, smoothed |
| $\psi$ | Smoothing function |
| $y_A, p_A$ | The predicted class of input $\boldsymbol{x}$ by $g$, and the corresponding probability |
| $\underline{p_A}$ | The estimated lower bound of $p_A$ through Monte Carlo sampling |
| $\tilde{y}_A, \tilde{p}_A$ | The predicted class of adversarial input $\tilde{\boldsymbol{x}}$ by $g$, and the corresponding probability |
| $n, d$ | Number of the original, randomly selected packets in $\boldsymbol{x}$ |
| $\boldsymbol{l}, \boldsymbol{t}$ | Packet length and inter-arrival time vectors of $\boldsymbol{x}$ |
| $\boldsymbol{\delta}^l, \boldsymbol{\delta}^t$ | Noise vectors on packet length and inter-arrival time |
| $\beta_l, \beta_t$ | Hyper-parameters for Exponential noises |
| $(\cdot)_i$ | For $\boldsymbol{x}, \boldsymbol{l}, \boldsymbol{t}, \boldsymbol{\delta}^l, \boldsymbol{\delta}^t$: the $i$-th dimension |

ness region against both additive and discrete perturbations. In § 4.1, we introduce the preliminary of randomized smoothing. To construct our multi-modal smoothing mechanism, in § 4.2, we incorporate a discrete smoothing mechanism based on randomized packet selection and an additive smoothing mechanism based on Exponential noise. In § 4.3, we derive the robustness region from the probability distribution of the smoothing samples and demonstrate its advantages in countering multi-modal adversarial perturbations. The frequently used notations are summarized in Table 2.

## 4.1 Preliminary of Randomized Smoothing

Consider a traffic classification problem from traffic flows $\mathcal{X}$ to classes $\mathcal{Y}$ and let $\mathcal{Z}$ be the set of random variables in $\mathcal{X}$. Given a *base traffic classifier* $f : \mathcal{X} \to \mathcal{Y}$ and a *smoothing function* $\psi : \mathcal{X} \to \mathcal{Z}$ that generates a random variable $\boldsymbol{z}$ of smoothing samples from flow $\boldsymbol{x}$, we construct a *smoothed classifier* $g : \mathcal{X} \to \mathcal{Y}$ that returns the most probable prediction by $f$ of smoothing samples from $\psi(\boldsymbol{x})$:

$$g(\boldsymbol{x}) \triangleq \arg\max_{y \in \mathcal{Y}} p_y(\boldsymbol{x}), \quad p_y(\boldsymbol{x}) \triangleq \mathbb{P}_{\boldsymbol{z} \sim \psi(\boldsymbol{x})}(f(\boldsymbol{z}) = y). \quad (1)$$

Given an input $\boldsymbol{x}$, we denote the smoothed classifier's prediction $g(\boldsymbol{x})$ as $y_A$, and the corresponding probability $p_{y_A}(\boldsymbol{x})$ as $p_A$. In practice, a lower bound $\underline{p_A}$ of $p_A$ is estimated through Monte Carlo sampling with a confidence level $\alpha$.

Without imposing any assumptions about underlying base classifier but the probability lower bound $\underline{p_A}$ given an input $\boldsymbol{x}$, the goal of robustness certification is to derive a robustness region $R(\boldsymbol{x}) \subseteq \mathcal{X}$ around $\boldsymbol{x}$, in which the smoothed classifier's prediction is guaranteed to be consistent, *i.e.,*

$$g(\tilde{\boldsymbol{x}}) = g(\boldsymbol{x}) = y_A \Leftrightarrow p_{y_A}(\tilde{\boldsymbol{x}}) \geq \max_{y \neq y_A} p_y(\tilde{\boldsymbol{x}}), \quad \forall \tilde{\boldsymbol{x}} \in R(\boldsymbol{x}).$$

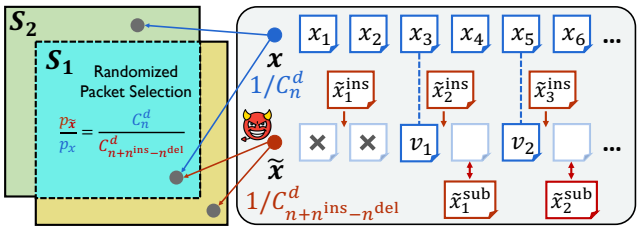


Figure 3: Partition of the smoothing samples generated by CertTA's discrete smoothing mechanism.

Denote $p_{y_A}(\tilde{\boldsymbol{x}})$ as $\tilde{p}_A$, for simplicity, we consider all classes excluding $y_A$ as a combined class, such that $p_{y_A}(\tilde{\boldsymbol{x}}) \geq \max_{y \neq y_A} p_y(\tilde{\boldsymbol{x}}) \Leftrightarrow \tilde{p}_A \geq 1/2$. Based on [5, 19], the theoretical derivations can be easily extended to multi-class certification.

## 4.2 Multi-Modal Smoothing Mechanism

In this section, we present the multi-modal smoothing mechanism in CertTA, which consists of a discrete smoothing mechanism and an additive smoothing mechanism. Specifically, given an input flow $\boldsymbol{x}$, the multi-modal smoothing mechanism generates smoothing samples in two steps: (i) it randomly selects $d$ packets from the input flow $\boldsymbol{x}$ while preserving their original orders ($d \leq n$), (ii) it applies Exponential noises to the metadata (*i.e.,* the packet length and inter-arrival time) of each selected packet. The combination of smoothing mechanisms with different modality results in a highly complicated probability distribution for the smoothing samples. Next, we give the robustness region results when the discrete and additive smoothing mechanisms are employed individually, which are the foundations of derivations in § 4.3.

The discrete smoothing mechanism is designed based on randomized packet selection. Given an input flow $\boldsymbol{x}$ with $n$ packets, it generates smoothing samples by randomly selecting $d$ packets from $\boldsymbol{x}$ while preserving their original orders ($d \leq n$). Based on the probability distribution of these smoothing samples, we give the certified robustness region against three types of discrete perturbations (*i.e.,* packet insertion, substitution and deletion) in the following lemma.

**Lemma 1.** *Consider a pair of traffic flows $\boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathcal{X}$, where $\boldsymbol{x}$ contains $n$ packets and $\boldsymbol{x}$ can be perturbed into $\tilde{\boldsymbol{x}}$ by inserting $n^{ins}$ packets, substituting $n^{sub}$ packets and deleting $n^{del}$ packets. Let $\psi^{sel}(\boldsymbol{x}, d) : \mathcal{X} \times \mathbb{Z}^+ \to \mathcal{Z}$ be the smoothing function that randomly selects $d$ packets from flow $\boldsymbol{x}$ while preserving their original orders ($d \leq n$), and define the smoothed classifier $g^{del}$ as in Equation (1). Suppose $y_A \in \mathcal{Y}$ and $\underline{p_A} \in [1/2, 1]$ satisfy $g^{del}(\boldsymbol{x}) = y_A$ and $p_A \geq \underline{p_A} \geq 1/2$, then we have $g^{del}(\tilde{\boldsymbol{x}}) = g^{del}(\boldsymbol{x}) = y_A$ if:*

$$\frac{C_n^d}{C_{n+n^{ins}-n^{del}}^d}\left(\underline{p_A} - 1 + \frac{C_{n-n^{sub}-n^{del}}^d}{C_n^d}\right) \geq \frac{1}{2}, \quad (2)$$

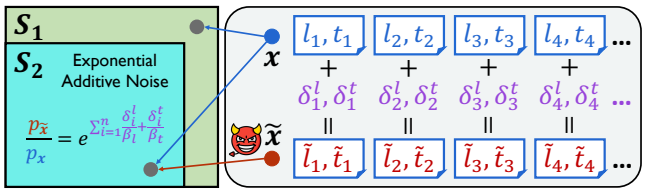


Figure 4: Partition of the smoothing samples generated by CertTA's additive smoothing mechanism.

*where $C_n^r$ is the combination formula $n!/(r!(n-r)!)$.*

*Proof.* Let $X = \{x_1, x_2, \ldots, x_n\}$ be the set of packets in flow $\boldsymbol{x}$, the packets in $\tilde{\boldsymbol{x}}$ can be divided into three categories: (i) a group of original packets $V = \{v_1, \ldots, v_{n-n^{\text{sub}}-n^{\text{del}}}\}$ in $\boldsymbol{x}$; (ii) packets $\{\tilde{x}_1^{\text{sub}}, \ldots, \tilde{x}_{n^{\text{sub}}}^{\text{sub}}\}$ obtained by substituting a group of packets in $X - V$; (iii) packets $\{\tilde{x}_1^{\text{ins}}, \ldots, \tilde{x}_{n^{\text{ins}}}^{\text{ins}}\}$ obtained through insertion. Let $\boldsymbol{v}$ be the flow obtained by selecting packets $\{v_1, \ldots, v_{n-n^{\text{sub}}-n^{\text{del}}}\}$ from flow $\boldsymbol{x}$ while preserving their original orders. As illustrated in Figure 3, we define two sets of flows $S_1, S_2$ as follows:

$$S_1 = \{\boldsymbol{s} \in \mathcal{X} : \mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{sel}}(\boldsymbol{v}, d)}(\boldsymbol{z} = \boldsymbol{s}) > 0\},$$
$$S_2 = \{\boldsymbol{s} \in \mathcal{X} : \mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{sel}}(\boldsymbol{x}, d)}(\boldsymbol{z} = \boldsymbol{s}) > 0 \ \wedge \ \boldsymbol{s} \notin S_1\}.$$

Based on the probability distributions of $\psi^{\text{sel}}(\boldsymbol{x}, d)$ and $\psi^{\text{sel}}(\tilde{\boldsymbol{x}}, d)$, we have:

$$\frac{\mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{sel}}(\tilde{\boldsymbol{x}}, d)}(\boldsymbol{z} = \boldsymbol{s})}{\mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{sel}}(\boldsymbol{x}, d)}(\boldsymbol{z} = \boldsymbol{s})} = \frac{C_n^d}{C_{n+n^{\text{ins}}-n^{\text{del}}}^d}, \quad \forall \, \boldsymbol{s} \in S_1.$$

Let $K = C_n^d / C_{n+n^{\text{ins}}-n^{\text{del}}}^d$, the lower bound of $\tilde{p}_A$ can be derived as follows:

$$\begin{aligned}
p_A &= \mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{sel}}(\boldsymbol{x}, d)}(f(\boldsymbol{z}) = y_A \ \wedge \ \boldsymbol{z} \in S_1) \\
&\quad + \mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{sel}}(\boldsymbol{x}, d)}(f(\boldsymbol{z}) = y_A \ \wedge \ \boldsymbol{z} \in S_2), \\
\tilde{p}_A &= \mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{sel}}(\tilde{\boldsymbol{x}}, d)}(f(\boldsymbol{z}) = y_A) \\
&\geq \mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{sel}}(\tilde{\boldsymbol{x}}, d)}(f(\boldsymbol{z}) = y_A \ \wedge \ \boldsymbol{z} \in S_1) \\
&\geq K * \mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{sel}}(\boldsymbol{x}, d)}(f(\boldsymbol{z}) = y_A \ \wedge \ \boldsymbol{z} \in S_1) \\
&= K * (p_A - \mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{sel}}(\boldsymbol{x}, d)}(f(\boldsymbol{z}) = y_A \ \wedge \ \boldsymbol{z} \in S_2)) \\
&\geq K * (\underline{p_A} - \mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{sel}}(\boldsymbol{x}, d)}(\boldsymbol{z} \in S_2)) \\
&= K * (\underline{p_A} - 1 + \mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{sel}}(\boldsymbol{x}, d)}(\boldsymbol{z} \in S_1)) \\
&= K * (\underline{p_A} - 1 + C_{n-n^{\text{sub}}-n^{\text{del}}}^d / C_n^d). \qquad (3)
\end{aligned}$$

Finally, we can get Equation (2) by solving the inequality that the lower bound of $\tilde{p}_A$ is not less than $1/2$. □

When employed individually, the additive smoothing mechanism generates smoothing samples by applying Exponential noises to the metadata (*i.e.,* packet lengths and inter-arrival times) of all packets in input flow $\boldsymbol{x}$. Recognizing the disparities in feature importance and numerical scale between packet length and inter-arrival time, we apply Exponential noise of different shapes to each of them. Based on the probability distribution of the generated smoothing samples, we give the certified robustness region against two types of additive perturbations (*i.e.,* packet length padding and timing delays) in the following lemma.

