Dimensional Robustness Certification for Deep Neural Networks in Network Intrusion Detection Systems

MENGDIE HUANG, Xidian University, Purdue University, China, United States
YINGJUN LIN, Purdue University, United States
XIAOFENG CHEN, Xidian University, China
ELISA BERTINO, Purdue University, United States

Network intrusion detection systems based on deep learning are gaining significant traction in network security, valued for their high detection accuracy and strong adaptability to evolving cyber threats. However, a serious drawback is their vulnerability to evasion attacks that rely on adversarial examples and natural corruptions caused by random noise in network environments. To provide robustness guarantees for deep neural networks against any possible perturbations, certified defenses against perturbations within a l_p-bounded region around the input are gaining attention. However, unlike existing image domain approaches that concentrate on homogeneous input feature spaces, the progress on certified defense for the network traffic domain, which is characterized by heterogeneous features, has been very limited. To address such a gap, we propose a novel framework, Multi-order Adaptive Randomized Smoothing (MARS), for certifying the robustness of network intrusion detectors based on deep neural networks. Experiments on various network intrusion detection systems show that MARS significantly improves the tightness of robustness certification (12.23% increase in l_2 certified radius), detection accuracy on evasion attack (7.17% improvement on l_∞-PGD, 10.11% improvement on l_1-EAD), and prediction accuracy on natural corruption (16.65% enhancement on latency, 18.23% enhancement on packet loss) compared to the SOTA method. We have also conducted an extensive analysis of the dimension-wise certified robustness of the network intrusion detector. The results show that dimensional certified radii obtained by MARS reveal the robustness differences between feature dimensions, which is consistent with the empirical evaluation findings.

CCS Concepts: • **Computing methodologies** → **Machine learning**; • **Networks** → **Network security**; • **Security and privacy** → **Intrusion detection systems**.

Additional Key Words and Phrases: Deep neural network, certified robustness, randomized smoothing, evasion attack, natural corruption, network intrusion detection

ACM Reference Format:
Mengdie Huang, Yingjun Lin, Xiaofeng Chen, and Elisa Bertino. 2024. Dimensional Robustness Certification for Deep Neural Networks in Network Intrusion Detection Systems. *ACM Trans. Priv. Sec.* 1, 1, Article 1 (December 2024), 33 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Deep learning (DL)-based network intrusion detection systems (NIDS) excel in detecting complex and evolving cyber threats by leveraging the capability of deep neural networks (DNNs) to analyze

Authors' Contact Information: Mengdie Huang, huan1932@purdue.edu, Xidian University, Purdue University, Xi'an, West Lafayette, Shaanxi, Indiana, China, United States; Yingjun Lin, lin1368@purdue.edu, Purdue University, West Lafayette, Indiana, United States; Xiaofeng Chen, xfchen@xidian.edu.cn, Xidian University, Xi'an, Shaanxi, China; Elisa Bertino, bertino@purdue.edu, Purdue University, West Lafayette, Indiana, United States.

ACM 2471-2574/2024/12-ART1
https://doi.org/10.1145/nnnnnnn.nnnnnnn

large-scale and diverse network traffic data [6, 52]. However, previous work has shown that DNN-based network traffic classifiers [13, 46] are as vulnerable to evasion attacks using adversarial examples as text [21], image [16, 53], speech [37], and video classifiers [2, 23]. An attacker can transform an otherwise correctly classified clean input into an adversarial example by adding subtle perturbations [3, 28], causing the victim classifier to misclassify these inputs. Adversarial malicious traffic usually mimics normal traffic patterns with constrained modifications, thereby evading detection while maintaining features closely resembling those of clean malicious traffic [31].

Empirical defense methods designed to enhance the robustness of DNNs, such as adversarial training [10, 28], feature denoising [24, 49], and model ensembling [26, 32], have been fully verified in the DL field, and their applicability has also been explored for network intrusion detection [54]. However, a common problem in various domains is that the robustness achieved by these empirical defenses driven by heuristic strategies is likely to be bypassed by new evasion attack methods [5, 45]. Such a shortcoming allows attackers to evade the "supposedly robust" model through adaptive attacks [40], leading to an endless arms race of evasion attacks and defenses. Moreover, in security-sensitive applications, such as autonomous driving, healthcare, and network intrusion detection, it is difficult to establish a high level of trust in model outputs when relying on empirical defenses.

Recognizing these shortcomings, research on the robustness of DNNs has gradually shifted to certified defense [22]. Such a defense aims to calculate a certified robust radius (a.k.a. *certified radius*) for each input, to indicate that the model's predictions remain consistent for any variant of the current input within the region bounded by this certified radius. The certified radius is provided as a *robustness guarantee* along with the prediction of the model on the input. For the same model and input sample, a larger certified radius obtained indicates a tighter robustness guarantee provided by the certified defense method. Incomplete certification, which aims to compute the lower bound of the exact robust radius of the model as the certified radius, avoids the NP-complete challenge of computing the exact robust radius of a DNN in complete certification [19, 22]. However, a notable drawback of incomplete certification is that the robustness guarantee it provides is often loose; that is, the calculated lower bound is significantly smaller than the exact robust radius.

To compute the non-trivial certified radius for DNNs, many approaches have been proposed to upgrade incomplete certification algorithms for image classifiers. This includes (i) deterministic robustness certification, such as activation polytope [9, 17, 25, 43, 48], interval bound propagation [1, 11, 38], relaxation [34–36, 51], neuron branching and bounding [27, 47, 50, 57], and (ii) probabilistic robustness certification, such as differential privacy [18, 33] and randomized smoothing [5, 8, 14, 19, 29]. Given that an ideal certified defense approach against evasion attacks should be model agnostic, that is, it should be applicable to various types of DNNs without modifying or being limited by the specific internal structures of these models, randomized smoothing-based methods have been proven to be the most competitive in terms of tighter and architecturally scalable certification in the field of image [22], text [30], and graph [44] classification.

However, certified defense efforts for NIDS have been minimal. The main challenges arise from the heterogeneity of network traffic features, where different dimensions carry varying semantics and characteristics. In contrast to image features representing pixel values, network traffic feature dimensions involve protocol types, destination network services, timestamps, data packet counts, flow-byte rates, and more. Furthermore, the diversity of NIDS architectures also introduces difficulties to certified defenses. Various detection tasks, such as binary or multi-class classification and known or unknown anomaly detection, often involve the use of different DNN models, including CADE (Contrastive Autoencoder for Drifting detection and Explanation) [52], ACID (Adaptive Clustering-based Intrusion Detection) [6], and others. This imposes strict demands on the scalability of certification methods across diverse model architectures. Until now, only one approach, BARS (Boundary-Adaptive Randomized Smoothing) [45], has been proposed to certify

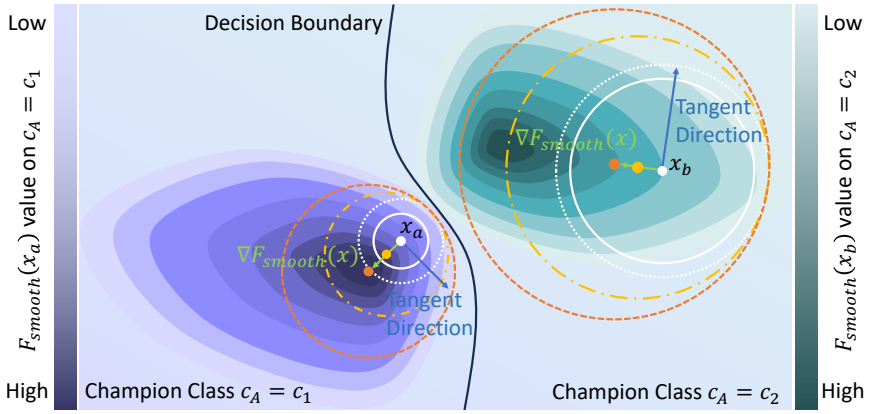


Fig. 1. Certified radius obtained through MARS. c_A denotes the class with the highest confidence returned by the smoothed classifier F_{smooth} on the input. x_a and x_b are two inputs predicted as different champion classes, c_1 and c_2. The solid white line outlines the certified region relying solely on zero-order output $F^{c_A}_{smooth}(x)$. The blue arrow indicates the tangent direction of $F^{c_A}_{smooth}(x)$ at x. The green arrow represents the gradient vector $\nabla F^{c_A}_{smooth}(x)$. The dotted white line outlines the certified region utilizing zero-order and a gradient with zero magnitude. The yellow and orange dotted lines outline the certified regions employing zero-order and a gradient with non-zero magnitude, where orange corresponds to a larger gradient magnitude.

the robustness of network traffic classifiers. Using randomized smoothing based on zero-order information, it obtained larger l_2 certified radii than leading neuron branching and bounding methods. Unfortunately, its l_2 robustness guarantee has been proven relatively loose, and it lacks certification for other l_p norm-bounded robustness guarantees. Providing multiple l_p-bounded certified radii can help in deeply analyzing model vulnerabilities and boosting general robustness against ever-changing evasion attacks using diverse norms like l_1 or l_∞ in different contexts. Moreover, BARS only handles evasion attacks, neglecting some natural corruptions, such as latency and packet loss, that may be induced by random noise in the network environment.

1.1 Overview of Our Method

To address the above shortcomings, we propose MARS—a novel framework that certifies the robustness of DNNs in NIDS using **M**ulti-order **A**daptive **R**andomized **S**moothing. The characteristics of MARS are mainly demonstrated in the following three aspects: (i) dynamically expands the certified robust region along the high confidence direction; (ii) adaptively samples random noise near the decision boundary; (iii) and supports the flexible selection of the smoothing distribution type.

First, to adapt to the heterogeneity of d-dimensional network traffic features, we design a dimensional-wise certified radius calculation method by expanding the real value of the certified radius into a certified radius vector for the network traffic domain, and quantifying the radius contribution of each dimension based on the dimensional feature sensitivity analysis.

Then, to tighten the robustness guarantee, we adopt a two-step strategy: (a) We optimize the dimensional parameters of the multivariate smoothing distribution so that the certification algorithm can adaptively sample dense noised samples near the boundary for probability statistics. (b) We iteratively shift the symmetry center of the l_p certified robust area along the gradient direction of the smoothed classifier; that is, the direction in which the confidence score of the output class increases, as illustrated in Figure 1. We then use binary search to estimate the upper and lower bounds of the interval of the certified radius so that the certified radius can be further improved.

Finally, to provide diverse l_p certificates, we construct a set of smoothing distribution candidates consisting of Gaussian, Laplacian, and Uniform distributions and implement specific parameter optimization and first-order gradient estimation for each distribution.

1.2 Contributions

We evaluate the performance of MARS on two advanced DL-based NIDS architectures, CADE [52] and ACID [6], with four datasets created from CSE-CIC-IDS-2018 [41], and compare MARS with the state-of-the-art (SOTA) network traffic-specific certification BARS [45], image-specific certification Vanilla Randomized Smoothing (VRS) [5], and First Order-based Randomized Smoothing (FRS) [29] in terms of overall and dimensional certified radius, certified accuracy, robust accuracy on evasion attacks and natural corruptions, clean accuracy on clean samples, and certification time cost.

In summary, we make the following contributions:

- We propose and develop a robustness certification framework, MARS, to certify the robust radius of DNNs in NIDS without requiring any modification to the model structure. It achieves a tighter l_2 robustness guarantee (12.23% average increase compared to BARS) and extends certification from l_2 to l_1 and l_∞ guarantees compared to other advanced methods.
- We design and implement a dimensional-wise robustness certification approach for input data with heterogeneous features. The obtained dimensional certified radius reflects the sensitivity of each feature dimension of network traffic in a fine-grained manner.
- We are the first to utilize the high-order information of the smoothed classifier to guide the expansion of the certified region obtained based on the zero-order output information in network traffic classification. Our approach demonstrates improved tightness in various l_p robustness guarantees.
- We are the first to introduce a threat model of random noise-based natural corruption in addition to the threat of evasion attacks in robustness certification for NIDS. Our experimental results confirm that MARS significantly enhances robustness against evasion attacks (33.93% higher on l_∞-PGD, 13.79% higher on l_2-PGD, 10.01% higher on l_1-EAD) and natural corruptions (16.87% higher on Latency, 19.85% higher on Packet Loss) compared to the base model.

An extended abstract of this paper has been published in TrustCom [15]. The main differences between this paper and the conference version are as follows: Firstly, we added four brand new sections, 2, 3, 4, and 8, focusing on introducing related work, preliminaries, and problem statement. The application scenarios of the proposed approach and the importance of diversity in the types of norms used to constrain the certified robust regions have also been discussed. Secondly, we added five whole new subsections, 6.5, 7.5, 7.6, 7.7, and 7.8, Subsection 6.5 aim to outline methods for constructing threat models, compare certified and empirical robustness of network traffic classifiers against SOTA methods for intrusion detection on a new dataset, evaluate the dimensional certified radius of the MARS-defended classifier, examine the impact of different smoothing distributions on robustness certification for various l_p norms, and compare certification time overhead across defense methods. using average time per class as a metric. Additionally, we optimized the entire content, including emphasizing overview of the proposed method and contributions related to certified robustness in Section 1, constructing Algorithm 1 and 2 as well as introducing three new subsections (5.3.1, 5.3.2, and 5.3.3) to explain the strategy for aligning the smoothing distribution sampling region with the l_p certified region in Section 5, introducing a new Section 6 to detail the experimental setup, redrawing all bar figures in our experimental evaluation to add texture distinction to the legend, refining the experimental analysis in Section 7 to provide a more comprehensive analysis and adding a summary of the core idea of the proposed method in Section 9. In total, nine additional figures and five tables are provided, as well as fourteen references.

2 Related Work

Robustness certification methods are categorized into complete and incomplete certification. Complete certification yields a certified radius equal to the exact robust radius, while incomplete certification provides a lower bound. They can also be classified as deterministic or probabilistic. Deterministic certification guarantees to produce not certified when the input is non-robust, while probabilistic methods do so with a certain probability. Most complete certification approaches are deterministic, whereas incomplete certification includes both types [22].

