MARS: Robustness Certification for Deep Network Intrusion Detectors via Multi-Order Adaptive Randomized Smoothing

Mengdie Huang
Xidian University, Purdue University
mdhuang1@stu.xidian.edu.cn

Yingjun Lin
Purdue University
lin1368@purdue.edu

Xiaofeng Chen ✉
Xidian University
xfchen@xidian.edu.cn

Elisa Bertino
Purdue University
bertino@purdue.edu

***Abstract*—Network intrusion detectors based on deep learning have high detection accuracy and the ability to adapt to evolving cyber threats. However, a serious drawback is their vulnerability to adversarial example attacks aimed at evading detectors and natural corruptions caused by random noise in the network environment. To provide robustness guarantees for deep neural networks against various perturbations, certified defenses against any possible perturbed inputs in the l_p-bounded region are gaining attention. mHowever, unlike existing approaches that focus on homogeneous image 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. Experiments on various deep learning-based network intrusion detector architectures show that MARS significantly improves the certification tightness (12.23% average increase in the l_2 certified radius), evasion attack detection accuracy (7.17% improvement on l_∞-PGD, 10.11% improvement on l_1-EAD), and natural corruption detection accuracy (16.65% enhancement on latency, 18.23% enhancement on packet loss) compared to BARS, the leading and only certified defense for network intrusion detectors.**

***Index Terms*—deep learning, certified robustness, adversarial attack, natural corruption, network intrusion detection.**

I. Introduction

Deep learning (DL)-based network intrusion detectors (NIDs) excel in detecting complex and evolving cyber threats by leveraging the capability of deep neural networks (DNNs) to analyze large-scale and diverse traffic data [1], [2]. However, previous work has shown that network traffic classifiers based on DNNs [3], [4] are as vulnerable to evasion attacks using adversarial examples as text [5], image [6], speech [7], and video classifiers [8]. An attacker can transform an otherwise correctly classified clean input into an adversarial example by subtly adding perturbations [9], resulting in the misclassification of these samples by the victim classifier. Adversarially modified malicious traffic usually mimics normal traffic patterns with constrained changes, making it close to clean samples in the representation space to evade detection [10].

Empirical defense methods designed to enhance the robustness of DNNs, such as adversarial training [9], [11], feature denoising [12], [13], and model ensembling [14], [15], have been fully verified in the DL field, and their applicability has also been explored for network intrusion detection [16]. However, a common problem in various domains is that the robustness achieved by these heuristic strategy-driven empirical defenses is likely to be bypassed by new evasion attack approaches [17], [18]. Such a shortcoming allows attackers to evade the empirically robust model through adaptive attacks [19], leading to an endless arms race of evasion attacks and defenses. Thus, it is hard to establish a high-level trust in the output of the model based on empirical defenses in security-sensitive applications.

Recognizing these shortcomings, research on the robustness of DL models has gradually shifted to certified defense [20]. Such a defense aims to calculate 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 radius. The radius is provided as a robustness guarantee along with the input's predictions. For the same model and input, a larger 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 [20], [21]. However, a notable drawback of incomplete certification is that the robustness guarantee provided is loose, that is, the calculated lower bound is far from 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, including deterministic certification such as interval bound propagation [22]–[24], relaxation [25]–[28], neuron branching and bounding [29]–[32], and probabilistic certification such as differential privacy [33], [34] and randomized smoothing [17], [21], [35]–[37]. Given that an ideal certified defense against evasion attacks should be model agnostic, that is, it should apply to various types of DL models without modifying or being limited by the specific internal structures of these models, randomized smoothing-based approaches have been proven to be the most competitive in terms of tighter and architecturally scalable certification in the image [20], text [38], and graph [39] fields.

Motivations. However, certified defense efforts for network intrusion detection have been minimal. The main challenges arise from the heterogeneity of network traffic features, where different dimensions carry varying semantics and charac-

teristics. 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. Additionally, the diversity of NID architectures also introduces difficulties to certified defenses. In binary/multi-class classification or known/unknown anomaly detection tasks, the detection principles employed lead to the utilization of various DNN models, such as CADE (Contrastive Autoencoder for Drifting detection and Explanation) [2], ACID (Adaptive Clustering-based Intrusion Detection) [1], etc. This imposes strict demands on the scalability of certification methods across diverse model architectures. Until now, only one approach, BARS (Boundary-Adaptive Randomized Smoothing) [18], has been proposed to certify the robustness of network traffic classifiers. By utilizing the zero-order information-based randomized smoothing, it obtained larger l_2 certified radii than neuron branching and bounding methods. Unfortunately, its l_2 robustness guarantee is proven relatively loose and it lacks certification for other l_p norms-bounded robustness guarantees. Providing multiple l_p-measured certified radii can help in deeply analyzing model vulnerabilities and boosting general robustness against 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.