**Lemma 2.** *Consider a pair of traffic flows $\boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathcal{X}$, where $\boldsymbol{x}$ contains $n$ packets with packet length vector $\boldsymbol{l} = (l_1, l_2, \ldots, l_n)$ and inter-arrival time vector $\boldsymbol{t} = (t_1, t_2, \ldots, t_n)$, and $\boldsymbol{x}$ can be perturbed into $\tilde{\boldsymbol{x}}$ by adding non-negative noise vectors $\boldsymbol{\delta}^l = (\delta_1^l, \delta_2^l, \ldots, \delta_n^l)$ and $\boldsymbol{\delta}^t = (\delta_1^t, \delta_2^t, \ldots, \delta_n^t)$ to $\boldsymbol{l}$ and $\boldsymbol{t}$, respectively. Let $\psi^{add}(\boldsymbol{x}, \beta_l, \beta_t) : \mathcal{X} \times \mathbb{R}^+ \times \mathbb{R}^+ \to \mathcal{Z}$ be the smoothing function that adds random variables $\varepsilon_i^l \overset{i.i.d.}{\sim} \text{Exp}(\beta_l^{-1})$ and $\varepsilon_i^t \overset{i.i.d.}{\sim} \text{Exp}(\beta_t^{-1})$ to $l_i$ and $t_i$ $(1 \leq i \leq n)$, respectively. Define the smoothed classifier $g^{add}$ as in Equation (1). Suppose $y_A \in \mathcal{Y}$ and $\underline{p_A} \in [1/2, 1]$ satisfy $g^{add}(\boldsymbol{x}) = y_A$ and $p_A \geq \underline{p_A} \geq 1/2$, then we have $g^{add}(\tilde{\boldsymbol{x}}) = g^{add}(\boldsymbol{x}) = y_A$ if:*

$$\sum_{i=1}^{n} \frac{\beta_l + \beta_t}{\beta_l} \cdot \delta_i^l + \frac{\beta_l + \beta_t}{\beta_t} \cdot \delta_i^t \leq r^{add}, \qquad (4)$$

*where the robustness radius $r^{add} = -(\beta_l + \beta_t) \log 2(1 - \underline{p_A})$.*

*Proof.* Let $S$ be the set of all possible smoothing samples generated by $\psi^{\text{add}}(\boldsymbol{x}, \beta_l, \beta_t)$. Represent a flow $\boldsymbol{s} \in S$ by its packet length vector $\boldsymbol{l^s} = (l_1^{\boldsymbol{s}}, l_2^{\boldsymbol{s}}, \ldots, l_n^{\boldsymbol{s}})$ and inter-arrival time vector $\boldsymbol{t^s} = (t_1^{\boldsymbol{s}}, t_2^{\boldsymbol{s}}, \ldots, t_n^{\boldsymbol{s}})$. As illustrated in Figure 4, we partition $S$ into two sets of flows $S_1, S_2$ as follows:

$$S_1 = \{\boldsymbol{s} \in S : \exists \, i \in [1, n], (l_i^{\boldsymbol{s}} - l_i < \delta_i^l) \vee (t_i^{\boldsymbol{s}} - t_i < \delta_i^t)\},$$
$$S_2 = \{\boldsymbol{s} \in S : \forall \, i \in [1, n], (l_i^{\boldsymbol{s}} - l_i \geq \delta_i^l) \wedge (t_i^{\boldsymbol{s}} - t_i \geq \delta_i^t)\}.$$

Based on the probability distributions of $\psi^{\text{add}}(\boldsymbol{x}, \beta_l, \beta_t)$ and $\psi^{\text{add}}(\tilde{\boldsymbol{x}}, \beta_l, \beta_t)$, for all $\boldsymbol{s} \in S_1$, we have:

$$\frac{\mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{add}}(\tilde{\boldsymbol{x}}, \beta_l, \beta_t)}(\boldsymbol{z} = \boldsymbol{s})}{\mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{add}}(\boldsymbol{x}, \beta_l, \beta_t)}(\boldsymbol{z} = \boldsymbol{s})} = 0.$$

For all $\boldsymbol{s} \in S_2$, we have:

$$\begin{aligned}
\frac{\mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{add}}(\tilde{\boldsymbol{x}}, \beta_l, \beta_t)}(\boldsymbol{z} = \boldsymbol{s})}{\mathbb{P}_{\boldsymbol{z} \sim \psi^{\text{add}}(\boldsymbol{x}, \beta_l, \beta_t)}(\boldsymbol{z} = \boldsymbol{s})} &= \frac{\prod_{i=1}^{n} e^{-(l_i^{\boldsymbol{s}} - l_i - \delta_i^l)/\beta_l} \cdot e^{-(t_i^{\boldsymbol{s}} - t_i - \delta_i^t)/\beta_t}}{\prod_{i=1}^{n} e^{-(l_i^{\boldsymbol{s}} - l_i)/\beta_l} \cdot e^{-(t_i^{\boldsymbol{s}} - t_i)/\beta_t}} \\
&= e^{\sum_{i=1}^{n} \delta_i^l/\beta_l + \delta_i^t/\beta_t}.
\end{aligned}$$

Let $K = e^{\sum_{i=1}^{n} \delta_i^l/\beta_l + \delta_i^t/\beta_t}$, the lower bound of $\tilde{p}_A$ can be derived

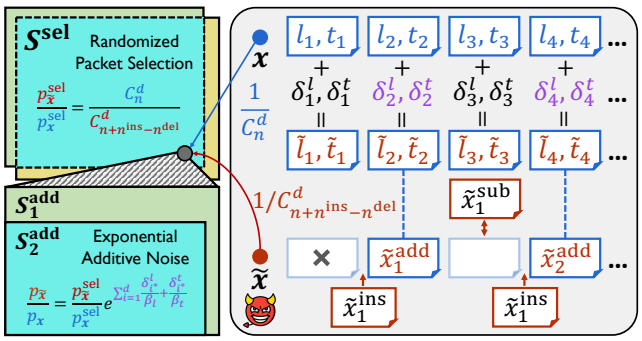


Figure 5: Partition of the smoothing samples generated by CertTA's multi-modal smoothing mechanism.

as follows:

$$\begin{aligned}
p_A &= \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{add}}(\boldsymbol{x},\beta_l,\beta_t)}(f(\boldsymbol{z})=y_A \ \wedge\ \boldsymbol{z}\in S_1) \\
&\quad + \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{add}}(\boldsymbol{x},\beta_l,\beta_t)}(f(\boldsymbol{z})=y_A \ \wedge\ \boldsymbol{z}\in S_2), \\
\tilde{p}_A &= \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{add}}(\tilde{\boldsymbol{x}},\beta_l,\beta_t)}(f(\boldsymbol{z})=y_A \ \wedge\ \boldsymbol{z}\in S_1) \\
&\quad + \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{add}}(\tilde{\boldsymbol{x}},\beta_l,\beta_t)}(f(\boldsymbol{z})=y_A \ \wedge\ \boldsymbol{z}\in S_2) \\
&= 0 + K\cdot\mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{add}}(\boldsymbol{x},\beta_l,\beta_t)}(f(\boldsymbol{z})=y_A \ \wedge\ \boldsymbol{z}\in S_2) \\
&= K\cdot[p_A - \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{add}}(\boldsymbol{x},\beta_l,\beta_t)}(f(\boldsymbol{z})=y_A \ \wedge\ \boldsymbol{z}\in S_1)] \\
&\geq K\cdot[p_A - \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{add}}(\boldsymbol{x},\beta_l,\beta_t)}(\boldsymbol{z}\in S_1)] \\
&= K\cdot[p_A - 1 + \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{add}}(\boldsymbol{x},\beta_l,\beta_t)}(\boldsymbol{z}\in S_2)] \\
&= K\cdot[p_A - 1 + 1/K] \\
&\geq K(\underline{p_A} - 1) + 1. \qquad (5)
\end{aligned}$$

Finally, we can get Equation (4) by solving the inequality that the lower bound of $\tilde{p}_A$ is not less than $1/2$. □

## 4.3 Robustness Region Derivation

When deriving the certified robustness regions in Lemma 1 and Lemma 2, we establish the probability distributions of the smoothing samples generated by the discrete and additive smoothing mechanisms, respectively. In this section, we combine these two probability distributions to derive the certified robustness region against both additive and discrete perturbations.

**Theorem 1.** *Consider a pair of traffic flows $\boldsymbol{x},\tilde{\boldsymbol{x}}\in\mathcal{X}$, where $\boldsymbol{x}$ contains $n$ packets with packet length vector $\boldsymbol{l}=(l_1,l_2,\ldots,l_n)$ and inter-arrival time vector $\boldsymbol{t}=(t_1,t_2,\ldots,t_n)$. $\boldsymbol{x}$ can be perturbed into $\tilde{\boldsymbol{x}}$ by two steps: (i) add non-negative noise vectors $\boldsymbol{\delta}^l=(\delta_1^l,\delta_2^l,\ldots,\delta_n^l)$ and $\boldsymbol{\delta}^t=(\delta_1^t,\delta_2^t,\ldots,\delta_n^t)$ to $\boldsymbol{l}$ and $\boldsymbol{t}$, respectively; (ii) insert $n^{ins}$ packets, substitute $n^{sub}$ packets and delete $n^{del}$ packets. Let $\psi^{int}(\boldsymbol{x},\beta_l,\beta_t,d):\mathcal{X}\times\mathbb{R}^+\times\mathbb{R}^+\times\mathbb{Z}^+\to\mathcal{Z}$ be the smoothing function that generates smoothing samples from $\boldsymbol{x}$ by two steps: (i) randomly selects $d$ packets while preserving their original orders ($d\leq n$); (ii) for every selected packet $x_i$, add random variables $\varepsilon_i^l \overset{i.i.d.}{\sim} \mathrm{Exp}(\beta_l^{-1})$ and $\varepsilon_i^t \overset{i.i.d.}{\sim} \mathrm{Exp}(\beta_t^{-1})$ to $l_i$ and $t_i$, respectively. Define the smoothed classifier $g^{int}$ as in Equation (1). Suppose $y_A\in\mathcal{Y}$ and $\underline{p_A}\in[1/2,1]$ satisfy $g^{int}(\boldsymbol{x})=y_A$ and $p_A\geq\underline{p_A}\geq 1/2$, then we have $g^{int}(\tilde{\boldsymbol{x}})=g^{int}(\boldsymbol{x})=y_A$ if:*

$$\begin{cases}
\sum_{i=1}^{d}\dfrac{\beta_l+\beta_t}{\beta_l}\cdot\bar{\delta}_i^l + \dfrac{\beta_l+\beta_t}{\beta_t}\cdot\bar{\delta}_i^t \leq r_*^{add}, \\
r_*^{add} = (\beta_l+\beta_t)\cdot[\log(P_1)-\log(P_2)], \\
P_1 = 1 - C^d_{n+n^{ins}-n^{del}}/2C^d_n, \\
P_2 = 2 - \underline{p_A} - C^d_{n-n^{sub}-n^{del}}/C^d_n,
\end{cases} \qquad (6)$$

*where $(\bar{\delta}_1^l,\bar{\delta}_2^l,\ldots,\bar{\delta}_n^l)$ and $(\bar{\delta}_1^t,\bar{\delta}_2^t,\ldots,\bar{\delta}_n^t)$ are obtained by sorting $\boldsymbol{\delta}^l$ and $\boldsymbol{\delta}^t$ in descending order of $\delta_i^l/\beta_l+\delta_i^t/\beta_t$, respectively.*

*Proof.* Let $X=\{x_1,x_2,\ldots,x_n\}$ be the set of packets in flow $\boldsymbol{x}$, the packets in $\tilde{\boldsymbol{x}}$ can be divided into three categories: (i) packets $\{\tilde{x}_1^{\mathsf{add}},\ldots,\tilde{x}_{n-n^{\mathsf{sub}}-n^{\mathsf{del}}}^{\mathsf{add}}\}$ obtained by adding length and time noise to a group of original packets $V=\{v_1,\ldots,v_{n-n^{\mathsf{sub}}-n^{\mathsf{del}}}\}$ in $\boldsymbol{x}$; (ii) packets $\{\tilde{x}_1^{\mathsf{sub}},\ldots,\tilde{x}_{n^{\mathsf{sub}}}^{\mathsf{sub}}\}$ obtained by substituting a group of packets in $X-V$; (iii) packets $\{\tilde{x}_1^{\mathsf{ins}},\ldots,\tilde{x}_{n^{\mathsf{ins}}}^{\mathsf{ins}}\}$ obtained through insertion.

Let $S$ be the set of all possible smoothing samples generated by $\psi^{\mathsf{int}}(\boldsymbol{x},\beta_l,\beta_t,d)$, for every $\boldsymbol{s}\in S$, $\boldsymbol{x}$ can be perturbed into $\boldsymbol{s}$ by two steps: (i) generate a flow $\boldsymbol{s}'$ by randomly selecting $d$ packets $(x_1^{\boldsymbol{s}'},x_2^{\boldsymbol{s}'},\ldots x_d^{\boldsymbol{s}'})$ from $\boldsymbol{x}$ while preserving their original orders; (ii) for every selected packet $x_i^{\boldsymbol{s}'}$, add a noise value generated from $\mathrm{Exp}(\beta_l^{-1})$ to $l_i^{\boldsymbol{s}'}$ and another noise value generated from $\mathrm{Exp}(\beta_t^{-1})$ to $t_i^{\boldsymbol{s}'}$, respectively. If $\{x_1^{\boldsymbol{s}'},x_2^{\boldsymbol{s}'},\ldots,x_d^{\boldsymbol{s}'}\}\not\subseteq V$, we have:

$$\frac{\mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{int}}(\tilde{\boldsymbol{x}},\beta_l,\beta_t,d)}(\boldsymbol{z}=\boldsymbol{s})}{\mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{int}}(\boldsymbol{x},\beta_l,\beta_t,d)}(\boldsymbol{z}=\boldsymbol{s})}=0.$$

Define $S^{\mathsf{sel}}=\{\boldsymbol{s}\in S:\{x_1^{\boldsymbol{s}'},x_2^{\boldsymbol{s}'},\ldots,x_d^{\boldsymbol{s}'}\}\subseteq V\}$. As illustrated in Figure 5, we partition $S^{\mathsf{sel}}$ into two sets of flows $S_1^{\mathsf{add}},S_2^{\mathsf{add}}$:

$$\begin{aligned}
S_1^{\mathsf{add}} &= \{\boldsymbol{s}\in S^{\mathsf{sel}}:\exists\, i\in[1,n],(l_i^{\boldsymbol{s}}-l_i^{\boldsymbol{s}'}<\delta_{i^*}^l)\vee(t_i^{\boldsymbol{s}}-t_i^{\boldsymbol{s}'}<\delta_{i^*}^t)\}, \\
S_2^{\mathsf{add}} &= \{\boldsymbol{s}\in S^{\mathsf{sel}}:\forall\, i\in[1,n],(l_i^{\boldsymbol{s}}-l_i^{\boldsymbol{s}'}\geq\delta_{i^*}^l)\wedge(t_i^{\boldsymbol{s}}-t_i^{\boldsymbol{s}'}\geq\delta_{i^*}^t)\}.
\end{aligned}$$

Based on the probability distributions of $\psi^{\mathsf{int}}(\boldsymbol{x},\beta_l,\beta_t,d)$ and $\psi^{\mathsf{int}}(\tilde{\boldsymbol{x}},\beta_l,\beta_t,d)$, for all $\boldsymbol{s}\in S_1^{\mathsf{add}}$, we have:

$$\frac{\mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{int}}(\tilde{\boldsymbol{x}},\beta_l,\beta_t,d)}(\boldsymbol{z}=\boldsymbol{s})}{\mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{int}}(\boldsymbol{x},\beta_l,\beta_t,d)}(\boldsymbol{z}=\boldsymbol{s})}=0.$$

For all $\boldsymbol{s}\in S_2^{\mathsf{add}}$, we have:

$$\frac{\mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{int}}(\tilde{\boldsymbol{x}},\beta_l,\beta_t,d)}(\boldsymbol{z}=\boldsymbol{s})}{\mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{int}}(\boldsymbol{x},\beta_l,\beta_t,d)}(\boldsymbol{z}=\boldsymbol{s})}=\frac{C_n^d}{C^d_{n+n^{\mathsf{ins}}-n^{\mathsf{del}}}}\cdot e^{\sum_{i=1}^{d}\frac{\delta_{i^*}^l}{\beta_l}+\frac{\delta_{i^*}^t}{\beta_t}},$$

where $i^*$ is the original index of packet $x_i^{\boldsymbol{s}'}$ in $(x_1,x_2,\ldots,x_n)$.