2.1 Complete Certification

Complete certification ensures strong robustness guarantees by confirming no adversarial examples exist within a certified radius around any input point. Early work, such as Cheng et al. [4], formulated certification as a mixed integer linear programming problem to encode ReLU operations and models, later extended to convolutional networks by Tjeng et al. [39]. However, these methods are limited to medium-sized models and lack scalability for DNNs. To address scalability, Zhang et al. [57] proposed CROWN, a boundary propagation framework calculating upper and lower bounds layer by layer. Simplified techniques like interval bound propagation (IBP) [12] improved speed but sacrificed tightness. Extensions such as CROWN-IBP [55] and β-CROWN [47] improved accuracy and computational efficiency. Recent advancements like GCP-CROWN [56] enhanced scalability with cutting-plane methods. Despite progress, complete certification remains computationally demanding, limiting its application to high-dimensional inputs and large models.

2.2 Incomplete Certification.

Incomplete certification approximates the exact robust radius as the certified radius [19, 22]. Various methods have been proposed to enhance incomplete certification for image classifiers, enabling non-trivial certified radius computation for DNNs.

2.2.1 Deterministic incomplete certification. Deterministic incomplete certification includes linear relaxation-based and Lipschitz constant-based approaches. Linear relaxation methods rely on activation polytope [9, 17, 25, 43, 48], which use hidden layer activation values to form polytope vertices, evaluating robustness through volume and surface area calculations. Lipschitz constant-based methods compute tight bounds to quantify output changes under input perturbations. To address the looseness of global Lipschitz constants, Lee et al. [20] refined them with convolutional layer analysis, while Fazlyab et al. [7] enhanced local Lipschitz estimation via semidefinite programming.

2.2.2 Probabilistic incomplete certification. Probabilistic incomplete certification mainly includes differential privacy (DP)-based and randomized smoothing (RS)-based approaches.

DP-based methods often leverage the first-order derivative of the output function. Lecuyer et al. [18] introduced a DP mechanism by adding noise to hidden layers during training, providing l_2 robustness guarantees. Phan et al. [33] combined DP with adversarial training, applying noise to gradient updates to limit model sensitivity.

An ideal certified defense should be model agnostic and applicable to various DL models without relying on their specific structures. RS-based approaches are the most competitive for tight and architecturally scalable certification and can be categorized into zero-order or first-order information-based techniques. Cohen et al. [5] proposed vanilla randomized smoothing (VRS), which uses Gaussian noise to transform classifiers into smoothed versions. Lee et al. [19] extended RS to broader distributions, guaranteeing l_0 robustness. Hao et al. [14] introduced GSmooth, a framework certifying robustness against semantic transformations. Mohapatra et al. [29] advanced first-order RS (FRS), combining gradients with zero-order outputs to calculate tighter l_2 bounds.

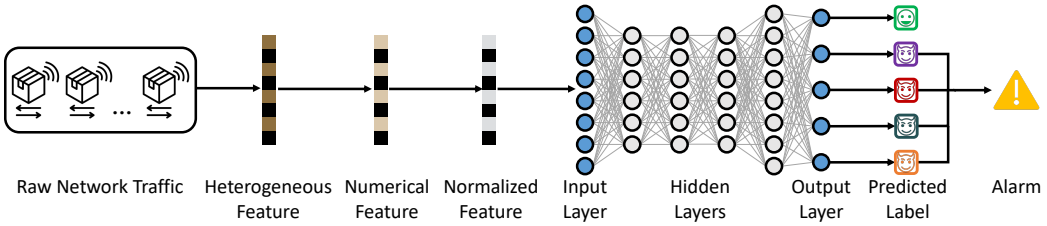


Fig. 2. Workflow of the deep learning-based network intrusion detector.

However, existing methods designed for DNNs with homogeneous input features, like image classifiers, cannot be applied to DL models with heterogeneous input features, such as NIDS. In 2023, Wang et al. [45] introduced Boundary-Adaptive Randomized Smoothing (BARS), the first robustness certification method aimed to provide l_2 robustness guarantees for network traffic classifiers. However, since it relies only on zero-order information, its l_2 norm-bounded certified radius may not be tight enough and lacks support for other norm-bounded certificates.

2.3 The Difference with Our Work

Our work focuses on probabilistic incomplete certification for network traffic data and differs from previous sole network traffic-domain certification defense work BARS in three aspects:

- Goal: They only compute the l_2-bounded certified radius of the NIDS, and we extend it to l_1 and l_∞ certification in addition to l_2 certification and achieve a larger certified radius.
- Threat model: They only focus on the robustness of NIDS against evasion attacks, while we define and evaluate the robustness against natural corruptions in addition to evasion attacks.
- Technique: They use the zero-order output information of the smoothed classifier, and we integrate zero-order information with first-order gradient information to enhance certification.

3 Preliminaries

This section outlines key definitions for robustness certification via randomized smoothing.

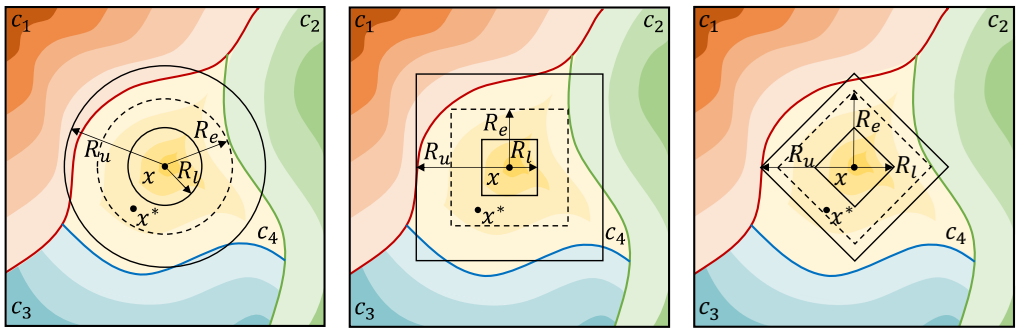
3.1 Base Classifier

As traffic data includes both numeric and non-numeric values, a typical pre-processing step is the transformation of non-numeric values into numeric values. Specifically, for a given d-dimensional raw network traffic feature vector x_{raw} containing some non-numerical feature dimensions (such as protocol, network service, timestamp, etc.), the vector is first transformed into a numerical feature vector x_{nmr} and then normalized into a feature vector $x \in \mathcal{X} \equiv \mathbb{R}^d$ that belongs to a continuous real number range, as shown in Figure 2.

The DNN model is formulated as a classification function f_θ: $\mathcal{X} \to \mathcal{Y} \equiv \mathbb{R}^C$, where the C-dimensional probability vector returned by the model reflects the confidence scores of x for all C classes. In the training stage, f_θ is supervised to map each sample x in the training set D_{train} to the C-dimensional one-hot vector of its ground-truth label $y_{true} \in [C] \equiv \{1, ..., C\}$ with the minimum loss, as shown in Equation (1).

$$\min_\theta \mathbb{E}_{(x, y_{true}) \sim D_{train}} \mathcal{L}(f_\theta(x), y_{true}), \tag{1}$$

where θ denotes the trainable model parameters, $\mathbb{E}$ is the expectation function, and $\mathcal{L}$ denotes the loss function. The base classifier F without certified defense is defined in Equation (2).



(a) l_2 radius: $||\delta||_2 < R$ (b) l_∞ radius: $||\delta||_\infty < R$ (c) l_1 radius: $||\delta||_1 < R$

Fig. 3. l_p-bounded certified radius of the multi-class classifier on the input x. Darker colors indicate higher confidence in the predicted class. Suppose x is predicted as c_4. The l_p-measured distance between the perturbed sample x^* and the clean sample x is $||\delta||_p$. R_u and R_l are the upper and lower bounds of the exact robust radius R_e on x.

Definition 1 (Base Classifier). *Given a DNN model f_θ that maps input samples x to a C-dimensional confidence score vector, a base classifier F is defined as:*

$$F(x) = \arg\max_{i \in [C]} f_\theta(x)_i, \tag{2}$$

where $f_\theta(x)_i$ represents the confidence score of the model f_θ on the i-th category for a given input x.

3.2 Robustness Guarantee

The robustness guarantee provided by a classifier for an input x against l_p-bounded perturbations is formalized as a l_p-bounded certified radius R of the robust region containing x, typically centered at x, as defined in Equation (3). Common ones include l_2 radius and l_∞ radius, as shown in Figure 3. This work aims to calculate the lower bound of the exact robust radius R_e as tight as possible, so the certified radius R provided in the following refers to the lower bound R_l.

Definition 2 (l_p Robustness Guarantee). *Given a classifier F, an input x, a perturbation δ, a distance measure l_p, robustness guarantee of F on x against l_p-bounded δ is defined as:*

$$\text{For all } \delta \text{ such that } ||\delta||_p < R,\ F(x+\delta) = F(x), \tag{3}$$

where R is the certified radius.

3.3 Smoothed Classifier

The smoothed classifier, defined in Equation (4), is transformed from the base classifier F through randomized smoothing. When queried at x, smoothed classifier F_{smooth} returns the class most likely predicted by F when x is perturbed by a large amount of noise from a smoothing distribution.

Definition 3 (Smoothed Classifier). *Given a base classifier F, an input x, a smoothing distribution $\mathcal{D}$ with Probability Density Function (PDF) φ, smoothed classifier F_{smooth} is defined as:*

$$F_{smooth}(x) = \arg\max_{c \in [C]} \mathbb{P}_{\eta \sim \mathcal{D}}(F(x+\eta) = c) = \arg\max_{c \in [C]} \int_{\eta \sim \mathcal{D}} \mathbb{I}[F(x+\eta) = c]\varphi(\eta)d\eta, \tag{4}$$

where $\mathbb{P}$ is the probability function, $\mathbb{I}$ is the indicator function that returns 1 when the condition is true, and $\varphi(\eta)$ denotes the probability density of the sampled noise data η.

Since the integral in the above equation cannot be solved exactly, the Monte-Carlo estimation defined in Equation (5) is commonly used to approximate the exact solution.

$$F_{smooth}(x) = \arg\max_{c\in[C]} \frac{1}{n}\sum_{k=1}^{n}\mathbb{I}[F(x+\eta_k)=c], \tag{5}$$

where n is the number of noise data η sampled from the smoothing distribution $\mathcal{D}$ and η_k denotes the k-th noise sample. Thus, the performance of the smoothed classifier F_{smooth} is determined by the base classifier F, the number of sampled noise n, and the smoothing distribution $\mathcal{D}$.

3.4 Multi-Order Information

3.4.1 Certification with Zero-order Information. Most randomized smoothing-based certification approaches only utilize the zero-order information of the smoothed classifier F_{smooth}, as shown in Equation (6), to calculate the certified radius R.

Definition 4 (Zero-order Information). *Given a smoothed classifier F_{smooth}, an input x, the zero-order information refers to the basic characteristics of F_{smooth}, such as the statistical probability P_c that F_{smooth} predicts x as each class c in $[C]$, defined as:*

$$For\ all\ c \in [C],\ P_c = \mathbb{P}(F_{smooth}(x)=c) = \mathbb{P}_{\eta\sim\mathcal{D}}(F(x+\eta)=c). \tag{6}$$

3.4.2 Certification with First-order Information. There are also a few approaches that attempt to introduce the high-order information of the smoothed classifier F_{smooth}, as shown in Equation (7), to compute a tighter robustness guarantee.

Definition 5 (First-order Information). *Given a smoothed classifier F_{smooth}, an input sample x, first-order information involves the derivative or gradient of F_{smooth}, defined as:*

$$\begin{aligned} For\ all\ c \in [C],\ &||\nabla F^c_{smooth}(x)||_p = ||\nabla \mathbb{P}(F_{smooth}(x)=c)||_p \\ &= ||\nabla \mathbb{P}_{\eta\sim\mathcal{D}}(F(x+\eta)=c)||_p = ||\frac{\partial(\mathbb{P}_{\eta\sim\mathcal{D}}(F(x+\eta)=c))}{\partial(x)}||_p, \end{aligned} \tag{7}$$

where $F^c_{smooth}(x) = \mathbb{P}(F_{smooth}(x)=c)$ is the statistical probability that F_{smooth} predicts x as c, and $||\nabla F^c_{smooth}(x)||_p$ is the l_p-measured magnitude of the gradient of F^c_{smooth} at x.

3.5 Evasion Attack

Evasion attacks aim to bypass detection models by manipulating input data. Unlike conventional evasions against NIDS involving straightforward manipulations like increasing packet numbers, interval, and payload length, attacks on network traffic classifier f_θ we focus on here often use sophisticated techniques to exploit vulnerabilities in deep architectures. Specifically, attackers generate adversarial examples $x^* = x + \delta$ by modifying the clean input x with a slight perturbation δ, as shown in Equation (8).

Definition 6 (l_p evasion Attack). *Given a target label y_{target} desired by the attacker, l_p evasion attack aims to find a perturbation δ that minimizes the loss $\mathcal{L}$ between $f_\theta(x+\delta)$ and y_{target} and satisfy the l_p-bounded budget ϵ, defined as:*

$$\min_{x^*} \mathcal{L}(f_\theta(x^*), y_{target}) = \min_{||\delta||_p<\epsilon} \mathcal{L}(f_\theta(x+\delta), y_{target}). \tag{8}$$

4 Problem Statement

This section describes the threat model, research problems, approach directions, and key challenges.

4.1 Threat Model

We consider two robustness threats faced by DNNs: evasion attacks—deliberately launched by attackers using adversarial examples, and natural corruptions—unintentionally caused distribution shift by random noise in the network environment.

4.1.1 Evasion Attacks. We first focus on white-box evasion attacks. The adversary creates strong l_p-bounded adversarial examples based on complete knowledge of the victim network traffic classifier f_θ. By assuming that the attacker possesses full knowledge of the model, we simulate the most powerful threat in the adversarial scenario where the adversary has maximum visibility into the model internals. This enables the evaluation of model robustness against sophisticated attacks leveraging its inner workings while also revealing potential vulnerabilities of the target model.

Evasion attacks can target clean samples belonging to any category. In NIDS, however, especially in multi-classification scenarios, the more realistic situation is that the evasion goal only includes causing the originally malicious traffic to be classified as benign, but does not include causing originally benign traffic to be classified as malicious or causing malicious traffic to be classified as another attack type. Thus, we assume that *the adversary will launch evasion attacks only on originally malicious traffic*, by setting the target label y_{target} in Equation (8) to benign.

4.1.2 Natural Corruptions. We also consider the robustness of the classifier to distribution shifts arising from natural variations in datasets. Natural corruptions result from uncontrollable environmental factors, such as lighting changes in images or recording device alterations in speech. As the first work to consider natural corruption in robustness certification in the traffic domain, we focus on the distribution shifts caused by random noise added to time-related and quantity-related traffic features. By assuming noise background in temporal and spatial characteristics, we aim to mimic a scenario where natural corruptions like *latency* and *packet loss* arise from network congestion or electromagnetic interference. Unlike evasion attacks, these corruptions are typically unintentional, thus *both clean benign and malicious traffic can be corrupted.*

4.2 Certified Defense Goal

Our design goal is to provide the traffic classifier prediction with a robustness guarantee that reflects the tight lower bound of the robustness of the model on any unknown perturbations.

4.2.1 Problems. We need to address the following three problems.

Problem 1. Formally define a certified radius as the l_p robustness guarantee that can constrain heterogeneous network traffic features. Homogeneous image feature vectors typically share semantics and value ranges across dimensions, resulting in a real-value certified radius R, constraining all dimensions concurrently. Conversely, heterogeneous traffic feature vectors exhibit varied semantics, value ranges, and significance in traffic analysis and noise tolerance. Thus, a traffic-specific certified radius form must be designed to reflect robust regions within heterogeneous feature dimensions.

Problem 2. Tighten the l_2 certified radius to provide a stricter l_2 robustness guarantee than the only existing certification approach for NIDS. A larger certified radius signifies higher prediction credibility of the model, essential in applications demanding high-confidence results. For NIDS, detection combined with a certified radius can minimize false positives and false negatives. For example, if it is required that only predictions with certified radii higher than the robust radius threshold on the predicted malicious class trigger anomaly warnings, alert fatigue can be alleviated.

Problem 3. The l_1 and l_∞ robustness guarantees of the model on a given input must also be provided. While l_2 attacks are common, adversaries may employ other norms (e.g., l_1, l_∞) for escape purposes.

It is vital to have a comprehensive approach for calculating diverse l_p measured certified radii to assess the overall robustness. Especially when confronting unknown types of evasion attacks, comparing various l_p certified radii is particularly valuable to thoroughly analyze the model's weaknesses and enhance l_p robustness.

4.2.2 Approach Directions and Challenges. To address these problems, the considered approach directions and main challenges are as follows.

Approach Direction to Problem 1. The solution we consider is to extend the real-value certified radius R to a vector $(R_1, ..., R_d) \in \mathbb{R}^d$, where R_i denotes the dimension-wise robustness guarantee for the i-th feature x_i of the input x. Yet, computing the certified radius vector R poses a challenge.

Challenge 1. Efficiently calculate the dimension-wise certified radius R_i for each dimension of the input $x = (x_1, ..., x_d) \in \mathbb{R}^d$. An easy way to calculate R_i is employing randomized smoothing separately for each dimension by adding noise to feature x_i while keeping other dimensions constant. However, since we assume that network traffic feature dimensions are not independent, this approach has two issues: ① The certified radius vector composed of the independently calculated dimension-wise certified radius represents the upper bound of the exact certified radius vector rather than the lower bound, because it does not account for the correlation between different feature dimensions of network traffic, such as x_i and x_j. ② Performing randomized smoothing independently for each feature dimension increases the time cost significantly by a factor of d, making it impractical for network traffic classification tasks with real-time requirements.

Approach Direction to Problem 2. To tighten the l_2 robustness guarantee, the solution we envision is to improve the randomized smoothing-based certification by introducing high-order information about the smoothed classifier. Yet, this solution requires addressing the following challenge.

Challenge 2. Characterize the correlation between the first-order information of a smoothed classifier and the certified radius and derive a tighter robustness guarantee by exploiting it. The first-order gradient information of the classification function on the input reveals how subtle changes in the input affect predictions. This valuable insight could determine robust regions in smoothed classifiers. However, this area remains unexplored in NIDS and is in the early stage of image classification. The challenge is the lack of analysis connecting this information to the certified radius, impeding its use as a supplementary condition for certified radius calculations.

Approach Direction to Problem 3. The solution we envision is to select distinct smoothing distributions for various l_p guarantees to align the sampling area of the noise samples used for smoothing with the l_p-bounded surroundings of the input x. Nevertheless, the challenge is as follows.

Challenge 3. For various l_p perturbations, choose suitable smoothing distribution types and parameter settings to obtain non-trivial tight robustness lower bounds. While Gaussian distributions have proven effective for l_2 perturbations when combined with zero-order information, they might not always be optimal for l_1 and l_∞ attacks, especially when first-order information is also used. Hence, it is essential to integrate prior knowledge to generate candidates for appropriate smoothing distributions and to experimentally ascertain whether the zero-order or first-order information under these distributions is appropriate for various l_p robustness certifications.

5 Design of MARS

This section introduces the design of the proposed certified defense method, Multi-order Adaptive Randomized Smoothing (MARS), to provide non-trivial tight l_p-bounded robustness guarantees. The framework includes three modules: Dimensional Radius Weight Calculation, Multi-Order Robustness Certification, and Smoothing Distribution Alignment.

Algorithm 1 Dimensional Radius Weight Calculation

Input: d-dimensional training samples $(x, y) \in D_{train}$, base C-class classifier F with the classification function f_θ, smoothed classifier F_{smooth}.
Output: d-dim weight vector w_c for each class $c \in \{1, .., C\}$

for $j = 1$ to $len(D_{train})$ **do**
  load one training example x_j, y_j;
  for $c = 1$ to C **do**
    $n_c \leftarrow 0$
    class c-specific dataset $D_{train,c} \leftarrow \emptyset$
    if $y_j = c$ **then**
      add (x_j, y_j) to $D_{train,c}$
      $n_c \leftarrow n_c + 1$
      sensitivity score $s_j = (s_j{}^1, s_j{}^2, ..., s_j{}^d) \leftarrow \frac{d(f_\theta^c(x_j))}{d(x_j)}$
    end if
  end for
end for
radius weight vector set $D_{weight} \leftarrow \emptyset$
for $c = 1$ to C **do**
  class c-specific average sensitivity score $\bar{s_c} = (\bar{s_c{}^1}, \bar{s_c{}^2}, ..., \bar{s_c{}^d}) \leftarrow {}^{\sum_{j=1}^{n_c} s_j}/_{n_c}$
  class c-specific unit sensitivity score $\tilde{s_c} \leftarrow ({}^{e^{\bar{s_c{}^1}}}/_{\sum_{k=1}^d e^{\bar{s_c^k}}}, {}^{e^{\bar{s_c{}^2}}}/_{\sum_{k=1}^d e^{\bar{s_c^k}}}, ..., {}^{e^{\bar{s_c{}^d}}}/_{\sum_{k=1}^d e^{\bar{s_c^k}}})$
  class c-specific radius weight $w_c \leftarrow {}^1/_{d\tilde{s_i}}$
  add (w_c, c) to D_{weight}
end for

5.1 Dimensional Radius Weight Calculation

To calculate the certified radius vector $(R_1, ..., R_d)$ of the input $x = (x_1, ..., x_d)$ while accounting for feature dimension correlations, we design to first calculate a real-value overall certified radius R using the randomized smoothing-based certification. This provides an equal robustness region size for all dimensions. Then, we weigh R according to the robustness contribution of each feature dimension to obtain the dimension-wise certified radius $R_i = w_i \times R$. The certified radius weight w_i is obtained through two steps: Dimensional Feature Sensitivity Analysis and Dimensional Radius Contribution Quantification, as detailed in Algorithm 1.

5.1.1 Dimensional Feature Sensitivity Analysis. In this step, we quantify the sensitivity of each dimension of the input feature vector x to the prediction score on the output class. For all dimensions, the more sensitive features are more likely to change the output results. Therefore, sensitive features are also important features for NID. We calculate a sensitivity score s_i for each dimension of the input sample x belonging to class c according to $s_i = {}^{d(f_\theta^c(x))}/_{d(x_i)}$, where i denotes the i-th dimension. Since our goal is to obtain a sensitivity score vector $s = (s_1, ..., s_d)$ corresponding to a specific category, we average the sensitivity scores of all samples belonging to the same category and denote the result as $\bar{s} = (\bar{s_1}, ..., \bar{s_d})$.

5.1.2 Dimensional Radius Contribution Quantification. In this step, we convert the average feature sensitivity score $\bar{s}$ into the contribution of the robustness of each dimension to the overall certified radius R of x, thereby proportionally allocating the overall certified radius to each dimension of the input vector. We first normalize the sensitivity score vector $\bar{s}$ to $\tilde{s} = (\tilde{s_1}, ..., \tilde{s_d}) = ({}^{e^{\bar{s_1}}}/_{\sum_{i=1}^d e^{\bar{s_i}}}, ..., {}^{e^{\bar{s_d}}}/_{\sum_{i=1}^d e^{\bar{s_i}}})$

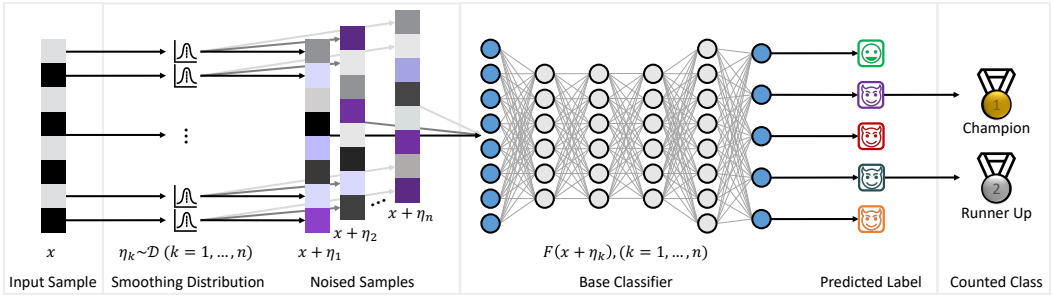


Fig. 4. Architecture of the smoothed network traffic classifier. Prediction: $n = n_{small}$, identify the champion class c_A that is predicted most times among n noised samples. Certification: $n = n_{large}$, count the number of noised samples predicted as c_A to estimate $P_A = \mathbb{P}(F_{smooth}(x) = c_A)$ and $P_B = \mathbb{P}(F_{smooth}(x) = c_B)$.

whose components sum to 1. Then, the dimensional robust radius contribution weight w_i is calculated according to Equation (9).

$$R_i = w_i \times R, w_i = \frac{R_i}{R} = \frac{1}{d\tilde{s}_i}, \tag{9}$$

where d is the number of dimensions in the input feature vector x, $^1/_d$ and R respectively denote the normalized sensitivity of a single dimension and the overall certified radius when assuming equal sensitivity across dimensions. Sensitivity and robustness proportions generally have an inverse relationship: higher sensitivity tends to correlate with lower robustness.

5.2 Multi-Order Robustness Certification

In MARS, robustness certification for NIDS starts from a smoothed network traffic classifier. In particular, to achieve a tight robustness guarantee, we adopt a two-step strategy: Smoothing Distribution Parameter Optimization and Gradient-based Certified Radius Calculation. First, we optimize the parameters of the smoothing distribution used for sampling noise η, making noised samples $x + \eta$ closer to the decision boundary. We then calculate the magnitude of the first-order gradient $||\nabla F^c_{smooth}(x)||_p$ of the smoothed network traffic classifier w.r.t x and expand the certified robust region along the direction in which the confidence score F^c_{smooth} increases.

5.2.1 Architecture of the Smoothed Network Traffic Classifier. The main difference between the smoothed network traffic classifier F_{smooth} and the base network traffic classifier F is that the predicted label of F_{smooth} on the input traffic x is the champion class c_A, which is the most often predicted class by the base classifier $F(x)$ across a set of noised samples $x + \eta$. The architecture of the smoothed network traffic classifier is shown in Figure 4. With the smoothed classifier, the MARS defended-NIDS can perform two procedures: Prediction and Certification.

Prediction. This procedure aims to determine the class by the smoothed classifier for the input x. It begins by choosing a smoothing distribution $\mathcal{D}$ with mean 0. Then, n_{small} (defaults to 100) noise vectors η are sampled and added to x to obtain n_{small} noised samples. The base classifier predicts them and identifies the champion class c_A and runner-up class c_B.

Certification. This procedure aims to calculate a l_p-bounded certified radius R. First, n_{large} (defaults to $10,000$) noises are randomly sampled from the smoothing distribution $\mathcal{D}$ and sequentially added to the input x to obtain n_{large} noised samples. Then, the number of these samples predicted as the champion class c_A is recorded as $n_A = \sum_{k=1}^{n_{large}} \mathbb{I}[F(x + \eta_k) = c_A]$. With n_{large} and n_A, the certified

Algorithm 2 Multi-Order Robustness Certification

Input: test samples $(x, y) \in D_{test}$, base classifier F with the classification function f_θ, smoothed classifier F_{smooth}
Output: overall certified radius R for each test sample

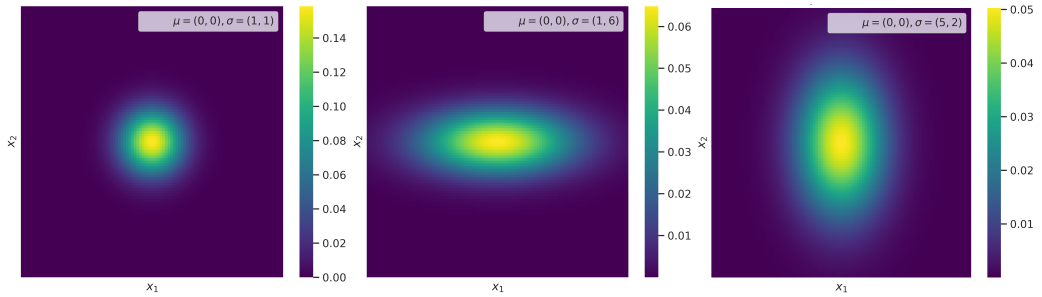
for $j = 1$ to $len(D_{test})$ **do**
  load one test example x_j, y_j;
  count champion class c_A according to Equation (5);
  sample n_{large} noise samples $\{\eta_1, ..., \eta_{n_{large}}\}$ from smoothing distribution $\mathcal{D}$;
  count $n_A \leftarrow \sum_{k=1}^{n_{large}} \mathbb{I}[F_\theta(x + \eta_k) = c_A]$;
  estimate $\underline{P_A}$ by LowerConfidenceBound(n_{large}, n_A, α) that estimates the interval $[\underline{P_A}, \overline{P_A}]$ where P_A holds;
  $\overline{P_B} \leftarrow 1 - \underline{P_A}$;
  if $\underline{P_A} < 0.5$ **then**
    output abstain certification;
  else if $\underline{P_A} \geq 0.5$ **then**
    calculate the certified radius R_{zero};
    estimate gradient of the smoothed classifier in the champion class dimension c_A on x: $\nabla F^{c_A}_{smooth}(x)$;
    calculate the magnitude of the gradient;
    if $||\nabla F^{c_A}_{smooth}(x)||_p \geq \varphi(R_{zero})$ **then**
      certified radius $R = R_{zero}$;
    else if $||\nabla F^{c_A}(x)||_p < \varphi(R_{zero})$ **then**
      certified radius $R = R_{first}$ calculated according to Equation (13);
    end if
    output R for test sample x.
  end if
end for

radius is then calculated using the zero-order output and first-order gradient of the smoothed classifier. The specific calculation process of the certified radius is shown in Algorithm 2.