Define $p_A^{\mathsf{sel}} = \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{int}}(\boldsymbol{x},\beta_l,\beta_t,d)}(f(\boldsymbol{z}) = y_A \wedge \boldsymbol{z} \in S^{\mathsf{sel}})$, similar to Equation (3), the lower bound of $p_A^{\mathsf{sel}}$ can be derived as follows:

$$\begin{aligned} p_A^{\mathsf{sel}} &= p_A - \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{int}}(\boldsymbol{x},\beta_l,\beta_t,d)}(f(\boldsymbol{z}) = y_A \wedge \boldsymbol{z} \notin S^{\mathsf{sel}}) \\ &\geq \underline{p_A} - \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{int}}(\boldsymbol{x},\beta_l,\beta_t,d)}(\boldsymbol{z} \notin S^{\mathsf{sel}}) \\ &= \underline{p_A} - 1 + C^d_{n-n^{\mathsf{sub}}-n^{\mathsf{del}}}/C^d_n. \end{aligned} \tag{7}$$

Define $\tilde{p}_A^{\mathsf{sel}} = \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{int}}(\tilde{\boldsymbol{x}},\beta_l,\beta_t,d)}(f(\boldsymbol{z}) = y_A \wedge \boldsymbol{z} \in S^{\mathsf{sel}})$. Assume that $(\overline{\delta}^l_1, \overline{\delta}^l_2, \ldots, \overline{\delta}^l_n)$ and $(\overline{\delta}^t_1, \overline{\delta}^t_2, \ldots, \overline{\delta}^t_n)$ are obtained by sorting $\boldsymbol{\delta}^l$ and $\boldsymbol{\delta}^t$ in descending order of $\delta^l_i/\beta_l + \delta^t_i/\beta_t$, respectively. Let $K = C^d_n/C^d_{n+n^{\mathsf{ins}}-n^{\mathsf{del}}} \cdot e^{\sum_{i=1}^d \overline{\delta}^l_i/\beta_l + \overline{\delta}^t_i/\beta_t}$, similar to Equation (5), the lower bound of $\tilde{p}_A^{\mathsf{sel}}$ can be derived as follows (see § A for detailed proof):

$$\tilde{p}_A^{\mathsf{sel}} \geq K(p_A^{\mathsf{sel}} - 1) + C^d_n/C^d_{n+n^{\mathsf{ins}}-n^{\mathsf{del}}}. \tag{8}$$

With the combination of $\tilde{p}_A \geq \tilde{p}_A^{\mathsf{sel}}$, Equation (7) and Equation (8), the lower bound of $\tilde{p}_A$ can be derived as follows:

$$\begin{aligned} &\tilde{p}_A \geq \tilde{p}_A^{\mathsf{sel}} \geq C^d_n/C^d_{n+n^{\mathsf{ins}}-n^{\mathsf{del}}} \cdot \\ &[(\underline{p_A} - 2 + \frac{C^d_{n-n^{\mathsf{sub}}-n^{\mathsf{del}}}}{C^d_n}) \cdot e^{\sum_{i=1}^d \overline{\delta}^l_i/\beta_l + \overline{\delta}^t_i/\beta_t} + 1]. \end{aligned}$$

Finally, we can get Equations (6) by solving the inequality that the lower bound of $\tilde{p}_A$ is not less than $1/2$. □

Equations (6) specify CertTA's robustness region when the adversary simultaneously applies additive perturbations (*i.e.,* packet length padding and timing delays) and discrete perturbations (*i.e.,* packet insertion, substitution and deletion). Given the number of inserted, substituted and deleted packets $n^{\mathsf{ins}}, n^{\mathsf{sub}}$ and $n^{\mathsf{del}}$, we can calculate $P_1, P_2$ and subsequently obtain the corresponding additive robustness radius $r_*^{\mathsf{add}}$. The overall attack intensity is determined by the co-function of additive and discrete perturbations. As $n^{\mathsf{ins}}, n^{\mathsf{sub}}$ and $n^{\mathsf{del}}$ increase, the corresponding additive robustness radius $r_*^{\mathsf{add}}$ becomes smaller. In contrast to certification methods that address additive perturbations or discrete perturbations separately, the certified robustness region offered by CertTA is more aligned with the multi-modal adversarial perturbations in traffic analysis.

Moreover, the robustness region provided by CertTA exhibits several critical advantages in countering multi-modal adversarial perturbations. Specifically, in inequality

$$\sum_{i=1}^d \frac{\beta_l + \beta_t}{\beta_l} \cdot \overline{\delta}^l_i + \frac{\beta_l + \beta_t}{\beta_t} \cdot \overline{\delta}^t_i \leq r_*^{\mathsf{add}}$$

of Equations (6), we directly map the deviation values introduced by additive perturbations (*i.e.,* the items being summed on the left side of the inequality) to original packets in the flow. Therefore, when discrete perturbations (*e.g.,* packet insertion) result in the displacement of the packet sequence, the effectiveness of our additive robustness radius $r_*^{\mathsf{add}}$ will not be diminished. Further, since we only sum the largest $d$ deviation values to compare with $r_*^{\mathsf{add}}$, rather than the total $n$ deviation values, the issue of "curse of dimensionality" is alleviated. Consequently, we only need to consider $d$ packets when deriving the robustness region, regardless of the number of packets manipulated by attackers when applying additive perturbations. We also discuss the extension of CertTA to include new types of perturbations in § C.

Table 3: Traffic analysis models and adversarial attacks used in our evaluations.

| **Traffic Analysis Models** | Learning Algorithm | Flow Representation |
|---|---|---|
| kFP [11]<br>Kitsune [27]⋆ | ML-based<br>DL-based | flow statistics |
| Whisper [8]⋆<br>DFNet [38] | ML-based<br>DL-based | raw flow sequences |
| YaTC [55]<br>TrafficFormer [56] | Transformer-based<br>Transformer-based | raw bytes |
| **Adversarial Attacks** | Optimization Algorithm | Perturbation Operation |
| Blanket [29]<br>Amoeba [26]†<br>Prism [22] | GAN-based<br>RL-based<br>Explicit Modeling | insertion, padding, delaying |

⋆ Kitsune and Whisper are unsupervised models designed for anomaly detection. We extend these two models to supervised versions for multi-class classification.
† We use CertTA-certified models as the targeted models for Amoeba, making Amoeba an adaptive attack reacting to CertTA's defense.

## 5 Evaluation

We evaluate CertTA extensively to demonstrate:

- When faced with multi-modal adversarial attacks, CertTA outperforms the SOTA approaches significantly in both effectiveness and generality (§ 5.2). Across all six learning models, CertTA provides consistently high robustness guarantees against all three categories of adversarial attacks, while existing approaches have very limited applicability.
- We demonstrate a synergistic integration between CertTA and anomaly detection systems that creates a fundamental dilemma for the attacker (§ 5.3).
- We also evaluate the moving pieces in CertTA's design and application cases of CertTA (§ 5.4).

### 5.1 Experiment Setup

**Traffic Analysis Models and Adversarial Attacks.** We evaluate the performance of CertTA using six traffic analysis models against three types of multi-modal adversarial attacks. As summarized in Table 3, the six traffic analysis models use different flow representations and learning algorithms. The three categories of adversarial attacks employ Generative Adversarial Network (GAN)-based, Reinforcement Learning (RL)-based, and explicit modeling-based optimization algorithms,

Table 4: Setting of smoothing hyper-parameters.

| Datasets | CICDOH20 | | | | | | | TIISSRC23 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | VRS | BARS | | RS-Del | CertTA | | | VRS | BARS | | RS-Del | CertTA | | |
| Hyper-param. | $\sigma$ | $\lambda$ | $H_f$ | $p^{\text{del}}$ | $\beta_l$ | $\beta_t$ | $d\star$ | $\sigma$ | $\lambda$ | $H_f$ | $p^{\text{del}}$ | $\beta_l$ | $\beta_t$ | $d\star$ |
| kFP | 0.1 | NA | NA | 0.8 | 100 | 20ms | $\lceil 0.2n \rceil$ | 0.03 | NA | NA | 0.85 | 70 | 20ms | $\lceil 0.15n \rceil$ |
| Kitsune (sup.) | 0.1 | 0.001 | Gaussian | 0.8 | 100 | 20ms | $\lceil 0.2n \rceil$ | 0.03 | 0.001 | Gaussian | 0.8 | 50 | 10ms | $\lceil 0.2n \rceil$ |
| Whisper (sup.) | 80 | NA | NA | 0.8 | 100 | 20ms | $\lceil 0.2n \rceil$ | 20 | NA | NA | 0.8 | 30 | 10ms | $\lceil 0.2n \rceil$ |
| DF | 80 | 0.001 | Gaussian | 0.85 | 100 | 40ms | $\lceil 0.15n \rceil$ | 35 | 0.001 | Gaussian | 0.85 | 70 | 10ms | $\lceil 0.15n \rceil$ |
| YaTC | 80 | 0.01 | Gaussian | 0.9 | 200 | 40ms | $\lceil 0.1n \rceil$ | 80 | 0.01 | Gaussian | 0.9 | 200 | 40ms | $\lceil 0.1n \rceil$ |
| TrafficFormer | 80 | 0.01 | Gaussian | 0.9 | 200 | 40ms | $\lceil 0.1n \rceil$ | 80 | 0.01 | Gaussian | 0.9 | 200 | 40ms | $\lceil 0.1n \rceil$ |

$^\star$ Based on experimental experience, we set the smoothing hyper-parameter $d$ as a proportion of flow length $n$ for better performance.

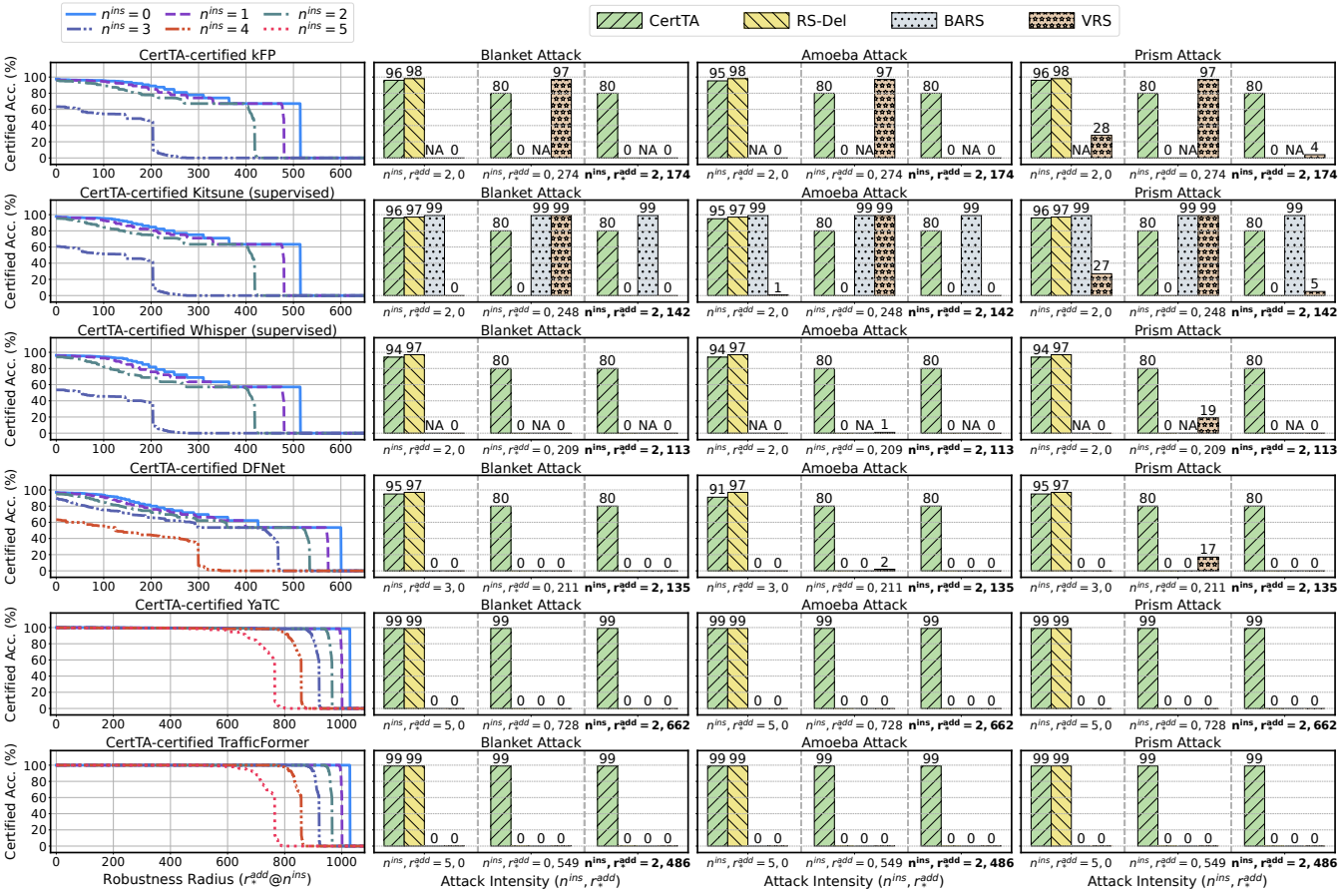


Figure 6: Certified accuracies of different certification methods against attacks Blanket, Amoeba and Prism (on CICDOH20).

respectively, to generate adversarial flows. For each experimental configuration, we can employ these attack methodologies to generate adversarial flows using either multi-modal perturbations (combining packet insertion, length padding, and timing delays) or single-modal perturbations. We use the open-source implementations of these approaches and additional details about these methods are deferred to § B.