5.2.2 Smoothing Distribution Parameters Optimization. The difference between a smoothed network traffic classifier and a smoothed image classifier is that the noise values η_i in each dimension of the noise vector η are sampled from a dimension-specific optimized distribution, where all dimensions of heterogeneous x are matched to the optimal smoothing distribution parameters ϑ.

Take the Gaussian distribution for example, before the optimization, the d-dimensional noise vector η is sampled from the multivariate standard Gaussian distribution $\mathcal{D}_{std} = \mathcal{N}(\mu = O, \sigma^2 = I)$, where O and I denote the d-dimensional full-zero vector and the full-one vector, respectively. After the optimization, the noise vector η is sampled from the optimized multivariate Gaussian distribution $\mathcal{D}_\vartheta = \mathcal{N}(\mu = O, \sigma^2 = \vartheta I)$. Parameters ϑ optimized dimensionally for the multivariate distribution facilitate an adaptive approach to the classification boundary in feature space. Figure 5 shows the PDF of the smoothing distribution with the same or different σ values across dimensions. The optimization of $\vartheta = \vartheta_{shape} \times \vartheta_{scale}$ involves two steps: distribution shape and scale optimization.

Distribution Shape Optimization. The aim is to optimize the vector parameter ϑ_{shape} in a multivariate distribution, keeping $\vartheta_{scale} = 1$. This encourages the sampling region of noised samples $x + \eta$ to be close to the decision boundary of the class predicted by the classifier F for x by



(a) Same σ across dimensions (b) Different σ across dimensions (c) Different σ across dimensions

Fig. 5. PDF of binary Gaussian distributions $\mathcal{N}(\mu, \sigma)$ with the same or different σ values across dimensions.

optimizing Equation (10), where $\vartheta = \vartheta_{shape} \times 1$ and $\mathbb{I}$ is the indicator function.

$$\begin{aligned} &\min_{\vartheta} \mathbb{E}_{x \sim D_{train}} \mathbb{I}[F(x + \eta_\vartheta) \neq F(x)] L(f(x + \eta_\vartheta), F(x)) \\ &-\mathbb{I}[F(x + \eta_\vartheta) = F(x)] L(f(x + \eta_\vartheta), F(x)). \end{aligned} \tag{10}$$

Distribution Scale Optimization. The optimization goal of the distribution scale, as defined in Equation (11), is to expand the coverage of the sampling area by adjusting the scalar parameter ϑ_{scale} of the multivariate distribution while maintaining the contour shape of the sampling area by fixing ϑ_{shape} to the optimized value ϑ^*_{shape}, so that the certified radius R can be as large as possible.

$$\begin{aligned} \max_{\vartheta} R &= \max_{\vartheta} \frac{\sigma}{2} (\Phi^{-1}(\underline{P_A}) - \Phi^{-1}(\overline{P_B})) \\ &= \max_{\vartheta} \frac{\vartheta}{2} (\Phi^{-1}(\mathbb{P}_{\eta \sim \mathcal{D}_{std}}(F(x + \vartheta\eta) = c_A)) - \Phi^{-1}(\mathbb{P}_{\eta \sim \mathcal{D}_{std}}(F(x + \vartheta\eta) = c_B))) \end{aligned} \tag{11}$$

5.2.3 Gradient-based Certified Radius Calculation. In this subsection, we focus on calculating the certified radius using the zero-order and first-order information together. The zero-order information we use is the statistical probability $P_A = \mathbb{P}(F_{smooth}(x) = c_A) = F^{c_A}_{smooth}(x)$ of the smoothed classifier when predicting x as the champion class c_A. The first-order information we use is the gradient magnitude $||\nabla F^{c_A}_{smooth}||_p$ of the F_{smooth}. The overall calculation process can be divided into two steps: Probability-based Radius Calculation and Gradient-based Radius Extension.

Zero-order Probability-based Radius Calculation. This step is to calculate the lower bound of the perturbation radius R that the smoothed classifier can tolerate on x based on $F^{c_A}_{smooth}(x)$, which is the estimated probabilities of the smoothed classifier predicting x as the champion class.

Suppose the champion class c_A is returned by F_{smooth} with probability $P_A = \mathbb{P}_{\eta \sim \mathcal{D}}(F(x + \eta) = c_A)$, and the runner-up class c_B is returned with probability $P_B = \mathbb{P}_{\eta \sim \mathcal{D}}(F(x + \eta) = c_B)$. We need to estimate the $\underline{P_A}$ and $\overline{P_B}$, which represent the lower bound of P_A and the upper bound of P_B, respectively. $\underline{P_A}$ is estimated like [5], using LowerConfidenceBound(n_{large}, n_A, α), which first calculates the interval $[\underline{P_A}, \overline{P_A}]$ where P_A holds with a probability of at least $(1-\alpha)$ for k-fold Binomial(n_{large}, P_A) sampling and then returns the left boundary of the interval. Then simply take $\overline{P_B} = 1 - \underline{P_A}$. The certified radius R_{zero} based only on the zero-order information is calculated like [5] as Equation (12).

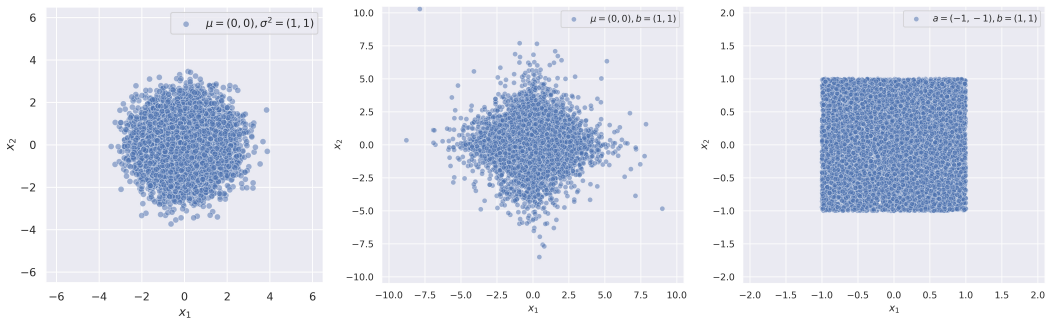
$$R_{zero} = \frac{\sigma}{2} (\Phi^{-1}(\underline{P_A}) - \Phi^{-1}(\overline{P_B})), \tag{12}$$

where Φ^{-1} is the inverse of the cumulative distribution function (CDF) of the standard Gaussian Distribution $\mathcal{N}(O, I)$. The size of R_{zero} is shown in the white solid line in Figure 1.

First-order Gradient-based Radius Extension. The goal of this step is to move and expand the certified robust region with radius R_{zero} along the gradient direction $\nabla F^{c_A}_{smooth}$ at x. Take a l_2-bounded radius as an example. We can see from Figure 1 that the gradient-based certification breaks the symmetry of the certified region centered at x and admits non-isotropic certified radius bounds. As the gradient direction reflects the area where the prediction confidence $F^{c_A}_{smooth}$ is higher than at the current data point x, moving the center x of the certified region along the gradient direction and exploring a larger radius is conducive to further expanding the original certified region.

Refer to [29], we obtain the final certified radius R by solving the system of simultaneous equations shown in Equation (13). Specifically, our goal is to reduce the length of the interval $[R_{low}, R_{high}]$ with an initial value of $[R_{low_0} = R_{zero},\ R_{high_0} = \varphi(^{(1+\underline{P_A})}/_2)]$ by binary searching, where φ is the PDF of the smoothing distribution. To this end, we continuously increase R_{low}, reduce R_{high}, and take the $r = {}^{(R_{low}+R_{high})}/_2$ as the certified radius which matches the requirement in Equation (13), where z_1 and z_2 are fixed. The search stops when $R_{high} - R_{low} \leq 0$.

$$\begin{aligned} \Phi(z_1 - R) - \Phi(z_2 - R) &= 0.5, \\ \Phi(z_1) - \Phi(z_2) \leq & F_{smooth}(x) = P_A,\ \varphi(z_2) - \varphi(z_1) \geq \sigma ||\nabla F^{c_A}_{smooth}(x)||_2 \end{aligned} \tag{13}$$



(a) Gaussian distribution $\mathcal{N}(\mu, \sigma)$ (b) Laplacian distribution $\mathcal{L}(\mu, b)$ (c) Uniform distribution $\mathcal{U}(a, b)$

Fig. 6. 10,000 noised samples drawn from different smoothing distributions respectively. (a) Gaussian distribution aligns the l_2-bounded region. (b) Laplacian distribution aligns the l_1-bounded region. (c) Uniform distribution aligns the l_∞-bounded region.

5.3 Smoothing Distribution Alignment

To compute the certified radius as tight as possible for the robust region under various types of l_p norm measures, it is crucial to choose an appropriate smoothing distribution type. Therefore, we empirically explore which smoothing distribution can align the shape of the high-probability sampling region of the noise data as closely as possible to the l_p-bounded certified robust region. After simulating the sampling areas of various probability distributions, the areas formed by 10,000 noise samples respectively sampled from three distributions can be seen in Figure 6.

5.3.1 l_2-bounded certified region alignment. Noise data from a Gaussian distribution form areas resembling a circle in 2D feature space, where the boundaries align closely with the constraints of the l_2 norm. This reflects the isotropic nature of the Gaussian distribution, as the magnitude of the noise is evenly distributed in all directions, resulting in smooth and symmetric contours. The l_2

norm emphasizes energy conservation, making this smoothing distribution suitable for scenarios requiring balanced noise allocation across dimensions.

5.3.2 l_1-bounded certified region alignment. Noise data from a Laplacian distribution form diamond-shaped areas in the 2D space. This shape arises from the distribution's characteristic of having a sharper peak at the center and heavier tails compared to Gaussian noise. The higher probability density near the center signifies that smaller perturbations are more likely, while the l_1 norm's emphasis on sparsity ensures that the noise components are distributed in a way that prioritizes deviations along fewer directions.

5.3.3 l_∞-bounded certified region alignment. Noise data from a Uniform distribution take on shapes akin to a square in the 2D space, where each dimension is independently constrained by the same maximum magnitude. This shape corresponds to the l_∞ norm, which focuses on controlling the largest component of the noise. The uniform distribution ensures that noise variations are evenly spread across all values within the bounds, capturing the robustness required for handling extreme outliers or adversarial perturbations.

Using an aligned smoothing distribution can encourage the spatial distribution of noise data to be closer to the shape of the l_p-bounded certified region under the same amount of sampling noise, thereby facilitating the search for the perturbed sample that can change the model prediction results but are difficult to find based solely on the commonly used Gaussian smoothing distribution.

6 Experimental Setup

In this section, we introduce the testbed (Section 6.1), model architecture (Section 6.3), dataset preprocessing (Section 6.4), attack settings (Section 6.5), certified defense baselines (Section 6.2), and evaluation metrics (Section 6.6) for the experimental evaluation. The code for MARS has been released at https://github.com/CertNID/MARS.

Table 1. Comparison of certified defense methods

Method	Heterogeneity	Universality	Robustness Guarantee Diversity			Evasion Attacks			Natural Corruptions	
			l_2 Radius	l_1 Radius	l_∞ Radius	l_2 Attack	l_1 Attack	l_∞ Attack	Latency	Loss
VRS [5]	○	●	●	○	○	○	○	○	○	○
FRS [29]	○	●	●	●	●	○	○	○	○	○
BARS [45]	●	●	●	○	○	○	○	●	○	○
MARS	●	●	●	●	●	●	●	●	●	●

6.1 Testbed

The method was implemented using PyTorch 2.0.1 and SciPy 1.11.2 [42]. Each experiment was repeated three times with random seeds {42, 43, 44} on an NVIDIA GeForce RTX 3090 GPU with CUDA 11.7, and the average results are reported.

6.2 Certified Defense Baselines

To ensure a fair performance comparison, we selected three state-of-the-art (SOTA) robustness certification methods known for their good architecture-level scalability as certified defense baselines. These methods rely on randomized smoothing, allowing for flexible robustness certification applicable to various NIDS architectures. Table 1 illustrates their properties.

6.2.1 Vanilla Randomized Smoothing (VRS). VRS [5] is the first randomized smoothing-based robustness certification technique designed for image classifiers based on homogeneous inputs. It uses Gaussian noise and the Monte Carlo sampling method to obtain the l_2-measured radius only through the zero-order information.

6.2.2 First Order-based Randomized Smoothing (FRS). Since zero order-based methods so far still have some gaps between the actual robust radius and the available certified radius, FRS [29] (2020) was proposed to provide certified robustness for image classifiers based on homogeneous inputs by using the first-order gradient of the smoothed classifier, which further tightens the l_p radius.

6.2.3 Boundary-Adaptive Randomized Smoothing (BARS). BARS [45] is the only existing robustness certification framework for NIDS based on heterogeneous inputs, which is built on top of VRS. It focuses on zero-order information and adapts to the network traffic features with dimension-wise optimized smoothing distribution but only provides l_2 robustness guarantees.

6.3 Model Architectures

We evaluated two advanced DNN-based NIDS architectures, CADE [52] and ACID [6], which feature different detection model structures and specialized functionalities, to assess the certified defense performance of MARS and other certified defense methods.

6.3.1 Contrastive Autoencoder for Drifting detection and Explanation (CADE). CADE is a concept drift model trained on $n-1$ classes and tested on n classes to detect unknown samples. It employs an encoder-decoder model architecture with a monitoring system to analyze the relationship between input data and training data. It is well-suited for scenarios where the nature and attributes of observed samples may change compared to the knowledge acquired during training.

6.3.2 ACID (Adaptive Clustering-based Intrusion Detection). ACID is a multi-classification detection model that integrates unsupervised and supervised learning. It identifies input from n categories through the learning of n classes during the training phase. ACID's strength lies in its utilization of traffic features obtained via a clustering-based representation learning approach. This enables the expression of more comprehensive sample information compared to manually defined traffic features, particularly for high-dimensional data.

Table 2. Information on network intrusion detection datasets used for evaluation.

Dataset	CSE-CIC-IDS-2018-CADE				CSE-CIC-IDS-2018-ACID			
	DoS-Hulk-Drift Dataset		Infiltration-Drift Dataset		Diverse-Intrusions Dataset		Similar-Intrusions Dataset	
	Class	Number	Class	Number	Class	Number	Class	Number
Training	Benign	52996	Benign	52996	Benign	52996	Benign	52996
	SSH-Bruteforce	9385	SSH-Bruteforce	9385	FTP-Bruteforce	12590	DoS-GoldenEye	26565
	Infiltration	7390	DoS-Hulk	34789	DDoS-HOIC	53476	DoS-SlowHTTPTest	11191
	-	-	-	-	Bot	22584	DDoS-LOIC-HTTP	46095
Test	Benign	13249	Benign	13249	Benign	13249	Benign	13249
	SSH-Bruteforce	2346	SSH-Bruteforce	2346	FTP-Bruteforce	3148	DoS-GoldenEye	6641
	Infiltration	1894	DoS-Hulk	8697	DDoS-HOIC	13369	DoS-SlowHTTPTest	2798
	DoS-Hulk	43486	Infiltration	9327	Bot	5646	DDoS-LOIC-HTTP	11524

6.4 Datasets and Preprocessing

Following and extending the dataset settings in the leading work BARS [45], we evaluated the performance of MARS on two NIDS architectures using four sub-datasets, all of which are derived from the CSE-CIC-IDS-2018 dataset [41], as shown in Table 2.

For the construction of each sub-dataset, we not only preprocessed the data in the original CSE-CIC-IDS-2018 dataset to filter out duplicate samples, samples with invalid timestamps, and samples with infinite values, but also *digitized* and *normalized* the raw feature values, including one-hot encoded categorical features. This process mapped the network traffic vector, which includes both discrete and continuous features, into a space that can be measured by the l_p norm. For each dataset, the split ratio of training samples and test samples is $8:2$.

6.4.1 CSE-CIC-IDS-2018-CADE. For CADE, we used traffic samples in the CSE-CIC-IDS-2018 dataset belonging to the Benign class and three malicious categories (SSH-Bruteforce, DoS-Hulk, and Infiltration). Specifically, we used one day's traffic of Benign (02/14), SSH-Bruteforce (02/14), DoS-Hulk (02/16), and Infiltration (03/01) to populate the dataset. Due to the amount of samples in the original dataset, for each class, we collected 10% of the samples in the original dataset.

Since CADE is a NIDS that supports concept drift detection, the test set must contain at least one unseen category in addition to the sample categories that the CADE model has seen during the training phase. To this end, according to different offset categories, the CSE-CIC-IDS-2018-CADE dataset is further divided into the *DoS-Hulk-Drift dataset* where DoS-Hulk appears only in the test set and the *Infiltration-Drift dataset* where Infiltration appears exclusively in the test set.

6.4.2 CSE-CIC-IDS-2018-ACID. For ACID, we used samples in the CSE-CIC-IDS-2018 dataset belonging to the Benign class and six malicious classes, including FTP-Bruteforce, DDoS-HOIC, Bot, DoS-GoldenEye, DoS-SlowHTTPTest, and DDoS-attacks-LOIC-HTTP. These data were divided into *Diverse-Intrusions dataset* with diverse intrusions, and *Similar-Intrusions dataset* with similar intrusions (See Table 2), used for conventional and fine-grained multi-class detection, respectively.

Specifically, we used one day's traffic of Benign (02/14), FTP-Bruteforce (02/14), Bot (03/02), and DDoS-HOIC (02/21) to populate the Diverse-Intrusions dataset. For the FTP-Bruteforce class, we collected 40% of the samples in the original dataset. For each other class, we collected 10% of the samples in the original dataset. For the Similar-Intrusions dataset, we have Benign (02/14), DoS-GoldenEye (02/15), DoS-SlowHTTPTest (02/16), and DDoS-attacks-LOIC-HTTP (02/20). For the DoS-GoldenEye class, we used 80% of the samples in the original dataset. For each other class, we collected 10% in the original dataset.

6.4.3 One-hot Encoding of Categorial Features. For the categorial features, we have one-hot encoded them. Destination Port features have been encoded to 0, 1, and 2, with 0 for the low-frequency port (< 1000), 1 for medium ($1000 \sim 10000$), and 2 for high (> 10000). For the Protocol feature, we encoded '0' to 0, '6-TCP' to 1, and '17-UDP' to 2. Note that 0, 1, and 2 means index where the 1 occupies rather than a number. For example, 0 for $[1, 0, 0]$. Furthermore, timestamp features were converted to UNIX seconds format before normalization and MinMax scaling.

6.5 Attack Configuration

Two threat models were implemented as outlined in Section 4.1, with the following configurations.

6.5.1 Parameters in Evasion Attack. We employ two types of white-box evasion attacks: Projected Gradient Descent (PGD) [28] and Elastic-Net Attack to DNN (EAD) [3]. For l_2-PGD, l_1-PGD, and l_1-EAD attacks, the perturbation budget ϵ is set to 1.0, allowing for a maximum adversarial perturbation that can significantly alter the input. Additionally, the per-step perturbation budget ϵ_s is configured

to 0.75, which restricts the perturbation at each iteration to ensure controlled modifications of the input. For the l_∞-PGD attack, the parameters are adjusted to accommodate its unique characteristics: the perturbation budget ϵ is set to 0.2, and the per-step perturbation budget ϵ_s is limited to 0.1. l_p-PGD is typically most powerful under the L_∞ norm constraint, so the perturbation budget required to achieve evasion is smaller. In all cases, the maximum number of iterations, $N_{\text{iteration}}$, is uniformly set to 20, providing a consistent framework for evaluating the effectiveness of each attack method.

6.5.2 Parameters in Natural Corruption. Perturbable features under two natural corruption threats are shown in Figure 7. For Latency, we use a Gaussian distribution with a standard deviation of 1 and a mean of 0 to add random noise to the time-related traffic features, such as the time between two flows, time between two packets, time a flow was active before becoming idle, time a flow was idle before becoming active, etc. For PacketLoss, we use the same Gaussian distribution to add random noise to packet quantity-related features and partial length-related features correlated with the packet number, such as the number of total packets, number of packets transferred per second, number of packets bulk rate, number of packets in a sub-flow, etc.

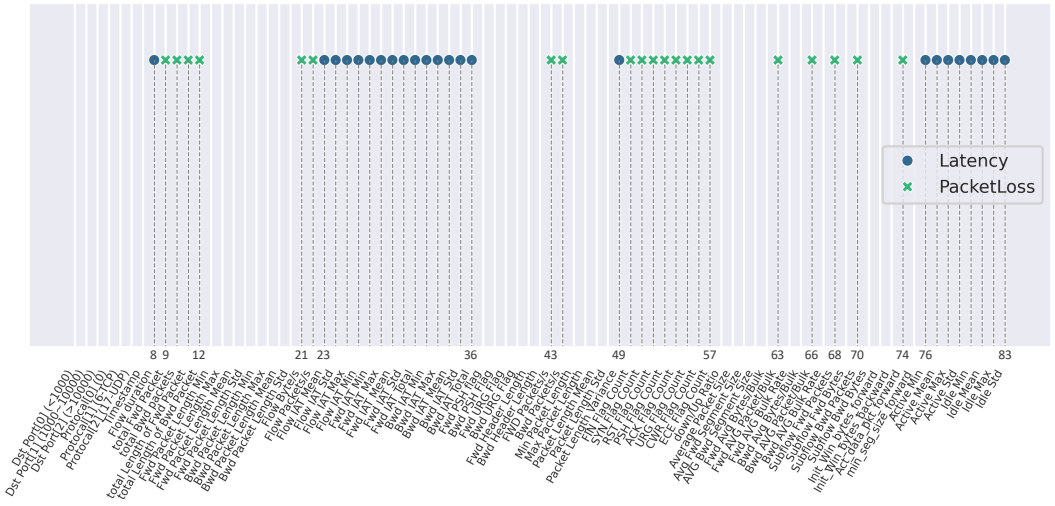


Fig. 7. Network traffic features perturbed under natural corruptions.

6.6 Evaluation Metrics

Multiple metrics are used to assess certified robustness against potential perturbations, empirical robustness against adversarial and corrupted examples, and regular performance on clean examples.

6.6.1 Certified Robustness. We use two metrics to evaluate the certified robustness of the model.

Mean Certified Radius. The average certified radius per class is calculated as in Equation (14).

$$Mean\ Certified\ Radius\ (MCR) = \frac{1}{N}\sum_{i=1}^{N} R_i, \tag{14}$$

where N represents the total number of test samples belonging to the same class, and R_i denotes the certified radius for each sample. The MCR is calculated using all samples within the same class to assess the certified robustness of the detector across different categories. This approach enables

evaluation of the model's certified robustness against evasion attacks towards each class. A larger value of *MCR* indicates a tighter robustness guarantee, indicating that the model can maintain its performance with a greater margin of robustness when faced with any possible perturbations.

Certified Accuracy. Given a certified radius threshold R_{given}, the certified accuracy measures the proportion of test samples that are correctly predicted by the certified defended classifier F_{smooth} with a certified radius R greater than R_{given}, as shown in Equation (15).

$$Certified\ Accuracy\ (CerAcc) = \frac{N_{(F_{smooth}(x)=y_{true})\&(R\geq R_{given})}}{N}, \tag{15}$$

where $N_{(F_{smooth}(x)=y_{true})\&(R\geq R_{given})}$ counts only those correctly predicted samples that also satisfy the condition $R \geq R_{given}$, N indicates the total number of test samples for certification evaluation. A higher certified accuracy reflects a greater number of samples passing the robustness certification under the specified threshold R_{given}. By tracking certified accuracy, we can assess the effectiveness of a DNN model in maintaining correct predictions while satisfying certified robustness.

6.6.2 Empirical Robustness. We use robust accuracy to assess the empirical robustness of the model against evasion attacks and natural corruptions.

Robust Accuracy. Robust accuracy reflects the proportion of perturbed test samples x^* (e.g., adversarial examples or corrupted examples) that the model predicts correctly in all perturbed test samples, as expressed in Equation (16).

$$Robust\ Accuracy\ (RobAcc) = \frac{N_{(F_{smooth}(x^*)=y_{true})}}{N} = \frac{TP+TN}{TP+TN+FP+FN}, \tag{16}$$

where, $N_{(F_{smooth}(x^*)=y_{true})}$ denotes the number of perturbed samples correctly predicted, and N denotes the total number of perturbed test samples for empirical robustness evaluation. Specifically, TP is True Positives (malicious samples correctly classified), TN is True Negatives (benign samples correctly classified), FP is False Positives (benign samples incorrectly classified), and FN is False Negatives (malicious samples incorrectly classified). We use robust accuracy to evaluate the empirical robustness of detectors on four evasion attacks, including l_2-PGD, l_∞-PGD, l_1-PGD, and l_1-EAD, as well as two natural corruptions, including PacketLoss and Latency.

When evaluating adversarial examples, robust accuracy is equivalent to Recall, as defined in Equation (17). This is because the adversarial test set consists solely of adversarial malicious traffic, as detailed Section 4.1.1, resulting in TN and FP values of zero.

$$Recall = \frac{TP}{TP+FN} \tag{17}$$

For evaluations on natural corrupted examples, where the corrupted test set contains both benign and malicious traffic as described in Section 4.1.2, robust accuracy is measured as the ratio of correctly predicted samples among all perturbed test samples. Also, metrics like Recall, Precision, False Positive Rate (FPR), and False Negative Rate (FNR) are calculated on the corrupted test data.

Precision. It denotes the proportion of TP among all positive predictions, as in Equation (18).

$$Precision = \frac{TP}{TP+FP}. \tag{18}$$

False Positive Rate (FPR). It is also known as False Alarm Rate, measures the proportion of negative test samples (such as benign) that are incorrectly classified as positive samples (such as malicious) in all negative test samples, calculated as Equation (19).

$$FPR = \frac{FP}{FP+TN}. \tag{19}$$

False Negative Rate (FNR). It quantifies the proportion of positive test samples that are incorrectly classified as negative in all positive test samples, defined as Equation (20).

$$FNR = \frac{FN}{FN + TP}. \tag{20}$$

6.6.3 Regular Performance. We use Clean Accuracy to evaluate the regular predictive performance of the model on clean data without perturbation. We also evaluate metrics such as Recall, Precision, F1-score, FPR, and FNR on the clean test data.

Clean Accuracy. It is the ratio of correctly predicted clean test samples among all clean test samples, as defined in Equation (21):

$$Clean\ Accuracy\ (CleAcc) = \frac{N_{(F_{smooth}(x)=y_{true})}}{N}. \tag{21}$$

7 Evaluation Results and Analysis

In this section, we first compare the performance of relying on randomized smoothing across several aspects. These include horizontal comparisons of the tightness of l_2 certified robustness guarantees (Section 7.1), l_1 and l_∞ robustness guarantees tightness (Section 7.2), empirical robustness against evasion attacks (Section 7.3), empirical robustness against natural corruptions (Section 7.4), as well as certified and empirical robustness in fine-grained intrusion detection (Section 7.5). Then, we present the vertical analysis of dimension-wise certified robustness (Section 7.6), the tightness of l_1 and l_∞ certified robustness guarantees with different smoothing distributions (Section 7.7), and the certification time cost of the proposed method (Section 7.8).

7.1 Comparison of l_2-bounded Certified Robustness with SOTA Methods

We first compare the tightness of the l_2 robustness guarantees provided by MARS with VRS [5] and FRS [29] for the image domain, and BARS [45] for the network traffic domain. For a fair comparison, we used the same dataset and settings as BARS. We calculated the MCR and certified accuracy (defined in Section 6.6) for each category on ACID and CADE.

7.1.1 Setup. To be comparable with VRS and FRS, which do not consider dimension-wise radius, the object we compare is the overall certified radius R of the smoothed model. For the smoothed classifier, n_{small} and n_{large} are set to 100 and 10,000. The parameters for the smoothed classifier are: n_{small} =100, n_{large} =10,000. The learning rate for optimizing the noise shape is set to 0.01. The maximum number of training epochs is limited to 10. The failure probability α for radius calculation is set to 0.001. To be consistent with the evaluation setup in BARS, MCR and certified accuracy are measured by category for each test set.