**Datasets.** We evaluate CertTA on two traffic analysis datasets. (i) The CICDOH20 [28] dataset, which identifies tunneling traffic that utilize DNS over HTTPS (DoH). We collect the original pcap files for 4 classes (Benign, DNS2TCP, DNSCat2, Iodine) from this dataset. The number of flows in each class is 3000, 1000, 1000, 1000, respectively. (ii) The TIISSRC23 [12] dataset for intrusion detection. We collect the original pcap files for 5 classes (Benign-audio, Benign-video, BruteForce-http, BruteForce-telnet, Mirai) from this dataset. The number of flows in each class is 1200, 1200, 800, 800, 800, respectively. Each dataset is split into training set, validation set,

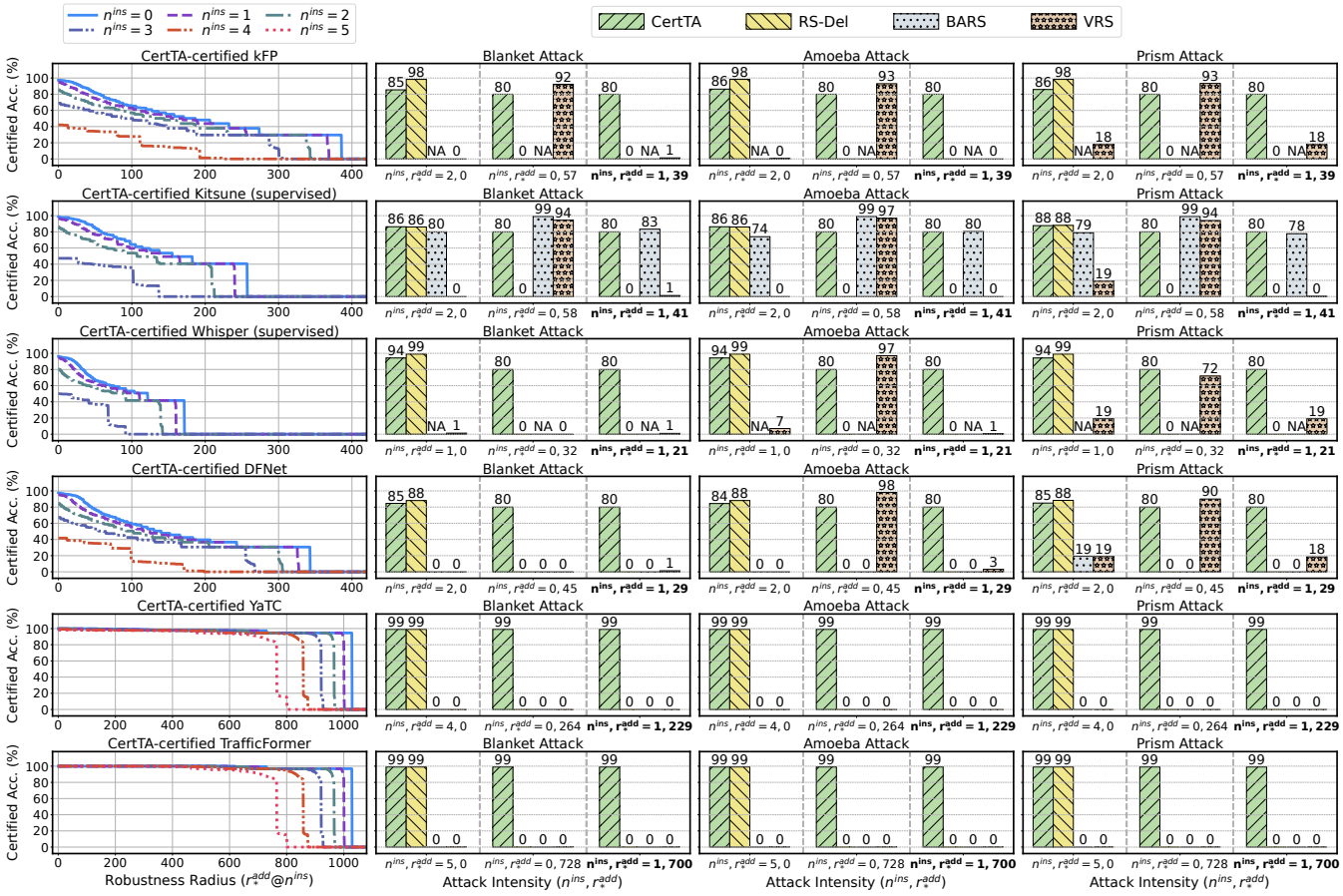


Figure 7: Certified accuracies of different certification methods against attacks Blanket, Amoeba and Prism (on TIISSRC23).

and test set with a ratio of 8:1:1.

**Baselines.** We compare CertTA with three baseline certification methods including VRS [5], BARS [39] and RS-Del [13]. VRS and BARS treat network flow features as an $1 \times D$ vector and provide a $\ell_2$-norm robustness radius against additive perturbations on these features, while RS-Del views a network flow as a discrete sequence of packets and provides robustness guarantees against discrete perturbations like packet insertion. The smoothing hyper-parameters of these methods across different traffic analysis models and datasets are listed in Table 4. We provide additional details regarding these methods and hyper-parameter settings in § B.

**Implementation.** We follow standard practice in randomized smoothing to obtain the smoothed classifier's prediction $y_A$ and the lower bound of corresponding probability $\underline{p_A}$ by Monte Carlo sampling. The number of smoothing samples is 1000 and the confidence level of Monte Carlo sampling is set to 0.999. When constructing a smoothed classifier $g$, we fine-tune the base traffic classifier $f$ by augmenting the training data with smoothing samples, which is a common technique in randomized smoothing for performance enhancement. Additional details about our testbed are deferred to § B.

## 5.2 Certified Robustness against Adversarial Attacks

In this experiment, we demonstrate that CertTA provides much stronger robustness guarantees against adversarial attacks than the SOTA approaches.

### 5.2.1 Robustness Regions of CertTA

We begin by evaluating CertTA's robustness region, which we quantify through certified accuracy measurements against various combinations of adversarial perturbations, including packet insertion, length padding, and timing delays. Our evaluation methodology fixes the number of inserted packets $n^{\text{ins}}$ and measures the system's certified accuracy across different additive robustness radius thresholds $r_*^{\text{add}}$. The certified accuracy $y$ represents the percentage of test flows that simultaneously maintain correct classification and achieve an additive robustness radius greater than the specified threshold $r_*^{\text{add}}$. Specifically, when considering adversarial flows generated with an attack intensity of $(n^{\text{ins}}, r_*^{\text{add}})$ - where $n^{\text{ins}}$ represents the number of inserted packets and $r_*^{\text{add}}$ bounds the magnitude of additive perturbations - this certified accuracy indicates the

Table 5: Classification performance of certified and non-certified traffic analysis models on clean traffic.

| Methods | Non-certified | | | VRS | | | BARS | | | RS-Del | | | CertTA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metrics | macro-$P$ | macro-$R$ | macro-$F_1$ | macro-$P$ | macro-$R$ | macro-$F_1$ | macro-$P$ | macro-$R$ | macro-$F_1$ | macro-$P$ | macro-$R$ | macro-$F_1$ | macro-$P$ | macro-$R$ | macro-$F_1$ |
| Encrypted Traffic Classification on DNS-over-HTTPS (CICDOH20) | | | | | | | | | | | | | | | |
| kFP | 0.998 | 0.998 | 0.997 | 0.980 | 0.980 | 0.980 | NA | NA | NA | 0.998 | 0.998 | 0.997 | 0.974 | 0.973 | 0.972 |
| Kitsune (sup.) | 0.998 | 0.998 | 0.997 | 0.993 | 0.992 | 0.992 | 0.998 | 0.998 | 0.997 | 0.993 | 0.993 | 0.992 | 0.974 | 0.973 | 0.972 |
| Whisper (sup.) | 0.995 | 0.995 | 0.995 | 0.959 | 0.958 | 0.957 | NA | NA | NA | 0.998 | 0.998 | 0.997 | 0.957 | 0.957 | 0.955 |
| DFNet | 0.995 | 0.995 | 0.995 | 0.973 | 0.973 | 0.973 | 0.984 | 0.983 | 0.982 | 0.998 | 0.998 | 0.997 | 0.972 | 0.970 | 0.970 |
| YaTC | 1.000 | 1.000 | 1.000 | 0.899 | 0.858 | 0.866 | 0.903 | 0.884 | 0.892 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| TrafficFormer | 1.000 | 1.000 | 1.000 | 0.892 | 0.864 | 0.878 | 0.916 | 0.895 | 0.907 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| Network Intrusion Detection (TIISSRC23) | | | | | | | | | | | | | | | |
| kFP | 0.998 | 0.998 | 0.997 | 0.959 | 0.937 | 0.942 | NA | NA | NA | 0.998 | 0.998 | 0.998 | 0.972 | 0.968 | 0.969 |
| Kitsune (sup.) | 0.989 | 0.984 | 0.986 | 0.985 | 0.982 | 0.983 | 1.000 | 1.000 | 1.000 | 0.988 | 0.986 | 0.987 | 0.986 | 0.982 | 0.983 |
| Whisper (sup.) | 1.000 | 1.000 | 1.000 | 0.979 | 0.976 | 0.977 | NA | NA | NA | 1.000 | 1.000 | 1.000 | 0.968 | 0.962 | 0.963 |
| DFNet | 0.991 | 0.993 | 0.992 | 0.990 | 0.989 | 0.990 | 0.995 | 0.997 | 0.996 | 0.996 | 0.996 | 0.996 | 0.976 | 0.973 | 0.974 |
| YaTC | 1.000 | 1.000 | 1.000 | 0.900 | 0.880 | 0.881 | 0.909 | 0.900 | 0.903 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| TrafficFormer | 1.000 | 1.000 | 1.000 | 0.903 | 0.895 | 0.898 | 0.915 | 0.911 | 0.912 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |

percentage of adversarial flows that CertTA can guarantee to classify correctly.

We present our quantitative results obtained from the CICDOH20 and TIISSRC23 datasets in the sub-figures on the leftmost column of Figure 6 and Figure 7, respectively. For each attack intensity configuration $(n^{\mathsf{ins}}, r_*^{\mathsf{add}})$, higher certified accuracy values indicate stronger robustness guarantees against adversarial flows. The results clearly reveal the disparity of robustness across different models. Among the models using the same flow representations, DFNet demonstrates superior robustness compared to Whisper (supervised) on both datasets, while kFP outperforms Kitsune (supervised) on the TIISSRC23 dataset. Notably, the Transformer-based models - YaTC and TrafficFormer - exhibit substantially higher robustness compared to other architectures.

### 5.2.2 Robustness Comparison Across Different Methods

In this section, we compare the effectiveness of the robustness regions derived by CertTA and three baseline methods (*i.e.,* VRS [5], BARS [39] and RS-Del [13]). Since the robustness regions of the baseline approaches are derived from single-modal perturbations, we cannot directly compare the robustness regions across different approaches. Instead, we create adversarial flows based on the attack methodology in Blanket [29], Amoeba [26] and Prism [22], and then use the certified accuracy on these adversarial flows to quantify the robustness guarantees of different approaches. To determine the attack intensities when generating adversarial flows, we set a lower bound threshold $T_{\mathsf{lower}}$ and select three attack intensities that are strong enough to degrade CertTA's certified accuracy to $T_{\mathsf{lower}}$. The attack intensity configurations are represented as tuples: $(n^{\mathsf{ins}}, 0)$ indicates insertion-only perturbations, $(0, r_*^{\mathsf{add}})$ represents additive-only perturbations, and $(n^{\mathsf{ins}}, r_*^{\mathsf{add}})$ denotes a combination of both perturbation types.

The experimental results from the CICDOH20 dataset and the TIISSRC23 dataset are shown in the right-side three columns of Figure 6 and Figure 7, respectively. Each row of sub-figures represents the results for one traffic analysis model. For the two transformer-based traffic analysis models YaTC and TrafficFormer, we set the threshold $T_{\mathsf{lower}}$ on CertTA's certified accuracy to 99%. For the remaining four models, $T_{\mathsf{lower}}$ is 80%.

Across all traffic analysis models, RS-Del can provide strong robustness guarantees against insertion-only perturbations. However, it fails to provide any certified accuracy against any of the three attacks with additive-noise-only perturbations or the combined perturbations. Further, the applicability of VRS is even more limited: it is only effective for certifying the traffic analysis models that use statistical flow features (*i.e.,* kFP and Kitsune (supervised)) against additive-noise-only perturbations. BARS is designed to improve VRS by applying random noises with customized scale and shape to different flow feature dimensions. However, such improvement is only applicable to the Kitsune model, since the flow representations or architectures used by the other five models are not compatible with BARS's noise shaping process.

In contrast, CertTA maintains consistently high certified performance guarantees against all categories of adversarial attacks across all model architectures. Notably, it is the only approach that provides effective robustness guarantees against multi-modal adversarial perturbations for both Transformer-based models and those utilizing raw flow sequences as input.