7.1.2 Results. The results of certified accuracy (see Figure 8) and MCR (see Figure 9) show that MARS always outperforms certified defense baselines. Especially on the CADE-Infiltration-Drift dataset and the CSE-CIC-IDS-2018-ACID dataset, MARS showed significant advantages over BARS when both VRS and FRS failed certification in many categories. Also, for the SSH-Bruteforce category in the CADE-DoS-Hulk-Drift dataset, we observe that the certified radius obtained by all methods is always 0, which indicates that CADE itself is very sensitive and vulnerable to the SSH-Bruteforce attack in the CADE-DoS-Hulk-Drift dataset, leading to failure to certify.

A certified radius equal to 0 indicates that the detection model is vulnerable to the specific attack category. Even if the clean sample is perturbated very weakly, that is, the noise budget is very small, it is easy for the model to lose confidence in the prediction results, leading to certification failure,

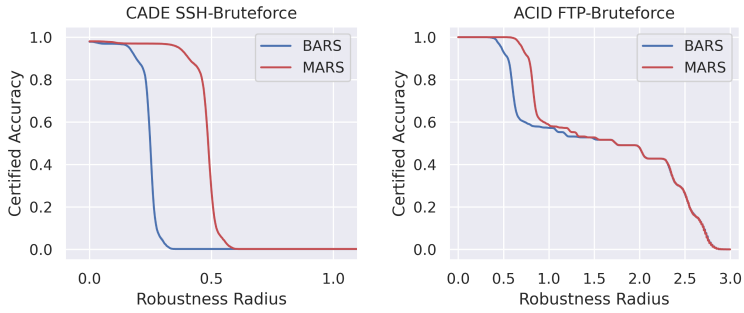


Fig. 8. Comparison of certified accuracy of l_2 robustness guarantee.

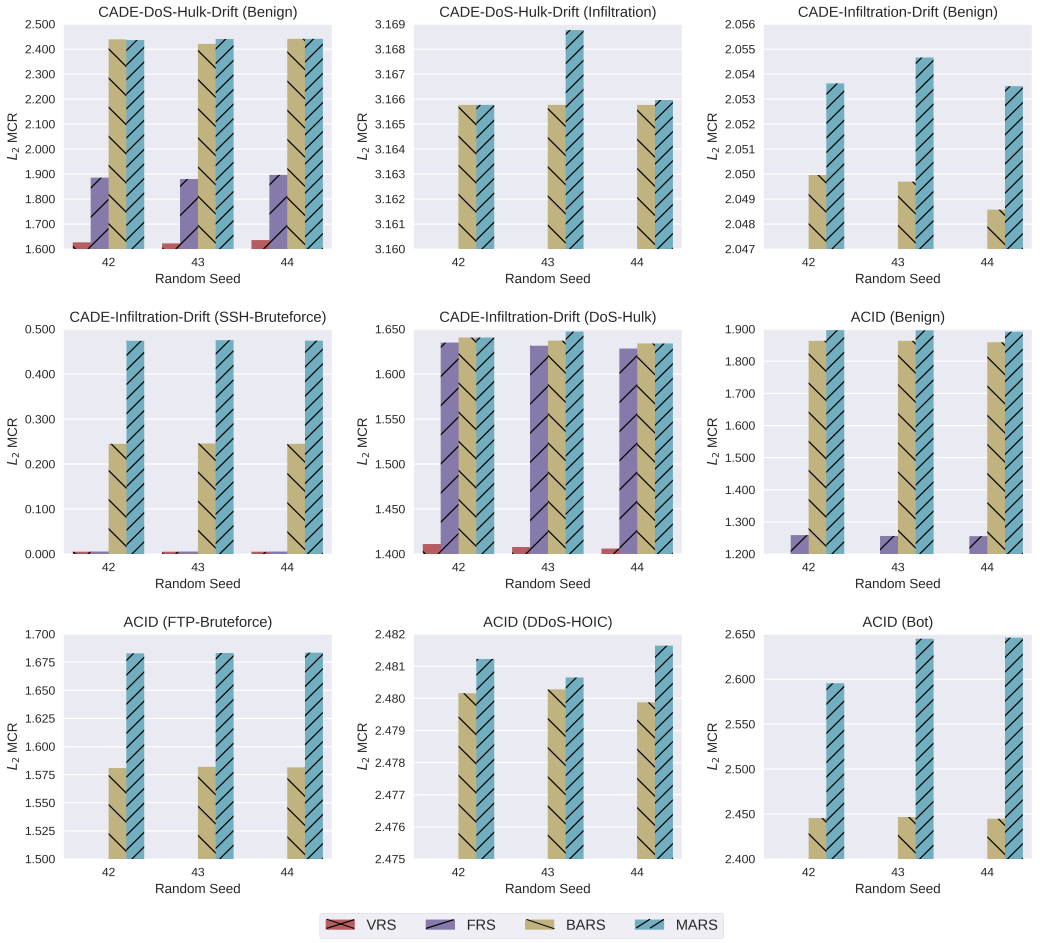


Fig. 9. Comparison of l_2 mean certified radius for each class.

that is, getting a radius equal to 0. For the CADE models trained with VRS or FRS on the CADE-DoS-Hulk-Drift dataset, SSH-Bruteforce and Infiltration are attacks that are relatively difficult to detect. For the CADE trained on the CADE-Infiltration-Drift dataset, benign is a vulnerable category.

For the ACID, the detection of FTP-Bruteforce, DDoS-HOIC, and Bot attacks is unstable. Fortunately, BARS and MARS can not only enhance the certified robustness against those attacks model is originally robust againt, but also increase the certified radius in these vulnerable classes, and the improvement brought by MARS has always been greater than BARS. Furthermore, we observed that CADE trained based on the CADE-Infiltration-Drift dataset has reached the consistent conclusion that it is difficult to pass certification in the SSH-Bruteforce category with all certification methods. This is due to the inherent vulnerability of the CADE model obtained under this training setting, because CADE trained on another dataset performs normally on the same class.

7.2 Comparison of Various l_p-bounded Certified Robustness with SOTA Methods

To assess the tightness of l_p robustness guarantees across different norms, we compare the sizes of l_2, l_1, and l_∞ certified radii with the leading method FRS [29], since neither VRS nor BARS supports l_1 and l_∞ robustness certification.

7.2.1 Setup. FRS incorporates l_2, l_1, and l_∞ guarantees but relies exclusively on standard Gaussian distribution for smoothing. For a fair comparison, we compare MARS with FRS using Gaussian smoothing distribution across all l_p norms, without distribution alignment.

7.2.2 Results. The results of MCR (see Figure 10) show that MARS consistently provides tighter l_p robustness guarantees compared to FRS. Especially when FRS fails certification on many classes (with radius 0) due to its nature of smoothing all traffic feature dimensions indiscriminately, MARS still outputs non-trivial l_2, l_1 and l_∞ radii. Furthermore, the results of certified accuracy (see Figure 11) show that l_2 and l_1 radii are close, but the l_∞ radius remains the smallest, as l_∞ is the most difficult to capture by the Gaussian distribution. We can see that if there is a lack of customized certification design for network traffic data, even the FRS that also uses first-order information will have a hard time effectively certifying the robust radius of the attack samples for the detectors. Meanwhile, we also observed that compared with l_2 and l_1 certification, it is more difficult to obtain a tight l_∞ robustness guarantee, that is, a large l_∞ certified radius. This is reasonable because under the same budget $||\delta||_p < R$, the l_∞-measured certified region will be larger in volume than l_2 and l_1, thus it is easier for samples to evade detection.

7.3 Comparison of Empirical Robustness against Evasion Attacks with SOTA Methods

Considering the diversity of adversary changes, unlike BARS, which only tests against l_∞ evasion attacks, we use various l_p norms to enrich the threat model. We employ Projected Gradient Descent (PGD) [28] and Elastic-Net Attack to DNN (EAD) [3] as threat models of white-box evasion attacks to generate adversarial examples $x^* = x + \delta$.

7.3.1 Setup. Following the BARS settings, we selected one of the malicious categories, Bot, as a representative to test whether the ACID model with certified defense can correctly identify adversarial Bot samples, and calculated the robust accuracy and clean accuracy (as defined in Section 6.6). For l_2-PGD, l_1-PGD, and l_1-EAD, perturbation budget ϵ that determines the maximum adversarial perturbation is set to 1.0 and per-step perturbation budget ϵ_s that determines the maximum allowed perturbation at each iteration is set to 0.75. For l_∞-PGD, ϵ is 0.2 and ϵ_s is 0.1. The maximum number of iterations $N_{iteration}$ is set to 20 for all attacks.

7.3.2 Results. Since we only measure the robust accuracy of the model against adversarial malicious examples, robust accuracy here is equivalent to the Recall. The results of robust accuracy on

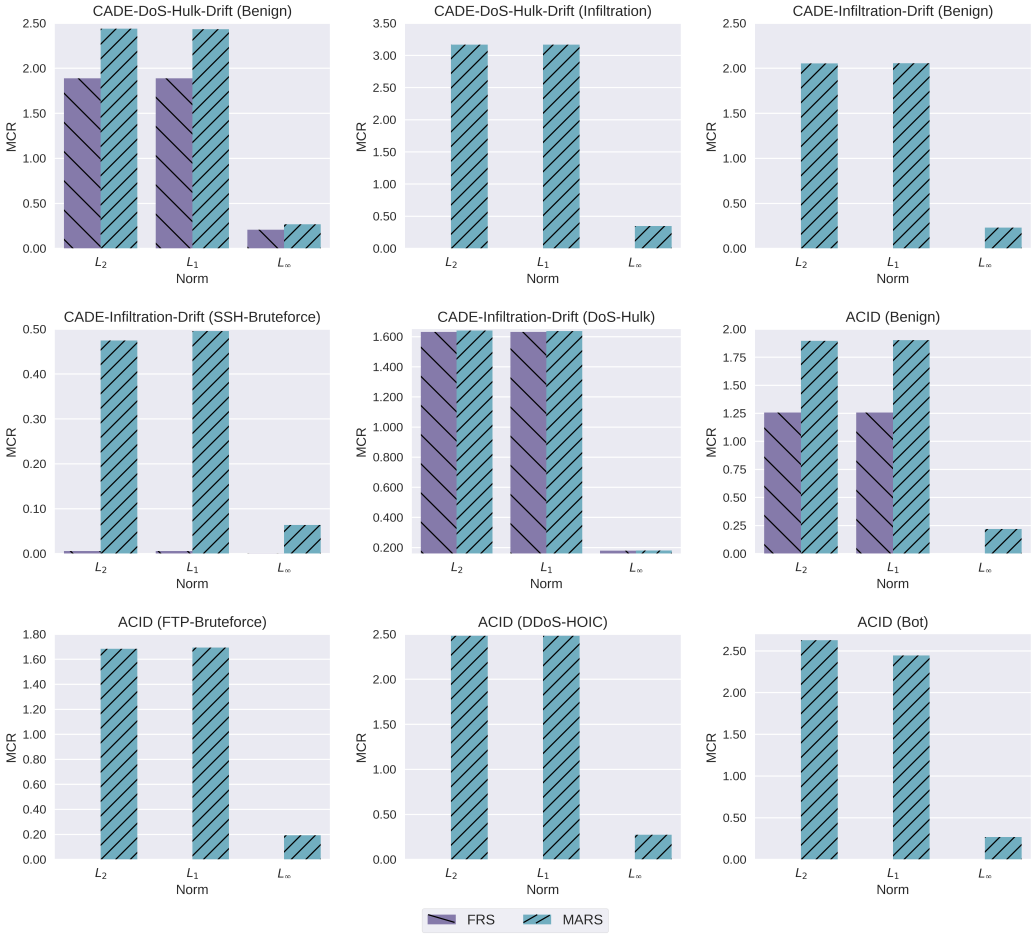


Fig. 10. Comparison of l_p mean certified radius under the same smoothing distribution.

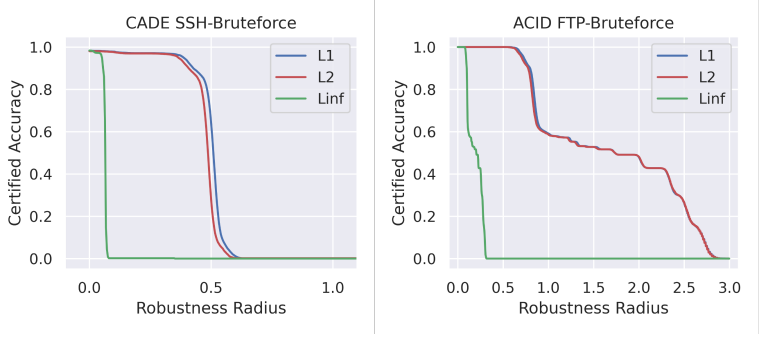


Fig. 11. Comparison of l_p certified accuracy under the same smoothing distribution.

adversarial examples (see Table 3) show that compared to BARS, our defense boosts the robust

Table 3. Comparison of empirical robustness of ACID against different evasion attacks

Method	Clean Accuracy (%)	Robust Accuracy (%) on Adversarial Examples		
		l_2-PGD	l_∞-PGD	l_1-EAD
Vanilla	100.00±00.00	83.95±00.00	55.02±00.01	00.27±00.00
BARS [45]	100.00±00.00	96.04±00.05	81.78±00.20	00.16±00.01
MARS	100.00±00.00	**97.74±00.13**	**88.95±00.31**	**10.28±00.06**

accuracy against evasion attacks by 1.70% for l_2-PGD, 7.17% for l_∞-PGD, and 10.11% for l_1-EAD. Compared to the base detector without any certified defense (noted as Vanilla), robust accuracy increases by 13.79% for l_2-PGD, 33.94% for l_∞-PGD, and 10.01% for l_1-EAD. Notably, we also observe that the base ACID detector itself is already very robust to l_1-PGD attacks, and both BARS and MARS preserve this robustness. Thus, we tested the more powerful l_1-EAD, essentially a linear mixture of l_1 and l_2 penalty functions. Neither Vanilla nor BARS can resist l_1-EAD, and only MARS enhances model robustness against it. Also, it can be seen from the table that both BARS and MARS well retain the detection accuracy of the vanilla ACID model on clean Bot samples, reaching 100%.

7.4 Comparison of Empirical Robustness against Natural Corruptions with SOTA Methods

Besides adversarial examples, natural corruptions from changes in the cyber environment can also lead to model misclassification. We generate naturally corrupted samples from clean benign and malicious inputs using Latency and PacketLoss, as detailed in Section 6.5.

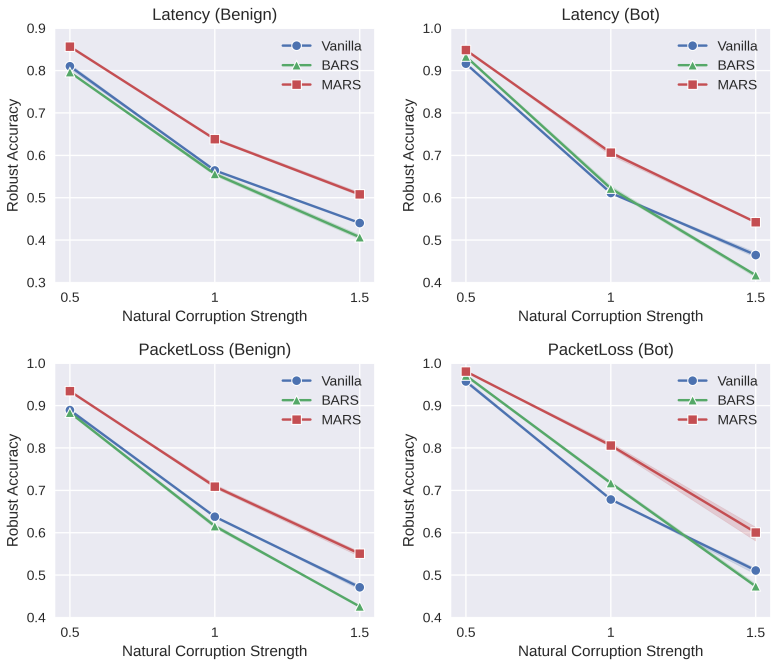


Fig. 12. Comparison of empirical robustness of ACID against varied natural corruptions.

7.4.1 Setup. Since natural corruption noise generally does not follow specific patterns, using an empirical distribution would make more sense. In our study, distribution shifts in the feature dimensions related to packet arrival time and packet number are simulated using random noise following a Gaussian distribution with mean 0. Particularly, we adjust the standard deviation σ in $\{0.5, 1.0, 1.5\}$ to mimic the different corruption strengths.

7.4.2 Results. The results of Robust Accuracy on corrupted samples (see Figure 12) show that we can see that MARS consistently outperforms BARS and Vanillia across various corruption intensities and certified classes. Also, under the same corruption strength, both vanilla and certified defended models show higher resilience to PacketLoss than to Latency for benign and malicious traffic. This suggests that ACID is more sensitive to perturbations in time-related features and more robust in quantity-related features. It can be seen from the consistent results that MARS always has the highest robust accuracy for increasing natural disturbances. Compared with the enhanced robustness from MARS, BARS weakens the detector's ability to identify natural corruptions.

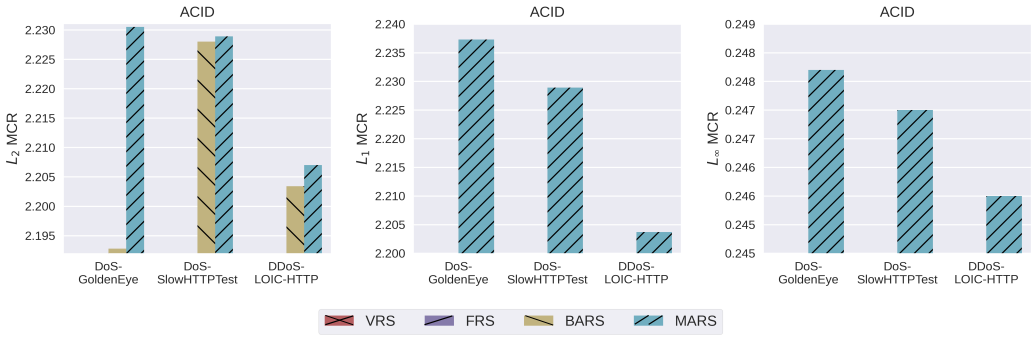


Fig. 13. Comparison of l_p mean certified radius of ACID in fine-grained detection.

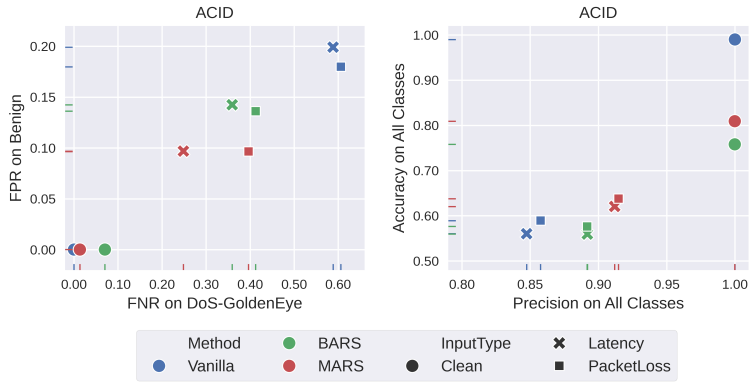


Fig. 14. Comparison of empirical robustness of ACID against natural corruptions in fine-grained detection.

7.5 Comparison of Robustness in Fine-grained Detection with SOTA Methods

Even with similar intrusion purposes, attackers often employ varied techniques and tools. Considering the differences among attackers, we assessed the fine-grained detection performance of MARS on datasets containing similar network intrusion types.

7.5.1 Setup. This evaluation includes various DoS-related attacks to extend performance assessment in fine-grained NIDS scenarios. Using the Similar-Intrusions Dataset involving benign (as shown in Table 2), we trained a four-class ACID model with 99% accuracy, 100% precision, and an F1-score of 1 on the clean test set in 312 seconds. Multiple l_p-bounded certified radii, as well as FPR, FNR, precision, and accuracy on clean and perturbed samples were tested.

7.5.2 Results. We compared l_2, l_1, and l_∞ radius with other randomized smoothing methods. Analogous to the previous cases, MARS achieved the largest certified radii in all three similar intrusions chosen (DoS-GoldenEye, DoS-SlowHTTPTest, and DDoS-attacks-LOIC-HTTP). The results of MCR (see Figure 13) show that MARS achieves the largest MCR across all l_p norms: l_2 radius increased by 9×10^{-4} to 3.77×10^{-2} compared to BARS, l_1 by 7.24×10^{-1} to 2.2373 and l_∞ by 2.20×10^{-1} to 2.48×10^{-1} compared to FRS.

Furthermore, we observe that MARS outperforms BARS in defending the vanilla classifier against Latency and PacketLoss corruption, improved accuracy by 6.12% under Latency and PacketLoss corruptions (See Figure 14). False Positive Rate (FPR) on corrupted Benign samples decreased by 4.27%, and False Negative Rate (FNR) on corrupted GoldenEye decreased by 6.35%. With MARS defense, the classifier achieves higher precision with corruption on both Latency and PacketLoss. This improvement is consistent across classes, as seen in Benign and DoS-GoldenEye examples.

Table 4. Top-5 Sensitive and robust features on DoS-GoldenEye

No	Radius	FeatureName	Description
24	0.0426	Flow_IAT_Std	Standard deviation time two flows.
20	0.0433	Bwd_Packet_Length_Std	Standard deviation size of packet in backward direction.
79	0.0488	Active_Std	Standard deviation time a flow was active before becoming idle.
72	0.0569	Init_Win_bytes_forward	Number of bytes sent in initial window in the forward direction.
78	0.0576	Active_Max	Maximum time a flow was active before becoming idle.
8	10.0741	Flow_Duration	Flow duration.
39	10.9644	Fwd_URG_Flag	Number of times URG flag was set in packets travelling in the forward direction (0 for UDP).
52	11.2367	RST_Flag_Count	Number of packets with RST.
38	11.3300	Bwd_PSH_Flag	Number of times PSH flag was set in packets travelling in the backward direction (0 for UDP).
13	11.4358	Fwd_Packet_Length_Min	Minimum size of packet in forward direction.
All	2.2305	MCR	Mean certified radius per class.

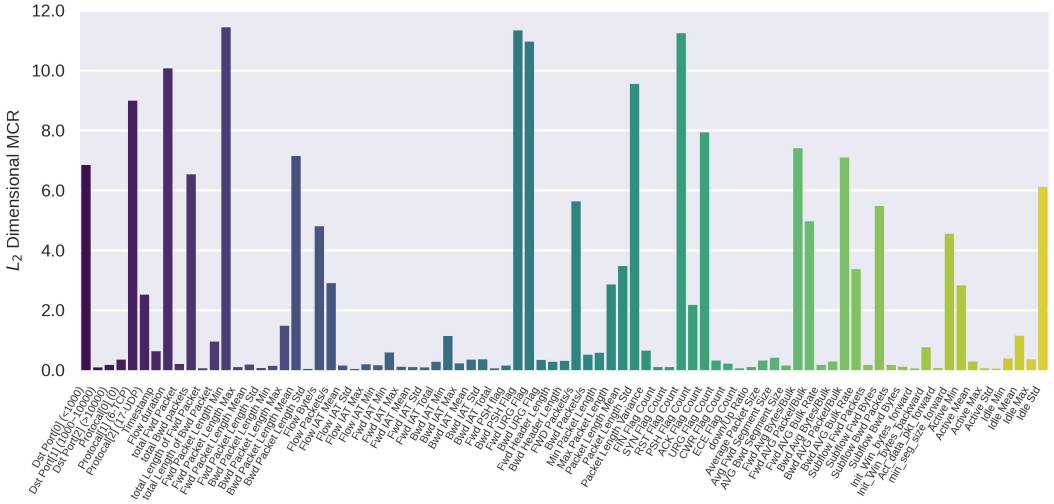


Fig. 15. l_2 dimensional mean certified radius of ACID on DoS-GoldenEye.

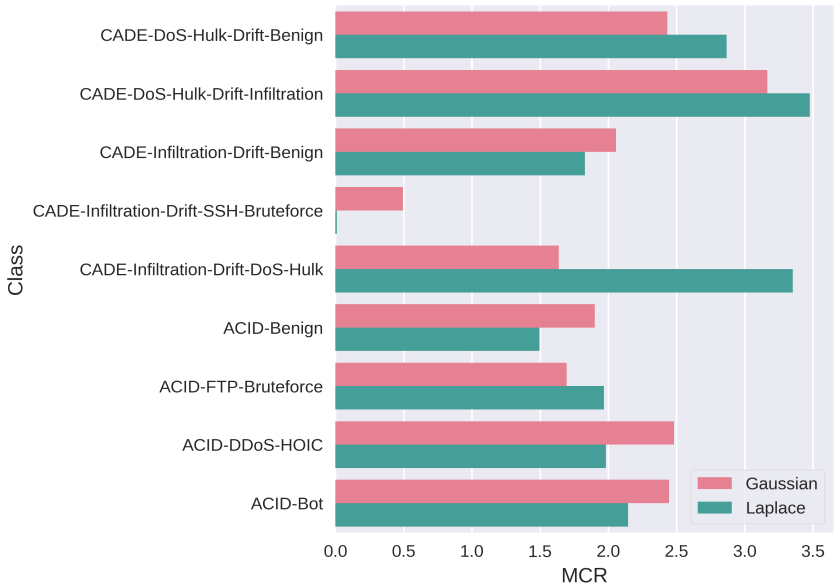


Fig. 16. l_1 Mean Certified Radius (MCR) under different smoothing distributions

7.6 Analysis of Dimensional Certified Robustness

The certified radius presented in the previous sections is the overall radius, which is derived from a weighted average of dimension-wise certified radii. Analyzing the dimensional radius R_i can help intrusion detection participants identify sensitive (important) and insensitive (robust) features, informing model feature-specific robustness.

7.6.1 Setup. We evaluate the l_2-bounded dimensional radius R_i of the ACID obtained by MARS on the GoldenEye sample and listed the top-5 and bottom-5 dimensional radius rankings. A smaller dimensional radius indicates greater sensitivity and importance of the input feature dimension to the NIDS, as perturbations to such features are more likely to alter the predictions of the detector.

7.6.2 Results. The dimensional MCR results across all feature dimensions are presented in Figure 15, with the top-5 and bottom-5 certified radius rankings detailed in Table 4. In the DoS-GoldenEye attack, the model demonstrates greater sensitivity to inter arrival time (IAT)-related features while showing greater robustness to forward packet length-related features. This finding is consistent with the previous observations that the vanilla ACID model exhibited significantly reduced accuracy and increased FNR on Latency-corrupted GoldenEye samples, as discussed in Section 7.5.

7.7 l_p-bounded Certified Robustness with Different Smoothing Distributions

VRS, FRS, and BARS all use the Gaussian distribution as the smoothing distribution; only MARS considers smoothing distribution alignment. Therefore, we additionally evaluate the certified radius under the smoothing distribution alignment setting and vertically compare the impact of different distributions on robustness certification. We sequentially used Gaussian, Laplacian, and Uniform distributions as smoothing distributions for l_2, l_1, and l_∞ guarantees calculation.

7.7.1 Setup. The Gaussian distribution $\mathcal{N}(\mu, \sigma)$ and the Laplacian distribution $\mathcal{L}(\mu, b)$ have the same mean 0 and standard deviation 1, lower bound a and upper bound b of the Uniform distribution $\mathcal{U}(a, b)$ are set to $-\sqrt{3}$ and $\sqrt{3}$ respectively. For the l_2 robustness guarantee, Gaussian is the default alignment distribution type in our setting, and the results have been shown in Section 7.1. Thus, we only show the tightness of l_1 and l_∞ robustness guarantees obtained through MARS under different smoothing distribution settings here.

7.7.2 Results. Evaluation results of l_1 MCR (see Figure 16) show that Gaussian and Laplacian distributions each excel in different classes, indicating that simply using a single distribution may miss a tighter certified radius, leading to a necessity of analyzing different distributions. For the l_1 robustness guarantee, the Gaussian distribution consistently used by the compared methods is not necessarily the best in various categories. Although the smoothing area of the uniform distribution is closer to l_∞, as it has upper and lower bounds, the choice of distribution parameters greatly limits the addition of noise. However, for the l_∞ certified radius, Gaussian distribution indeed has advantages. Although the l_∞-measured certified area best matches the Uniform distribution in shape, the smoothing region that the Uniform Distribution can cover is strictly controlled by the upper and lower bound parameters, and sharp truncation areas will appear. Therefore, the certification performance will be largely affected by the hyperparameters.