### 5.2.3 Performance on Clean Traffic

Certifiably robust models are trained on the adversarially manipulated datasets. Consequently, it is crucial to ensure that these certified models maintain their efficacy on the "clean dataset" devoid of adversarial perturbations. To this end, we evaluate the classification performance of certified and non-certified traffic analysis models on clean traffic using three metrics: macro averaging of precision ($P$), recall ($R$) and $F_1$-score. The experimental results are reported in Table 5. For models trained using statistical flow features and raw flow sequences, CertTA exhibits a slightly larger reduction in performance compared to the baseline certification techniques. This outcome is attributable to CertTA's strategy of aggres-

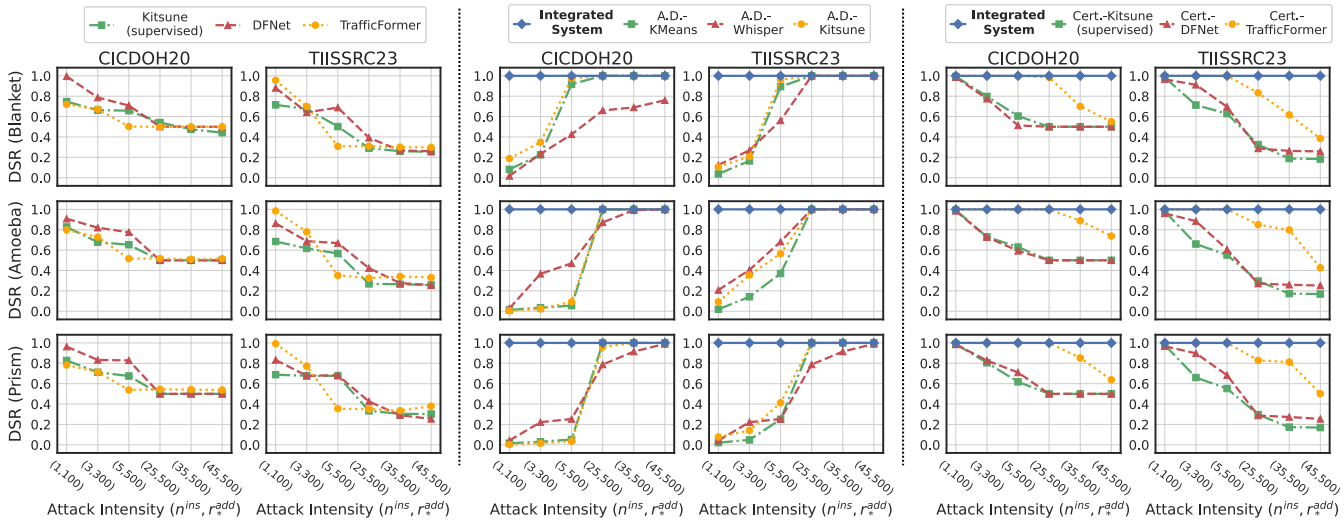


Figure 8: Defense Success Rate of integrated and standalone systems against adversarial attacks with different intensities. The left two columns represent models without defenses; the middle two columns represent "A.D." (anomaly detection) systems, and right two columns represent "Cert." (certified) traffic analysis models offered by CertTA.

sively incorporating both additive and discrete perturbations when generating smoothing samples, a design choice aimed at tolerating multi-modal adversarial attacks. Nevertheless, the performance degradation incurred by CertTA remains limited, manifesting as an average decrease of only 0.025 in the macro $F_1$-score across kFP, Kitsune (supervised), Whisper (supervised), and DFNet, relative to the performance of non-certified models. However, for Transformer-based models trained on raw byte sequences, CertTA achieves zero performance reduction, whereas both VRS and BARS cause non-trivial performance decreases. This discrepancy arises because VRS and BARS severely distort the byte-level information through the direct addition of numerical noises. Overall, these experimental results demonstrate that CertTA imposes very minimal performance reductions on clean traffic.

## 5.3 Integration with Anomaly Detection

**Experiment Motivation.** CertTA guarantees that any adversarial perturbations within the certified robustness regions will not disrupt the prediction of a smoothed traffic analysis model. In other words, an adversarial flow must carry non-trivial perturbations that are intensive enough to bypass CertTA. This, fortunately, will make it easier for anomaly detection systems (*e.g.,* [8–10, 21, 27]) to capture these adversarial flows. Therefore, *synergistically integrating an anomaly detector with CertTA can create a fundamental dilemma for the attacker:* stealth adversarial flows (*i.e.,* with small perturbations) which may bypass the anomaly detector are ineffective against CertTA; and the adversarial flows with significant perturbations which may exceed CertTA's certified robustness regions can be easily captured by the anomaly detector.

**Experiment Design.** In this experiment, we integrate an unsupervised anomaly detector Kitsune [27] with a CertTA-certified TrafficFormer to implement a two-phase defense system. We use attacks Blanket, Amoeba and Prism to generate adversarial flows with different levels of attack intensities and evaluate the Defense Success Rate (DSR) of our system against these adversarial flows. We compare our integrated system with 9 baselines: three traffic analysis models without defenses (*i.e.,* Kitsune (supervised), DFNet, TrafficFormer), three standalone anomaly detectors (*i.e.,* KMeans [23], Whisper [8], Kitsune [27]) and three standalone certified traffic analysis models (*i.e.,* Kitsune (supervised), DFNet, TrafficFormer). For all anomaly detectors used in this experiment, we adjust its detection threshold to ensure that the False Positive Rate on the original test dataset is less than 1%.

**Results.** The experimental results are reported in Figure 8. Clearly, the traffic analysis models without defenses achieve poor DSRs against adversarial flows with various intensities. The standalone anomaly detectors struggle to defend against low-intensity adversarial flows but are effective at identifying high-intensity adversarial flows as anomalies. On the contrary, the standalone certified traffic analysis models can identify low-intensity adversarial flows accurately but inevitably misclassify high-intensity adversarial flows. Through the synergistic combination of CertTA and anomaly detection, our integrated system exhibits consistently high DSRs against adversarial attacks across all attack intensities.

## 5.4 CertTA Deep Dive

**Certification Delay.** We measure the certification delays (*i.e.,* the average time to obtain the classification result and ro-

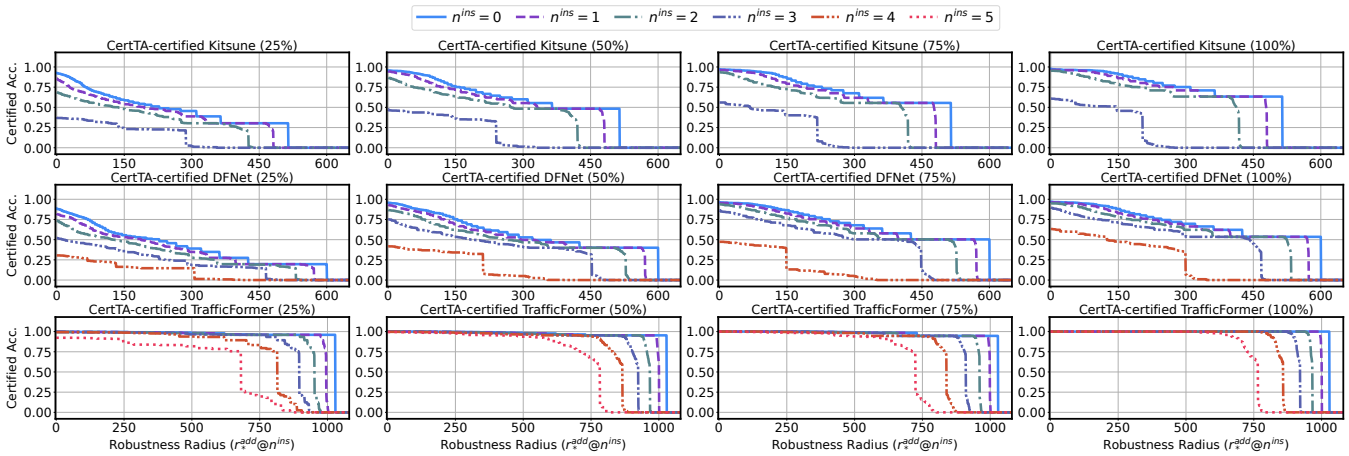


Figure 9: Certified accuracies of CertTA-certified traffic analysis models under truncated settings (on CICDOH20).

Table 6: Certification delay of different certification methods.

| Certification Methods | VRS | BARS | RS-Del | CertTA |
|---|---|---|---|---|
| kFP | 0.146s | NA | 0.224s | 0.218s |
| Kitsune (supervised) | 0.166s | 0.183s | 0.241s | 0.251s |
| Whisper (supervised) | 0.349s | NA | 0.316s | 0.342s |
| DFNet | 0.107s | 0.192s | 0.169s | 0.166s |
| YaTC | 0.533s | 0.537s | 0.708s | 0.731s |
| TrafficFormer | 3.401s | 3.410s | 3.573s | 3.648s |

Table 7: Macro-$F_1$ of CertTA-certified models on clean traffic under various truncated settings (on CICDOH20).

| Truncated Settings | 25% | 50% | 75% | 100% |
|---|---|---|---|---|
| Kitsune (supervised) | 0.9320 | 0.9553 | 0.9722 | 0.9722 |
| DFNet | 0.8895 | 0.9487 | 0.9578 | 0.9700 |
| TrafficFormer | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

bustness region for an input flow) induced by CertTA and baseline methods across six traffic analysis models. The experimental results using the CICDOH20 dataset are shown in Table 6. Since CertTA and RS-Del both operate on raw flow packets rather than extracted flow features, their certification delays are similar and slightly higher than those of VRS and BARS. One possible strategy to expedite the certification process is deriving certified classification results based on a subset of flow packets. To investigate this approach, we evaluate the certified accuracy and clean performance of three CertTA-certified models (*i.e.,* Kitsune (supervised), DFNet, TrafficFormer) when observing the first 25%, 50%, 75%, and 100% of the packets comprising a flow. The experimental results using the CICDOH20 dataset are shown in Figure 9 and Table 7. The results indicate that, generally, both Kitsune (supervised) and DFNet require no less than 50% of the packets to achieve performance comparable to that observed with

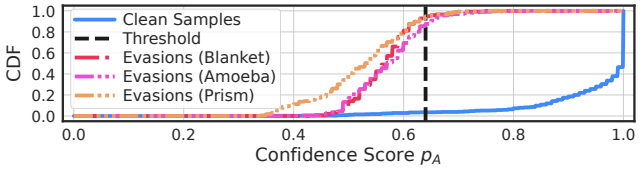


Figure 10: Evasion samples awareness by setting a threshold on CertTA's classification confidence (on CICDOH20).

complete flows, whereas TrafficFormer demonstrates to be more tolerant of such data truncation.

**Evasion Samples Awareness.** Given an input sample, the probability $p_A$ of the predicted class $y_A$ quantifies CertTA's classification confidence on this sample. By examining the magnitude of $p_A$, we can effectively recall the flows that have successfully deceived a certified model, which are referred to as evasion samples. This is because these evasion samples have different data distributions from clean samples in the original dataset, resulting in abnormally small confidence scores. To demonstrate this capability, we use attacks Blanket, Amoeba and Prism to generate adversarial flows from the CICDOH20 dataset, and evaluate these adversarial flows using the CertTA-certified DFNet model. With an attack intensity of $(n^{\text{ins}}, r_*^{\text{add}}) = (5, 500)$, the evasion success rate of Blanket, Amoeba and Prism reaches 48.7%, 37.2% and 29.0%, respectively. The distributions of the confidence score $p_A$ of these test flows are shown in Figure 10. By establishing a proper threshold on $p_A$, we can effectively recall 94.9%, 85.6%, 94.3% of the evasion samples generated by Blanket, Amoeba and Prism, respectively, while maintaining a low False Positive Rate of 2.0% on clean samples.

**Hyper-parameter Tuning.** We investigate how smoothing hyper-parameters and the number of smoothing samples affect the performance of CertTA. Specifically, we evaluate the performance of the CertTA-certified DFNet model under dif-

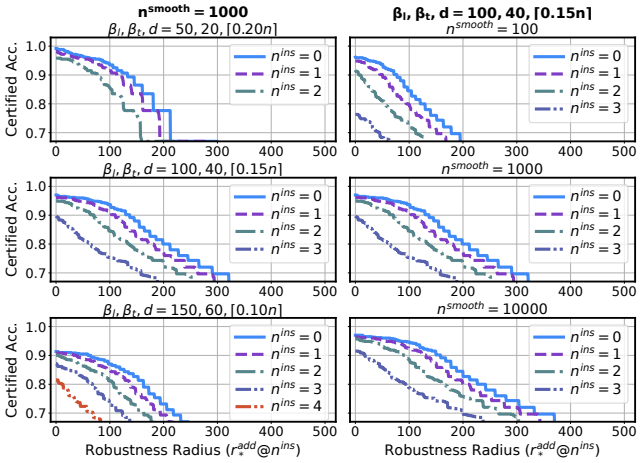


Figure 11: Performance of CertTA-certified DFNet under different paramter settings (on CICDOH20).

ferent parameter settings, using the CICDOH20 dataset. The experimental results are shown in Figure 11. In the sub-figures on the left column, we fix the number of smoothing samples at 1000 and quantify the robustness guarantees offered by CertTA using various smoothing hyper-parameters (*i.e.,* the scale parameters $\beta_l, \beta_t$ for Exponential noises and the number of selected packets $d$). The results indicate that in general, smaller parameters constrain the theoretical upper bound of CertTA's robustness region, while larger parameters decrease the classification accuracy of the smoothing samples. Empirically, we select larger smoothing hyper-parameters before experiencing significant accuracy losses for a traffic analysis model. In the right column of Figure 11, we fix the smoothing hyper-parameters and tune the number of smoothing samples. As the number of smoothing samples $n^{\text{smooth}}$ increases, CertTA-certified DFNet achieves better performance in robustness guarantees. This is because the estimation of $p_A$ through Monte Carlo sampling is more accurate. However, while the performance improvement diminishes with larger $n^{\text{smooth}}$, the inference overhead of smoothing samples will increase linearly. Considering the trade-off between the performance in robustness guarantees and the inference overhead, we choose 1000 as the number of smoothing samples in our experiments.