7.8 Certification Time Overhead

7.8.1 Setup. In this experiment, we present the evaluation of the *Average Certification Time* (ACT/seconds) per sample and mean certified radius (MCR) of network intrusion models (CADE and ACID) on different types of network traffic.

7.8.2 Results. The results are shown in Table 5 and Table 6. Compared to the SOTA method BARS, MARS achieves the highest MCR across all competitions, with an average increase in ACT per sample of 21.1 milliseconds (ms). Despite this slight increase, MARS maintains a reasonable certification time cost. For CADE, ACT per sample ranges from 8.0 to 14.6 ms, while for ACID, ACT per sample ranges from 24.4 to 36.1 ms.

Certification times for class sets were also evaluated. MARS-defend CADE takes approximately 25 seconds (sec) to certify 2K Infiltration or SSH-Bruteforce samples and around 125 sec for 13K

benign or 10K DoS-Hulk samples. For MARS-defend ACID, the certification times are about 300 sec for 13K benign/DDoS-HOIC or 11K DDoS-LOIC-HTTP samples, 200 sec for 6K Bot or 7K DoS-GoldenEye samples, and 100 sec for 3K FTP-Bruteforce or DoS-SlowHTTPTest samples.

The time cost gain of MARS comes mainly from the introduction of first-order gradient information to search the tight certified radius. This trade-off between slightly longer processing times and enhanced reliability is favorable in security-critical NIDS applications, especially since MARS maintains certification time in milliseconds for each sample. To improve certification efficiency while preserving accuracy, future work could leverage hardware acceleration, distributed computing, or other strategies to accelerate robustness certification using multi-order information.

Table 5. l_2-bounded MCR and ACT/seconds per sample of CADE.

MCR (ACT)	Benign	SSH-Bruteforce	Infiltration	DoS-Hulk
VRS	0.0000 (0.0008)	0.0049 (0.0015)	0.0000 (0.0017)	1.4082 (0.0012)
FRS	0.0000 (0.0089)	0.0054 (0.0125)	0.0000 (0.0122)	1.6316 (0.0134)
BARS	2.0494 (0.0004)	0.2450 (0.0011)	3.1658 (0.0013)	1.6372 (0.0007)
MARS	**2.0539 (0.0096)**	**0.4744 (0.0133)**	**3.1668 (0.0131)**	**1.6406 (0.0146)**

Table 6. l_2-bounded MCR and ACT/seconds per sample of ACID.

MCR (ACT)	Benign	FTP-Bruteforce	DDoS-HOIC	Bot	DoS-GoldenEye	DoS-SlowHTTPTest	DDoS-LOIC-HTTP
VRS	0.1950 (0.0028)	0.0000 (0.0045)	0.0000 (0.0028)	0.0000 (0.0040)	0.0001 (0.0039)	0.0000 (0.0053)	0.0000 (0.0033)
FRS	1.2077 (0.0238)	0.0000 (0.0323)	0.0000 (0.0235)	0.0000 (0.0565)	0.0024 (0.0309)	0.0014 (0.0459)	0.0000 (0.0269)
BARS	1.9036 (0.0029)	1.5814 (0.0046)	2.4801 (0.0029)	2.4456 (0.0041)	2.1928 (0.0040)	2.2280 (0.0051)	2.2034 (0.0033)
MARS	**1.9260 (0.0253)**	**1.6832 (0.0346)**	**2.4812 (0.0244)**	**2.6288 (0.0327)**	**2.2305 (0.0331)**	**2.2289 (0.0361)**	**2.2070 (0.0279)**

8 Discussion

8.1 Applications of the Dimensional Certified Radius

The certified radius reflects the robustness of DNNs in NIDS against potential input traffic variations, which is crucial for reliable network intrusion detection. A larger radius indicates stronger robustness, aiding prompt responses to complex intrusions, while a smaller radius highlights vulnerabilities, guiding targeted defenses. The dimensional certified radius pinpoints vulnerabilities in specific features, enabling focused defenses and reducing learning overhead on robust patterns.

8.2 Importance of the Norm Diversity

Calculating the certified radius under various norms (l_2, l_1, l_∞) provides a comprehensive evaluation of robustness against different attack vectors. The l_2 radius reflects robustness to overall feature variation, l_1 to sparse perturbations, and l_∞ to extreme outliers. In an environment with evolving cyber threats, multi-norm certification is valuable for preparing for dynamically changing attacks.

9 Conclusion

In this paper, we introduced MARS, a framework to certify the robustness of DNNs with heterogeneous input features, such as network traffic data. To strengthen robustness guarantees, we combine first-order function gradients with zero-order function output, iteratively adjusting the

certified region's center and expanding its radius while ensuring compliance with the certification hypothesis test. Experiments show that MARS achieves field-leading performance in certified robustness evaluated by certified radius and certified accuracy, as well as empirical robustness against evasion attacks and natural corruptions evaluated by robust accuracy, false alarm, etc. In future work, we plan to investigate non-l_p robustness certification against structural perturbations.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61960206014, Xidian University, and Purdue University.

References

[1] Julian Bitterwolf, Alexander Meinke, and Matthias Hein. 2020. Certifiably adversarially robust detection of out-of-distribution data. In *Advances in Neural Information Processing Systems (NeurIPS)*. 16085–16095.

[2] Jung-Woo Chang, Mojan Javaheripi, Seira Hidano, and Farinaz Koushanfar. 2023. RoVISQ: Reduction of Video Service Quality via Adversarial Attacks on Deep Learning-based Video Compression. In *Network and Distributed Systems Security (NDSS) Symposium*.

[3] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. 2018. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[4] Chih-Hong Cheng, Georg Nührenberg, and Harald Ruess. 2017. Maximum Resilience of Artificial Neural Networks. In *International Symposium on Automated Technology for Verification and Analysis (ATVA)*. 251–268.

[5] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning (ICML)*. 1310–1320.

[6] Alec F Diallo and Paul Patras. 2021. Adaptive clustering-based malicious traffic classification at the network edge. In *IEEE Conference on Computer Communications (INFOCOM)*. 1–10.

[7] Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George Pappas. 2019. Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*. 1–12.

[8] Marc Fischer, Maximilian Baader, and Martin Vechev. 2021. Scalable certified segmentation via randomized smoothing. In *International Conference on Machine Learning (ICML)*. 3340–3351.

[9] Aymeric Fromherz, Klas Leino, Matt Fredrikson, Bryan Parno, and Corina Pasareanu. 2020. Fast Geometric Projections for Local Robustness Certification. In *International Conference on Learning Representations (ICLR)*.

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*.

[11] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. 2018. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715* (2018).

[12] Sven Gowal, Krishnamurthy Dj Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. 2019. Scalable Verified Training for Provably Robust Image Classification. In *IEEE/CVF International Conference on Computer Vision (ICCV)*. 4842–4851.

[13] Dongqi Han, Zhiliang Wang, Ying Zhong, Wenqi Chen, Jiahai Yang, Shuqiang Lu, Xingang Shi, and Xia Yin. 2021. Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors. *IEEE Journal on Selected Areas in Communications* 39, 8 (2021), 2632–2647.

[14] Zhongkai Hao, Chengyang Ying, Yinpeng Dong, Hang Su, Jian Song, and Jun Zhu. 2022. Gsmooth: Certified robustness against semantic transformations via generalized randomized smoothing. In *International Conference on Machine Learning (ICML)*. 8465–8483.

[15] Mengdie Huang, Yingjun Lin, Xiaofeng Chen, and Elisa Bertino. 2024. MARS: Robustness Certification for Deep Network Intrusion Detectors via Multi-Order Adaptive Randomized Smoothing. In *IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. 1–8.

[16] Wei Jia, Zhaojun Lu, Haichun Zhang, Zhenglin Liu, Jie Wang, and Gang Qu. 2022. Fooling the eyes of autonomous vehicles: Robust physical adversarial examples against traffic sign recognition systems. In *Network and Distributed Systems Security (NDSS) Symposium*.

[17] Matt Jordan, Justin Lewis, and Alexandros G Dimakis. 2019. Provable certificates for adversarial examples: Fitting a ball in the union of polytopes. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[18] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. 2019. Certified robustness to adversarial examples with differential privacy. In *IEEE Symposium on Security and Privacy (S&P)*. 656–672.

[19] Guang-He Lee, Yang Yuan, Shiyu Chang, and Tommi Jaakkola. 2019. Tight certificates of adversarial robustness for randomly smoothed classifiers. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[20] Sungyoon Lee, Jaewook Lee, and Saerom Park. 2020. Lipschitz-certifiable training with a tight outer bound. In *Advances in Neural Information Processing Systems (NeurIPS)*. 16891–16902.

[21] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. TEXTBUGGER: Generating Adversarial Text Against Real-world Applications. In *Network and Distributed Systems Security (NDSS) Symposium*.

[22] Linyi Li, Tao Xie, and Bo Li. 2022. SoK: Certified Robustness for Deep Neural Networks. In *IEEE Symposium on Security and Privacy (S&P)*. 94–115.

[23] Shasha Li, Ajaya Neupane, Sujoy Paul, Chengyu Song, Srikanth V Krishnamurthy, Amit K Roy Chowdhury, and Ananthram Swami. 2019. Stealthy Adversarial Perturbations Against Real-Time Video Classification Systems. In *Network and Distributed Systems Security (NDSS) Symposium*.

[24] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. 2018. Defense against adversarial attacks using high-level representation guided denoiser. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[25] Cong Han Lim, Raquel Urtasun, and Ersin Yumer. 2020. Hierarchical verification for adversarial robustness. In *International Conference on Machine Learning (ICML)*. 6072–6082.

[26] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. 2018. Towards robust neural networks via random self-ensemble. In *European Conference on Computer Vision (ECCV)*. 369–385.

[27] Zhaoyang Lyu, Minghao Guo, Tong Wu, Guodong Xu, Kehuan Zhang, and Dahua Lin. 2021. Towards evaluating and training verifiably robust neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4308–4317.

[28] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations (ICLR)*.

[29] Jeet Mohapatra, Ching-Yun Ko, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. 2020. Higher-order certification for randomized smoothing. In *Advances in Neural Information Processing Systems (NeurIPS)*. 4501–4511.

[30] Han Cheol Moon, Shafiq Joty, Ruochen Zhao, Megh Thakkar, and Chi Xu. 2023. Randomized Smoothing with Masked Inference for Adversarially Robust Text Classificationsz. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

[31] Milad Nasr, Alireza Bahramali, and Amir Houmansadr. 2021. Defeating DNN-Based Traffic Analysis Systems in Real-Time With Blind Adversarial Perturbations.. In *USENIX Security Symposium (USENIX)*. 2705–2722.

[32] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. 2019. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning (ICML)*.

[33] Hai Phan, My T Thai, Han Hu, Ruoming Jin, Tong Sun, and Dejing Dou. 2020. Scalable differential privacy with certified robustness in adversarial learning. In *International Conference on Machine Learning (ICML)*. 7683–7694.

[34] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344* (2018).

[35] Aditi Raghunathan, Jacob Steinhardt, and Percy S Liang. 2018. Semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[36] Hadi Salman, Greg Yang, Huan Zhang, Cho-Jui Hsieh, and Pengchuan Zhang. 2019. A convex relaxation barrier to tight robustness verification of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[37] Lea Schonherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. 2019. Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding. In *Annual Network and Distributed System Security Symposium (NDSS)*.

[38] Zhouxing Shi, Yihan Wang, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. 2021. Fast certified robust training with short warmup. In *Advances in Neural Information Processing Systems (NeurIPS)*. 18335–18349.

[39] Vincent Tjeng, Kai Y Xiao, and Russ Tedrake. 2019. Evaluating Robustness of Neural Networks with Mixed Integer Programming. In *International Conference on Learning Representations (ICLR)*. 1–21.

[40] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. 2020. On adaptive attacks to adversarial example defenses. In *Advances in Neural Information Processing Systems (NeurIPS)*. 1633–1645.

[41] UNB. 2018. IPS/IDS dataset on AWS (CSE-CIC-IDS2018). https://www.unb.ca/cic/datasets/ids-2018.html.

[42] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* 17, 3 (2020), 261–272.

[43] Václav Voráček and Matthias Hein. 2022. Provably adversarially robust nearest prototype classifiers. In *International Conference on Machine Learning (ICML)*. 22361–22383.

[44] Binghui Wang, Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. 2021. Certified robustness of graph neural networks against adversarial structural perturbation. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*.

1645–1653.

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

[46] Ning Wang, Yimin Chen, Yang Hu, Wenjing Lou, and Y Thomas Hou. 2021. MANDA: On Adversarial Example Detection for Network Intrusion Detection System. In *IEEE Conference on Computer Communications (INFOCOM)*. 1–10.

[47] Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. 2021. Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. In *Advances in Neural Information Processing Systems (NeurIPS)*. 29909–29921.

[48] Eric Wong and Zico Kolter. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning (ICML)*. 5286–5295.

[49] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. 2019. Feature denoising for improving adversarial robustness. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 501–509.

[50] Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. 2020. Automatic perturbation analysis for scalable certified robustness and beyond. In *Advances in Neural Information Processing Systems (NeurIPS)*. 1129–1141.

[51] Kaidi Xu, Huan Zhang, Shiqi Wang, Yihan Wang, Suman Jana, Xue Lin, and Cho-Jui Hsieh. 2020. Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. *arXiv preprint arXiv:2011.13824* (2020).

[52] Limin Yang, Wenbo Guo, Qingying Hao, Arridhana Ciptadi, Ali Ahmadzadeh, Xinyu Xing, and Gang Wang. 2021. CADE: Detecting and explaining concept drift samples for security applications. In *USENIX Security Symposium (USENIX)*. 2327–2344.

[53] Yijun Yang, Ruiyuan Gao, Yu Li, Qiuxia Lai, and Qiang Xu. 2022. What you see is not what the network infers: Detecting adversarial examples based on semantic contradiction. In *Network and Distributed Systems Security (NDSS) Symposium*.

[54] Chaoyun Zhang, Xavier Costa-Perez, and Paul Patras. 2022. Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms. *IEEE/ACM Transactions on Networking* 30, 3 (2022), 1294–1311.

[55] Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. 2019. Towards Stable and Efficient Training of Verifiably Robust Neural Networks. In *International Conference on Learning Representations (ICLR)*.

[56] Huan Zhang, Shiqi Wang, Kaidi Xu, Linyi Li, Bo Li, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. 2022. General Cutting Planes for Bound-propagation-based Neural Network Verification. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*. 1656–1670.

[57] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. 2018. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Received 6 December 2024