**Application Cases Discussed in BARS [39].** BARS [39] discussed several use cases of certified robustness in traffic analysis. In this section, we show that CertTA is also applicable to these use cases, and even provides more benefits in these cases. The use cases in BARS can be summarized into three categories. (i) BARS can be applied to defend against adversarial attacks, such as providing stronger robustness guarantees than VRS, reducing false alarms and defending against evasion attacks. Based on the experimental results in § 5.2.2, we demonstrate that CertTA outperforms BARS significantly in both the applicability over heterogeneous traffic analysis models and the certified accuracies against adversarial attacks. Moreover, we propose a novel use case in § 5.3 that integrates CertTA with anomaly detection, which creates a fundamental dilemma for the attacker. (ii) BARS can be applied to quantitatively evaluate robustness. Yet, the robustness metric offered by BARS is based on model-specific flow features. In contrast, CertTA provides unified robustness regions across heterogeneous models, as shown in § 5.2.1, ensuring easy robustness comparison among these models. (iii) BARS is capable of detecting and explaining evasion samples. In § 5.4, we have just demonstrated that CertTA can also be applied to detect evasion samples by examining the magnitude of the classification confidence.

## 6 Discussion and Related work

**Transformer with CertTA.** In recent years, the paradigm of using Transformer-based models to infer from raw packet bytes led to promising improvements in the accuracy of traffic analysis [24,48,55,56]. Yet, by simply inserting a few packets at the start of the flow, the adversary can introduce a significant change in the raw bytes input and undermine the performance of these models. This vulnerability can be effectively mitigated by CertTA. As demonstrated by our evaluation results in § 5, CertTA-certified Transformer-based models achieve promising performance in both accuracy and robustness.

**Empirical Robustness Enhancement.** To improve the robustness of traffic analysis models, current works mainly focus on methods like data augmentation [2, 14, 34, 44, 50], adopting more robust traffic representation [8, 37] or improving model designs [38]. CertTA is orthogonal to all these approaches. Given an empirically robust model, its certified robustness offered by CertTA is also improved. For instance, CertTA achieves better performance after fine-tuning the base classifier with the data augmentation of smoothing samples.

**Robustness Certification Methods.** Robustness certification methods for individual input sample can be roughly categorized into two types: deterministic and probabilistic. Deterministic methods [41, 45, 51] aim to solve the deterministic mapping from input variations to output variations, making them computationally expensive and model-dependent. Probabilistic methods like randomized smoothing [5,18,47] employ sampling techniques to offer certification for arbitrary model architectures efficiently. Due to their flexibility, we explore randomized smoothing based approaches in this paper.

**Network Anomaly Detection Systems.** As one of the most important techniques in security domains [4], anomaly detection has been widely adopted for network intrusion detection [8–10, 21, 27]. Based on unsupervised learning, these network anomaly detection systems are trained with normal traffic data to detect anomalies that deviate from the learned data distribution. Compared to supervised approaches, network anomaly detection systems do not rely on well-labeled datasets for training and generalize better on unknown threats

such as zero-day attacks. However, it is challenging for these systems to detect stealth attacks accurately while maintaining a low False Positive Rate [9, 10]. Fortunately, the robustness guarantees offered by CertTA are highly effective in defending against stealth attacks, which enables a synergistic integration with anomaly detection.

**Practical Deployment.** As model certification introduces extra overhead, one of the key challenges to deploy certified models in production is to efficiently identify the "problematic flows" that may require in-depth analysis by the certified model, while leaving other flows processed by typical, real-time models. Therefore, in production environments where the vast majority flows are benign, the certified model only processes small amount of traffic. One possible approach is to employ an ensemble of heterogeneous (non-certified) ML/DL/Transformer-based models for real-time traffic analysis, and only forward the flows with inconsistent classification results from these models to the certified model (the implication is that gradient/RL-based adversarial samples are crafted to evade certain targeted models and exhibit poor transferability in models with different architectures and representations [1, 53]). An additional benefit of this approach is to mitigate the tradeoff discussed in § 5.2.3, because the certified model processes only a very limited volume of clean traffic. We leave further investigation of this matter to future work.

## 7 Conclusion

In this paper, we present CertTA, a novel robustness certification methodology that significantly advances state-of-the-art in terms of both effectiveness and generality. CertTA is the first certification framework that establishes robustness guarantees against multi-modal adversarial perturbations. Meanwhile, CertTA is universally applicable to various heterogeneous traffic analysis models and provides unified metrics of model robustness. We provide rigorous mathematical construction regarding the robustness guarantees offered by CertTA. We implement a prototype of CertTA and extensively evaluate the prototype in various settings to quantify the advantages of CertTA over the SOTA approaches.

## 8 Ethics Considerations

We have carefully considered the ethical implications at every stage of our research, including the design, evaluation and publication. The design and publication of CertTA will contribute positively to the field. 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. All experiments were conducted within in our private testbed, ensuring that our research did not introduce any risks or ethical concerns related to experimenting in live systems or public networks.

## 9 Open Science

Our research artifacts associated with this work are available on Zenodo[2] and Github[3]. The source code of our CertTA prototype, along with 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.

## Acknowledgement

We thank our shepherd and the anonymous reviewers for their insightful feedback. The research is supported in part by the National Key R&D Program of China under Grant 2024YFB2906803, and National Natural Science Foundation of China (NSFC) under Grant 62472247 and 62425201. The corresponding author of this paper is Zhuotao Liu.

## References

[1] Nour Alhussien, Ahmed Aleroud, Abdullah Melhem, and Samer Y Khamaiseh. Constraining Adversarial Attacks on Network Intrusion Detection Systems: Transferability and Defense Analysis. *IEEE TNSM*, 2024.

[2] Alireza Bahramali, Ardavan Bozorgi, and Amir Houmansadr. Realistic Website Fingerprinting By Augmenting Network Traces. In *ACM CCS*, 2023.

[3] Avrim Blum, Travis Dick, Naren Manoj, and Hongyang Zhang. Random Smoothing Might be Unable to Certify $\ell_\infty$ Robustness for High-Dimensional Images. *Journal of machine learning research*, 2020.

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly Detection: A Survey. *ACM computing surveys (CSUR)*, 2009.

[5] Jeremy Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified Adversarial Robustness via Randomized Smoothing. In *ICML*, 2019.

[6] Andrew C Cullen, Paul Montague, Shijie Liu, Sarah M Erfani, and Benjamin IP Rubinstein. It's Simplex! Disaggregating Measures to Improve Certified Robustness. In *IEEE S&P*, 2024.

[7] Alec F. Diallo and Paul Patras. Adaptive Clustering-based Malicious Traffic Classification at the Network Edge. In *IEEE INFOCOM*, 2021.

[2] Available at https://doi.org/10.5281/zenodo.15580292

[3] Available at https://github.com/InspiringGroup-Lab/CertTA

[8] Chuanpu Fu, Qi Li, Meng Shen, and Ke Xu. Realtime Robust Malicious Traffic Detection via Frequency Domain Analysis. In *ACM CCS*, 2021.

[9] Chuanpu Fu, Qi Li, Ke Xu, and Jianping Wu. Point Cloud Analysis for ML-based Malicious Traffic Detection: Reducing Majorities of False Positive Alarms. In *ACM CCS*, 2023.

[10] Dongqi Han, Zhiliang Wang, Wenqi Chen, Kai Wang, Rui Yu, Su Wang, Han Zhang, Zhihua Wang, Minghui Jin, Jiahai Yang, Xingang Shi, and Xia Yin. Anomaly Detection in the Open World: Normality Shift Detection, Explanation, and Adaptation. In *NDSS*, 2023.

[11] Jamie Hayes and George Danezis. k-Fingerprinting: A Robust Scalable Website Fingerprinting Technique. In *USENIX Security*, 2016.

[12] Dania Herzalla, Willian T Lunardi, and Martin Andreoni. TII-SSRC-23 Dataset: Typological Exploration of Diverse Traffic Patterns for Intrusion Detection. *IEEE Access*, 2023.

[13] Zhuoqun Huang, Neil G. Marchant, Keane Lucas, Lujo Bauer, Olga Ohrimenko, and Benjamin I. P. Rubinstein. RS-Del: Edit Distance Robustness Certificates for Sequence Classifiers via Randomized Deletion. In *NeurIPS*, 2023.

[14] Steve T. K. Jan, Qingying Hao, Tianrui Hu, Jiameng Pu, Sonal Oswal, Gang Wang, and Bimal Viswanath. Throwing Darts in the Dark? Detecting Bots with Limited Data using Neural Data Augmentation. In *IEEE S&P*, 2020.

[15] Meiyi Jiang, Baojiang Cui, Junsong Fu, Tao Wang, and Ziqi Wang. KimeraPAD: A Novel Low-Overhead Real-Time Defense Against Website Fingerprinting Attacks Based On Deep Reinforcement Learning. *IEEE TNSM*, 2024.

[16] Meiyi Jiang, Baojiang Cui, Junsong Fu, Tao Wang, Lu Yao, and Bharat K Bhargava. RUDOLF: An Efficient and Adaptive Defense Approach Against Website Fingerprinting Attacks Based on Soft Actor-Critic Algorithm. *IEEE TIFS*, 2024.

[17] Aounon Kumar, Alexander Levine, Tom Goldstein, and Soheil Feizi. Curse of Dimensionality on Randomized Smoothing for Certifiable Robustness. In *ICML*, 2020.

[18] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified Robustness to Adversarial Examples with Differential Privacy. In *IEEE S&P*, 2019.

[19] Mathias Lécuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified Robustness to Adversarial Examples with Differential Privacy. In *IEEE S&P*, 2019.

[20] Linyi Li, Tao Xie, and Bo Li. Sok: Certified Robustness for Deep Neural Networks. In *IEEE S&P*, 2023.

[21] Peiyang Li, Ye Wang, Qi Li, Zhuotao Liu, Ke Xu, Ju Ren, Zhiying Liu, and Ruilin Lin. Learning from Limited Heterogeneous Training Data: Meta-Learning for Unsupervised Zero-Day Web Attack Detection across Web Domains. In *ACM CCS*, 2023.

[22] Wenhao Li, Xiao-Yu Zhang, Huaifeng Bao, Binbin Yang, Zhaoxuan Li, Haichao Shi, and Qiang Wang. Prism: Real-Time Privacy Protection Against Temporal Network Traffic Analyzers. *IEEE TIFS*, 2023.

[23] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The Global K-means Clustering Algorithm. *Pattern recognition*, 2003.

[24] Xinjie Lin, Gang Xiong, Gaopeng Gou, Zhen Li, Junzheng Shi, and Jing Yu. Et-bert: A Contextualized Datagram Representation with Pre-training Transformers for Encrypted Traffic Classification. In *ACM WWW*, 2022.

[25] Chang Liu, Longtao He, Gang Xiong, Zigang Cao, and Zhen Li. Fs-Net: A Flow Sequence Network for Encrypted Traffic Classification. In *IEEE INFOCOM*, 2019.

[26] Haoyu Liu, Alec F Diallo, and Paul Patras. Amoeba: Circumventing ML-supported Network Censorship via Adversarial Reinforcement Learning. *Proceedings of the ACM on Networking*, 2023.

[27] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. In *NDSS*, 2018.

[28] Mohammadreza MontazeriShatoori, Logan Davidson, Gurdip Kaur, and Arash Habibi Lashkari. Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic. In *DASC/PiCom/CBDCom/CyberSciTech*, 2020.

[29] Milad Nasr, Alireza Bahramali, and Amir Houmansadr. Defeating DNN-Based Traffic Analysis Systems in Real-Time with Blind Adversarial Perturbations. In *USENIX Security*, 2021.

[30] Andriy Panchenko, Fabian Lanze, Jan Pennekamp, Thomas Engel, Andreas Zinnen, Martin Henze, and Klaus Wehrle. Website Fingerprinting at Internet Scale. In *NDSS*, 2016.

[31] Lingfeng Peng, Xiaohui Xie, Sijiang Huang, Ziyi Wang, and Yong Cui. PTU: Pre-trained Model for Network Traffic Understanding. In *IEEE ICNP*, 2024.

[32] Fabio Pierazzi, Feargus Pendlebury, Jacopo Cortellazzi, and Lorenzo Cavallaro. Intriguing Properties of Adversarial ML Attacks in the Problem Space. In *IEEE S&P*, 2020.

[33] Litao Qiao, Bang Wu, Heng Li, Cuiying Gao, Wei Yuan, and Xiapu Luo. Trace-agnostic and Adversarial Training-resilient Website Fingerprinting Defense. In *IEEE INFOCOM*, 2024.

[34] Yuqi Qing, Qilei Yin, Xinhao Deng, Yihao Chen, Zhuotao Liu, Kun Sun, Ke Xu, Jia Zhang, and Qi Li. Low-Quality Training Data Only? A Robust Framework for Detecting Encrypted Malicious Network Traffic. In *NDSS*, 2024.

[35] Jian Qu, Xiaobo Ma, Jianfeng Li, Xiapu Luo, Lei Xue, Junjie Zhang, Zhenhua Li, Li Feng, and Xiaohong Guan. An Input-Agnostic Hierarchical Deep Learning Framework for Traffic Fingerprinting. In *USENIX Security*, 2023.

[36] Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J Zico Kolter. Denoised Smoothing: A Provable Defense for Pretrained Classifiers. *NeurIPS*, 2020.

[37] Meng Shen, Kexin Ji, Zhenbo Gao, Qi Li, Liehuang Zhu, and Ke Xu. Subverting Website Fingerprinting Defenses with Robust Traffic Representation. In *USENIX Security*, 2023.

[38] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning. In *ACM CCS*, 2018.

[39] Kai Wang, Zhiliang Wang, Dongqi Han, Wenqi Chen, Jiahai Yang, Xingang Shi, and Xia Yin. BARS: Local Robustness Certification for Deep Learning based Traffic Analysis Systems. In *NDSS*, 2023.

[40] Minxiao Wang, Ning Yang, Nicolas J Forcade-Perkins, and Ning Weng. ProGen: Projection-based Adversarial Attack Generation against Network Intrusion Detection. *IEEE TIFS*, 2024.

[41] Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J. Zico Kolter. Beta-crown: Efficient Bound Propagation with Per-neuron Split Constraints for Neural Network Robustness Verification. In *NeurIPS*, 2021.

[42] Chong Xiang, Alexander Valtchanov, Saeed Mahloujifar, and Prateek Mittal. Objectseeker: Certifiably Robust Object Detection against Patch Hiding Attacks via Patch-agnostic Masking. In *IEEE S&P*, 2023.

[43] Renjie Xie, Jiahao Cao, Yuxi Zhu, Yixiang Zhang, Yi He, Hanyi Peng, Yixiao Wang, Mingwei Xu, Kun Sun, Enhuan Dong, et al. Cactus: Obfuscating Bidirectional Encrypted TCP Traffic at Client Side. *IEEE TIFS*, 2024.

[44] Renjie Xie, Yixiao Wang, Jiahao Cao, Enhuan Dong, Mingwei Xu, Kun Sun, Qi Li, Licheng Shen, and Menghao Zhang. Rosetta: Enabling Robust TLS Encrypted Traffic Classification in Diverse Network Environments with TCP-aware Traffic Augmentation. In *USENIX Security*, 2023.

[45] Kaidi Xu, Huan Zhang, Shiqi Wang, Yihan Wang, Suman Jana, Xue Lin, and Cho-Jui Hsieh. Fast and Complete: Enabling Complete Neural Network Verification with Rapid and Massively Parallel Incomplete Verifiers. In *ICLR*, 2020.

[46] Jinzhu Yan, Haotian Xu, Zhuotao Liu, Qi Li, Ke Xu, Mingwei Xu, and Jianping Wu. Brain-on-Switch: Towards Advanced Intelligent Network Data Plane via NN-Driven Traffic Analysis at Line-Speed. In *USENIX NSDI*, 2024.

[47] Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized Smoothing of All Shapes and Sizes. In *ICML*, 2020.

[48] Luming Yang, Lin Liu, Jun-Jie Huang, Zhuotao Liu, Shiyu Liang, Shaojing Fu, and Yongjun Wang. MM4flow: A Pre-trained Multi-modal Model for Versatile Network Traffic Analysis. In *ACM CCS*, 2025.

[49] Mao Ye, Chengyue Gong, and Qiang Liu. SAFER: A Structure-free Approach for Certified Robustness to Adversarial Word Substitutions. In *ACL*, 2020.

[50] Yucheng Yin, Zinan Lin, Minhao Jin, Giulia Fanti, and Vyas Sekar. Practical GAN-based Synthetic IP Header Trace Generation using NetShare. In *ACM SIGCOMM*, 2022.

[51] Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane S. Boning, and Cho-Jui Hsieh. Towards Stable and Efficient Training of Verifiably Robust Neural Networks. In *ICLR*, 2020.

[52] Xinyu Zhang, Hanbin Hong, Yuan Hong, Peng Huang, Binghui Wang, Zhongjie Ba, and Kui Ren. Text-CRS: A Generalized Certified Robustness Framework against Textual Adversarial Attacks. In *IEEE S&P*, 2024.

[53] Yechao Zhang, Shengshan Hu, Leo Yu Zhang, Junyu Shi, Minghui Li, Xiaogeng Liu, Wei Wan, and Hai Jin. Why does Little Robustness Help? A Further Step Towards

Understanding Adversarial Transferability. In *IEEE S&P*, 2024.

[54] Haiteng Zhao, Chang Ma, Xinshuai Dong, Anh Tuan Luu, Zhi-Hong Deng, and Hanwang Zhang. Certified Robustness against Natural Language Attacks by Causal Intervention. In *ICML*, 2022.

[55] Ruijie Zhao, Mingwei Zhan, Xianwen Deng, Yanhao Wang, Yijun Wang, Guan Gui, and Zhi Xue. Yet another Traffic Classifier: A Masked Autoencoder based Traffic Transformer with Multi-level Flow Representation. In *AAAI*, 2023.

[56] Guangmeng Zhou, Xiongwen Guo, Zhuotao Liu, Tong Li, Qi Li, and Ke Xu. TrafficFormer: An Efficient Pre-trained Model for Traffic Data. In *IEEE S&P*, 2024.

[57] Guangmeng Zhou, Zhuotao Liu, Chuanpu Fu, Qi Li, and Ke Xu. An Efficient Design of Intelligent Network Data Plane. In *USENIX Security*, 2023.

## A Supplementary Proof of Theorem 1

In this section, we present the detailed derivations for the lower bound of $\tilde{p}_A^{\mathsf{sel}}$ in § 4.3, *i.e.,*

$$\tilde{p}_A^{\mathsf{sel}} \geq K(p_A^{\mathsf{sel}} - 1) + C_n^d / C_{n+n^{\mathsf{ins}}-n^{\mathsf{del}}}^d,$$

where $K = C_n^d / C_{n+n^{\mathsf{ins}}-n^{\mathsf{del}}}^d \cdot e^{\sum_{i=1}^d \overline{\delta}_i^l/\beta_l + \overline{\delta}_i^t/\beta_t}$.

*Proof.* Given any $\boldsymbol{u} = (x_1^{\boldsymbol{u}}, x_2^{\boldsymbol{u}}, \ldots, x_d^{\boldsymbol{u}}) \in \mathcal{X}$ satisfies $\{x_1^{\boldsymbol{u}}, x_2^{\boldsymbol{u}}, \ldots, x_d^{\boldsymbol{u}}\} \subseteq V$, we define $\dot{S}^{\mathsf{sel}} \subseteq S^{\mathsf{sel}}$ as follows:

$$\dot{S}^{\mathsf{sel}} = \{\boldsymbol{s} \in S^{\mathsf{sel}} : \forall\, i \in [1,n], x_i^{\boldsymbol{s}'} = x_i^{\boldsymbol{u}}\}.$$

For any $\boldsymbol{s} \in \dot{S}^{\mathsf{sel}}$, we have $(x_1^{\boldsymbol{s}'}, x_2^{\boldsymbol{s}'}, \ldots, x_d^{\boldsymbol{s}'}) = (x_1^{\boldsymbol{u}}, x_2^{\boldsymbol{u}}, \ldots, x_d^{\boldsymbol{u}})$. Therefore, the original index $i^*$ of packet $x_i^{\boldsymbol{s}'}$ in $(x_1, x_2, \ldots, x_n)$ remains unchanged. Subsequently, we divide $\dot{S}^{\mathsf{sel}}$ into two sets of flows $\dot{S}_1^{\mathsf{add}} \subseteq S_1^{\mathsf{add}}$ and $\dot{S}_2^{\mathsf{add}} \subseteq S_2^{\mathsf{add}}$ as follows:

$$\dot{S}_1^{\mathsf{add}} = \{\boldsymbol{s} \in \dot{S}^{\mathsf{sel}} : \exists\, i \in [1,n], (l_i^{\boldsymbol{s}} - l_i^{\boldsymbol{s}'} < \delta_{i^*}^l) \vee (t_i^{\boldsymbol{s}} - t_i^{\boldsymbol{s}'} < \delta_{i^*}^t)\},$$
$$\dot{S}_2^{\mathsf{add}} = \{\boldsymbol{s} \in \dot{S}^{\mathsf{sel}} : \forall\, i \in [1,n], (l_i^{\boldsymbol{s}} - l_i^{\boldsymbol{s}'} \geq \delta_{i^*}^l) \wedge (t_i^{\boldsymbol{s}} - t_i^{\boldsymbol{s}'} \geq \delta_{i^*}^t)\}.$$

Based on the probability densities of $\psi^{\mathsf{int}}(\boldsymbol{x}, \beta_l, \beta_t, d)$ and $\psi^{\mathsf{int}}(\tilde{\boldsymbol{x}}, \beta_l, \beta_t, d)$, for all $\boldsymbol{s} \in \dot{S}_1^{\mathsf{add}}$, we have:

$$\frac{\mathbb{P}_{\boldsymbol{z} \sim \psi^{\mathsf{int}}(\tilde{\boldsymbol{x}}, \beta_l, \beta_t, d)}(\boldsymbol{z} = \boldsymbol{s})}{\mathbb{P}_{\boldsymbol{z} \sim \psi^{\mathsf{int}}(\boldsymbol{x}, \beta_l, \beta_t, d)}(\boldsymbol{z} = \boldsymbol{s})} = 0.$$

For all $\boldsymbol{s} \in \dot{S}_2^{\mathsf{add}}$, we have:

$$\frac{\mathbb{P}_{\boldsymbol{z} \sim \psi^{\mathsf{int}}(\tilde{\boldsymbol{x}}, \beta_l, \beta_t, d)}(\boldsymbol{z} = \boldsymbol{s})}{\mathbb{P}_{\boldsymbol{z} \sim \psi^{\mathsf{int}}(\boldsymbol{x}, \beta_l, \beta_t, d)}(\boldsymbol{z} = \boldsymbol{s})} = \frac{C_n^d}{C_{n+n^{\mathsf{ins}}-n^{\mathsf{del}}}^d} \cdot e^{\sum_{i=1}^d \frac{\delta_{i^*}^l}{\beta_l} + \frac{\delta_{i^*}^t}{\beta_t}}.$$

Define:

$$\dot{p}_A^{\mathsf{sel}} = \mathbb{P}_{\boldsymbol{z} \sim \psi^{\mathsf{int}}(\boldsymbol{x}, \beta_l, \beta_t, d)}(f(\boldsymbol{z}) = y_A \wedge \boldsymbol{z} \in \dot{S}^{\mathsf{sel}}),$$
$$\dot{\tilde{p}}_A^{\mathsf{sel}} = \mathbb{P}_{\boldsymbol{z} \sim \psi^{\mathsf{int}}(\tilde{\boldsymbol{x}}, \beta_l, \beta_t, d)}(f(\boldsymbol{z}) = y_A \wedge \boldsymbol{z} \in \dot{S}^{\mathsf{sel}}).$$

Let $\dot{K} = C_n^d / C_{n+n^{\mathsf{ins}}-n^{\mathsf{del}}}^d \cdot e^{\sum_{i=1}^d \delta_{i^*}^l/\beta_l + \delta_{i^*}^t/\beta_t}$, the lower bound of $\dot{\tilde{p}}_A^{\mathsf{sel}}$ can be derived as follows:

$$\begin{aligned}
\dot{p}_A^{\mathsf{sel}} &= \mathbb{P}_{\boldsymbol{z} \sim \psi^{\mathsf{int}}(\boldsymbol{x}, \beta_l, \beta_t, d)}(f(\boldsymbol{z}) = y_A \wedge \boldsymbol{z} \in \dot{S}_1^{\mathsf{add}}) \\
&\quad + \mathbb{P}_{\boldsymbol{z} \sim \psi^{\mathsf{int}}(\boldsymbol{x}, \beta_l, \beta_t, d)}(f(\boldsymbol{z}) = y_A \wedge \boldsymbol{z} \in \dot{S}_2^{\mathsf{add}}), \\
\dot{\tilde{p}}_A^{\mathsf{sel}} &= \mathbb{P}_{\boldsymbol{z} \sim \psi^{\mathsf{int}}(\tilde{\boldsymbol{x}}, \beta_l, \beta_t, d)}(f(\boldsymbol{z}) = y_A \wedge \boldsymbol{z} \in \dot{S}_1^{\mathsf{add}}) \\
&\quad + \mathbb{P}_{\boldsymbol{z} \sim \psi^{\mathsf{int}}(\tilde{\boldsymbol{x}}, \beta_l, \beta_t, d)}(f(\boldsymbol{z}) = y_A \wedge \boldsymbol{z} \in \dot{S}_2^{\mathsf{add}}) \\
&= 0 + \dot{K} \cdot \mathbb{P}_{\boldsymbol{z} \sim \psi^{\mathsf{int}}(\boldsymbol{x}, \beta_l, \beta_t, d)}(f(\boldsymbol{z}) = y_A \wedge \boldsymbol{z} \in \dot{S}_2^{\mathsf{add}}) \\
&= \dot{K} \cdot [\dot{p}_A^{\mathsf{sel}} - \mathbb{P}_{\boldsymbol{z} \sim \psi^{\mathsf{int}}(\boldsymbol{x}, \beta_l, \beta_t, d)}(f(\boldsymbol{z}) = y_A \wedge \boldsymbol{z} \in \dot{S}_1^{\mathsf{add}})] \\
&\geq \dot{K} \cdot [\dot{p}_A^{\mathsf{sel}} - \mathbb{P}_{\boldsymbol{z} \sim \psi^{\mathsf{int}}(\boldsymbol{x}, \beta_l, \beta_t, d)}(\boldsymbol{z} \in \dot{S}_1^{\mathsf{add}})] \\
&= \dot{K} \cdot [\dot{p}_A^{\mathsf{sel}} - \mathbb{P}_{\boldsymbol{z} \sim \psi^{\mathsf{int}}(\boldsymbol{x}, \beta_l, \beta_t, d)}(\boldsymbol{z} \in \dot{S}^{\mathsf{sel}}) \\
&\quad + \mathbb{P}_{\boldsymbol{z} \sim \psi^{\mathsf{int}}(\boldsymbol{x}, \beta_l, \beta_t, d)}(\boldsymbol{z} \in \dot{S}_2^{\mathsf{add}})] \\
&= \dot{K} \cdot [\dot{p}_A^{\mathsf{sel}} - 1/C_n^d + 1/(\dot{K} \cdot C_{n+n^{\mathsf{ins}}-n^{\mathsf{del}}}^d)] \\
&= \dot{K} \cdot (\dot{p}_A^{\mathsf{sel}} - 1/C_n^d) + 1/C_{n+n^{\mathsf{ins}}-n^{\mathsf{del}}}^d.
\end{aligned}$$

Assume that $(\overline{\delta}_1^l, \overline{\delta}_2^l, \ldots, \overline{\delta}_n^l)$ and $(\overline{\delta}_1^t, \overline{\delta}_2^t, \ldots, \overline{\delta}_n^t)$ are obtained by sorting $\boldsymbol{\delta}^l$ and $\boldsymbol{\delta}^t$ in descending order of $\delta_i^l/\beta_l + \delta_i^t/\beta_t$, respectively. Let $K = C_n^d / C_{n+n^{\mathsf{ins}}-n^{\mathsf{del}}}^d \cdot e^{\sum_{i=1}^d \overline{\delta}_i^l/\beta_l + \overline{\delta}_i^t/\beta_t}$, we have $\dot{K} \leq K$. Since $\dot{p}_A^{\mathsf{sel}} - 1/C_n^d \leq 0$, we have:

$$\begin{aligned}
\dot{\tilde{p}}_A^{\mathsf{sel}} &\geq \dot{K} \cdot (\dot{p}_A^{\mathsf{sel}} - 1/C_n^d) + 1/C_{n+n^{\mathsf{ins}}-n^{\mathsf{del}}}^d \\
&\geq K \cdot (\dot{p}_A^{\mathsf{sel}} - 1/C_n^d) + 1/C_{n+n^{\mathsf{ins}}-n^{\mathsf{del}}}^d.
\end{aligned}$$

Finally, we have:

$$\sum_{\{x_1^{\boldsymbol{u}}, x_2^{\boldsymbol{u}}, \ldots, x_d^{\boldsymbol{u}}\} \subseteq V} \dot{\tilde{p}}_A^{\mathsf{sel}} \geq \sum_{\{x_1^{\boldsymbol{u}}, x_2^{\boldsymbol{u}}, \ldots, x_d^{\boldsymbol{u}}\} \subseteq V} K \cdot (\dot{p}_A^{\mathsf{sel}} - \frac{1}{C_n^d}) + \frac{1}{C_{n+n^{\mathsf{ins}}-n^{\mathsf{del}}}^d},$$
$$\tilde{p}_A^{\mathsf{sel}} \geq K \cdot (p_A^{\mathsf{sel}} - 1) + \frac{C_n^d}{C_{n+n^{\mathsf{ins}}-n^{\mathsf{del}}}^d}.$$

□

## B Experiment Setup

**Traffic Analysis Models.** We comprehensively investigate recent learning-based traffic analysis models and observe that the flow representations used by these models can be categorized into three types: flow statistics, raw flow sequences and raw bytes. We choose kFP [11] and Kitsune [27] as representative models that use flow statistics as input. We extract 175 statistical features from the lengths, timestamps and directions of packets to serve as the input of these two models. In terms of learning algorithms, kFP is based on traditional

machine learning algorithm Random Forest, while Kitsune uses an ensemble of neural networks called autoencoders. To adapt Kitune from the unsupervised anomaly detection task to our multi-class classification tasks, we feed the hidden vectors encoded by Kitsune to a subsequent Multi-Layer Perceptron to obtain the predicted class. Whisper [8] and DFNet [38] are selected as representative models that take raw flow sequences as input. Specifically, these two models perform inference based on the directional packet length sequence and the inter-arrival time sequence of a flow. Given the raw flow sequences, Whisper performs frequency domain analysis based on Discrete Fourier Transformation, while DFNet employs a Convolution Neural Network for classification. To adapt Whisper from the unsupervised anomaly detection task to our multi-class classification tasks, we feed its frequency domain features to a subsequent Randomized Forest classifier. For representatives of models that use raw packet bytes as input, we select the MAE-based YaTC [55] and the BERT-based TrafficFormer [56]. These two models extract raw bytes from the header and payload of the first 5 packets in a flow for analysis and carefully design pre-training and fine-tuning tasks for traffic data. To avoid over-fitting on strong identification information, we remove the Ethernet header, mask the IP addresses and ports to zeros or ones to represent the packet direction.

**Adversarial Attacks.** We thoroughly review recent adversarial attack methods in the field of traffic analysis. The underlying optimization algorithms of these methods can be summarized into three categories: Generative Adversarial Networks (GAN), Reinforcement Learning (RL) and explicit modeling. We select a representative attack method from each of these categories. For GAN-based methods, we choose Blanket [29], which trains separate noise generators for different types of perturbations and combines them to generate adversarial flows. For RL-based methods, we select Amoeba [26], which optimizes the perturbation policy using black-box prediction results and attack overheads as rewards. For explicit modeling based methods, we use Prism [22], which utilizes a Time-Stacked State Transition Model to capture the temporal patterns of each flow class and crafts adversarial flows based on these patterns. Given an input flow from the test dataset, we use Blanket, Amoeba and Prism to optimize the perturbation operations of packet insertion, length padding and timing delays. Subsequently, we generate an adversarial flow by applying the optimized perturbation operations to the original input flow.

**Certification Methods.** Existing certification methods focus on single-modal adversarial perturbations. Specifically, we compare CertTA with three baseline certification methods including VRS [5], BARS [39] and RS-Del [13].

- VRS treats network flow features as an $1 \times D$ vector and applies Gaussian noises with unified shape parameters of $(0, \sigma^2)$ to generate smoothing samples. For traffic analysis models using different flow representations, $\sigma$ measures the scale of the numerical noise added to a statistical flow feature, the length and timing of a packet, or a raw byte. Further, VRS provides an isotropic $\ell_2$-norm robustness radius against additive perturbations on network flow features.
- BARS improves upon VRS by taking into account the diverse scales of different flow features and providing anisotropic robustness radii for different dimensions of the feature vector. It introduces a distribution transformer to automatically optimize the shape of random noise added to each dimension of the feature vector. $\lambda$ is a regularization weight for training the distribution transformer, while $H_f$ represents the type of noise distribution.
- RS-Del views a network flow as a discrete sequence of packets and provides robustness guarantees against discrete perturbations like packet insertion. When generating a smoothing sample, each packet of the flow will be deleted by RS-Del with a probability of $p^{\mathsf{del}}$.

We use the open-source implementations of these certification approaches and tune the smoothing hyper-parameters to ensure that the certified models retain sufficient efficacy on the clean dataset. The parameter tuning methods follow the recommendation in the original papers of these approaches.

**Software and Hardware.** We implement CertTA with PyTorch under Python 3. The "pathos.multiprocessing" Python library is utilized to generate multiple smoothing samples in parallel for acceleration. Experiments are conducted on a Supermicro SYS-740GP-TNRT server with two Intel(R) Xeon(R) Gold 6348 CPUs ($2 \times 28$ cores), 512GB RAM, one NVIDIA A100 GPU and two NVIDIA GeForce RTX 4090 GPUs.

## C Certification against Packet Reordering

In addition to the current five types of traffic perturbations (*i.e.,* packet insertion, substitution, deletion, length padding and timing delays), the network traffic may include other types of perturbations, such as packet reordering caused by networking variations or adversarial attacks. In this section, we provide a certification method against packet reordering, which can be integrated into the certification framework of CertTA in future work.

**Lemma 3.** *Consider a pair of traffic flows $\boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathcal{X}$, where $\boldsymbol{x}$ contains $n$ packets $(x_1, x_2, \ldots, x_n)$ and $\boldsymbol{x}$ can be perturbed into $\tilde{\boldsymbol{x}}$ by reordering $(x_1, x_2, \ldots, x_n)$ to $(x_{i_1}, x_{i_2}, \ldots, x_{i_n})$. Define the reordering perturbation vector $\boldsymbol{\delta}^{reo} = (\delta_1^{reo}, \delta_2^{reo}, \ldots, \delta_n^{reo})$, where $\delta_j^{reo} = i_j - j$. Let $\psi^{reo}(\boldsymbol{x}, \lambda) : \mathcal{X} \times \mathbb{Z}^+ \to \mathcal{Z}$ be the smoothing function that (i) randomly selects a start offset from $[-\lambda + 2, 1]$; (ii) splits flow $\boldsymbol{x}$ into windows of $\lambda$ consecutive packets from the start offset and randomly shuffles the packets within each window ($\lambda \leq n$). Define the smoothed classifier $g^{reo}$ as in Equation (1). Suppose $y_A \in \mathcal{Y}$*

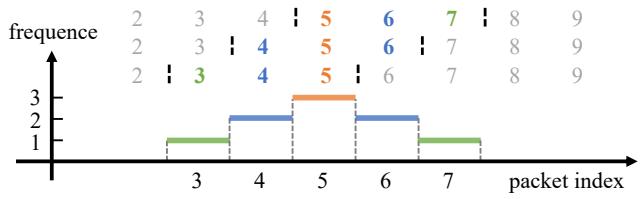


Figure 12: The possible new indices of packet $x_5$ in smoothing samples generated by the reordering smoothing mechanism.

*and* $\underline{p_A} \in [1/2,1]$ *satisfy* $g^{reo}(\boldsymbol{x}) = y_A$ *and* $p_A \geq \underline{p_A} \geq 1/2$, *then we have* $g^{reo}(\tilde{\boldsymbol{x}}) = g^{reo}(\boldsymbol{x}) = y_A$ *if:*

$$\sum_{j=1}^{n} |\delta_j^{reo}| < r^{reo}, \tag{9}$$

*where the robustness radius* $r^{reo} = \lambda(2\underline{p_A} - 1)$.

*Proof.* When generating smoothing samples by $\psi^{\mathsf{reo}}(\boldsymbol{x},\lambda)$, the $j$-th packet $x_j$ in flow $\boldsymbol{x}$ may fall into different windows based on the randomly selected start offset. Figure 12 gives an illustrative example where $j = 5$ and $\lambda = 3$. Since $x_5$ could fall into 3 windows, its new index in the smoothing samples is in the range of $[3,7]$. Similarly, denote the new index of $x_j$ in the smoothing samples as $j'$, $j'$ is in the range of $J = [j-(\lambda-1), j+\lambda-1]$. Based on the probability distribution of $\psi^{\mathsf{reo}}(\boldsymbol{x},\lambda)$, we have:

$$\mathbb{P}(j' = j+k) = \max(\lambda - |k|, 0)/\lambda^2. \tag{10}$$

According to the definition of $\boldsymbol{\delta}^{\mathsf{reo}}$, the new index of $x_j$ in flow $\tilde{\boldsymbol{x}}$ is $j + \delta_j^{\mathsf{reo}}$. Further, denote the new index of $x_j$ in the smoothing samples generated by $\psi^{\mathsf{reo}}(\tilde{\boldsymbol{x}},\lambda)$ as $\tilde{j}'$, $\tilde{j}'$ is in the range of $\tilde{J} = [j + \delta_j^{\mathsf{reo}} - (\lambda-1), j + \delta_j^{\mathsf{reo}} + \lambda - 1]$.

When $\delta_j^{\mathsf{reo}} \in [0, \lambda)$, according to Equation (10), we have:

$$\begin{aligned}
\mathbb{P}(j' \in J - \tilde{J}) =& \mathbb{P}(j' \in [j-(\lambda-1), j+\delta_j^{\mathsf{reo}} - \lambda)]) \\
=& \sum_{k=-(\lambda-1)}^{\delta_j^{\mathsf{reo}}-\lambda} \max(\lambda - |k|, 0)/\lambda^2 \\
=& \frac{\delta_j^{\mathsf{reo}}(\delta_j^{\mathsf{reo}}+1)}{2\lambda^2} \leq \frac{\delta_j^{\mathsf{reo}}}{2\lambda},
\end{aligned}$$

$$\begin{aligned}
\mathbb{P}(j' \notin J - \tilde{J}, \forall j \in [1,n]) =& 1 - \sum_{j=1}^{n} \mathbb{P}(j' \in J - \tilde{J}) \\
\geq& 1 - \frac{\sum_{j=1}^{n} \delta_j^{\mathsf{reo}}}{2\lambda}.
\end{aligned}$$

Let $S$ be the set of all possible smoothing samples generated by $\psi^{\mathsf{reo}}(\boldsymbol{x},\lambda)$. We partition $S$ into two sets of flows $S_1, S_2$:

$$\begin{aligned}
S_1 &= \{\boldsymbol{s} \in S : j' \notin J - \tilde{J}, \forall j \in [1,n]\}, \\
S_2 &= \{\boldsymbol{s} \in S : j' \in J - \tilde{J}, \exists j \in [1,n]\}.
\end{aligned}$$

Based on the probability distribution of $\psi^{\mathsf{reo}}(\boldsymbol{x},\lambda)$ and $\psi^{\mathsf{reo}}(\tilde{\boldsymbol{x}},\lambda)$, we have:

$$\frac{\mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{reo}}(\tilde{\boldsymbol{x}},\lambda)}(\boldsymbol{z}=\boldsymbol{s})}{\mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{reo}}(\boldsymbol{x},\lambda)}(\boldsymbol{z}=\boldsymbol{s})} = \begin{cases} 1, & \forall \boldsymbol{s} \in S_1, \\ 0, & \forall \boldsymbol{s} \in S_2. \end{cases}$$

The lower bound of $\tilde{p}_A$ can be derived as follows:

$$\begin{aligned}
p_A =& \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{reo}}(\boldsymbol{x},\lambda)}(f(\boldsymbol{z}) = y_A \ \wedge\ \boldsymbol{z} \in S_1) \\
&+ \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{reo}}(\boldsymbol{x},\lambda)}(f(\boldsymbol{z}) = y_A \ \wedge\ \boldsymbol{z} \in S_2), \\
\tilde{p}_A =& \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{reo}}(\tilde{\boldsymbol{x}},\lambda)}(f(\boldsymbol{z}) = y_A) \\
\geq& \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{reo}}(\tilde{\boldsymbol{x}},\lambda)}(f(\boldsymbol{z}) = y_A \ \wedge\ \boldsymbol{z} \in S_1) \\
=& \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{reo}}(\boldsymbol{x},\lambda)}(f(\boldsymbol{z}) = y_A \ \wedge\ \boldsymbol{z} \in S_1) \\
=& p_A - \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{reo}}(\boldsymbol{x},\lambda)}(f(\boldsymbol{z}) = y_A \ \wedge\ \boldsymbol{z} \in S_2) \\
\geq& p_A - \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{reo}}(\boldsymbol{x},\lambda)}(\boldsymbol{z} \in S_2) \\
=& p_A - 1 + \mathbb{P}_{\boldsymbol{z}\sim\psi^{\mathsf{reo}}(\boldsymbol{x},\lambda)}(\boldsymbol{z} \in S_1) \\
=& p_A - 1 + \mathbb{P}(j' \notin J - \tilde{J}, \forall j \in [1,n]) \\
\geq& \underline{p_A} - \frac{\sum_{j=1}^{n} \delta_j^{\mathsf{reo}}}{2\lambda}.
\end{aligned}$$

Finally, we can get Equation (9) by solving the inequality that the lower bound of $\tilde{p}_A$ is not less than $1/2$. When $\delta_j^{\mathsf{reo}} \in (-\lambda, 0]$, Equation (9) can be derived in a similar way. □