Our Method. To address the above shortcomings, we propose MARS — a novel framework that certifies the robustness of DL-based NIDs using **M**ulti-order **A**daptive **R**andomized **S**moothing, as shown in Fig. 1. Capabilities of MARS are demonstrated in three main aspects. ① 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. ② Then, to tighten the robustness guarantee, we adopt a two-step strategy: (i) 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. (ii) We iteratively move 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, and use binary search to estimate the upper and lower bounds of the interval of certified radius, so that the certified radius can be further improved. ③ Finally, to provide diverse l_p certificates, we construct a smoothing distribution candidate set consisting of Gaussian, Laplacian, and Uniform distributions, and implement specific parameter optimization and first-order gradient estimation for each distribution.

We evaluate the performance of MARS on two advanced NIDs, CADE [2] and ACID [1], with three datasets created from CSE-CIC-IDS-2018 [40], and compare MARS with the state-of-the-art (SOTA) traffic-specific certification BARS [18], image-specific certification Vanilla Randomized Smoothing (VRS) [17], and First Order-based Randomized Smoothing (FRS) [35] in terms of the certified radius, certified accuracy, robust accuracy, clean accuracy, and time overhead.

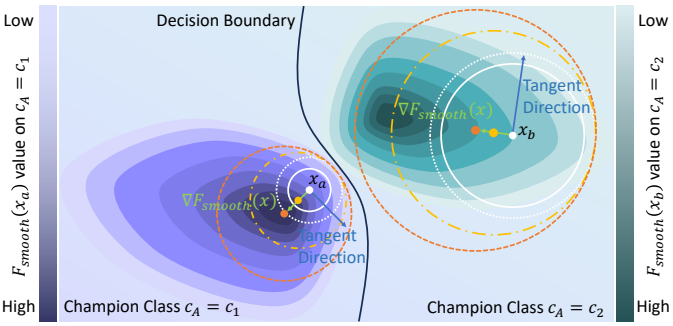

Fig. 1: Certified Radius Obtained through MARS. c_A denotes the class with the highest confidence returned by the smoothed classifier F_{smooth} on the input. x_a and x_b are two inputs predicted as different champion classes, c_1 and c_2. The solid white line outlines the certified region relying solely 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. See §III-B3 for a detailed explanation.

In summary, we make the following contributions:

- We propose MARS — a robustness certification framework, to certify the robust radius of DL-based 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 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 NID robustness certification. 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 detection model.

II. Threat Model

We focus on two robustness threats: (i) evasion attacks — deliberately launched by attackers using adversarial examples, and (ii) natural corruptions — unintentionally caused distribution shift by random noise in the network environment.

1) Evasion Attacks: We first focus on white-box evasion attacks. The adversary creates strong l_p 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 aim to simulate a most powerful threat in the adversarial scenario where the adversary has maximum visibility into the model internals. This enables evaluating the robustness of the model against sophisticated attacks leveraging the model's inner workings, while also revealing potential vulnerabilities of the target model. Although evasion attacks can target clean samples belonging to any category, in network intrusion detection, 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*.

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.*

III. Design of MARS

This section introduces the design of the proposed certified defense method 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.

A. 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 weight 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.

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 $\bar{s} = (\bar{s_1}, ..., \bar{s_d})$.

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 $\tilde{s} = (\tilde{s_1}, ..., \tilde{s_d}) = \left({}^{e^{\bar{s_1}}}/_{\sum_{i=1}^{d} e^{\bar{s_i}}}, ..., {}^{e^{\bar{s_d}}}/_{\sum_{i=1}^{d} e^{\bar{s_i}}}\right)$ whose components sum to 1. Then the dimensional robust radius contribution weight w_i is calculated according to (1).

$$R_i = w_i \times R, w_i = {}^{R_i}/_{R} = {}^{1}/_{d\tilde{s_i}} = {}^{1}/_{d\tilde{s_i}}, \tag{1}$$

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.

B. Multi-Order Robustness Certification

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 of the smoothed classifier F_{smooth} w.r.t x and expand the certified robust region along the direction in which the confidence score increases.

1) Architecture of the Smoothed NID: As shown in Fig. 2, the main difference between the smoothed NID and the base NID is that the predicted label of F_{smooth} on 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 $x + \eta$. MARS includes 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-measured certified radius R. First, n_{large} (defaults to $10,000$) noises are randomly sampled from the smoothing distribution $\mathcal{D}$ and 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}

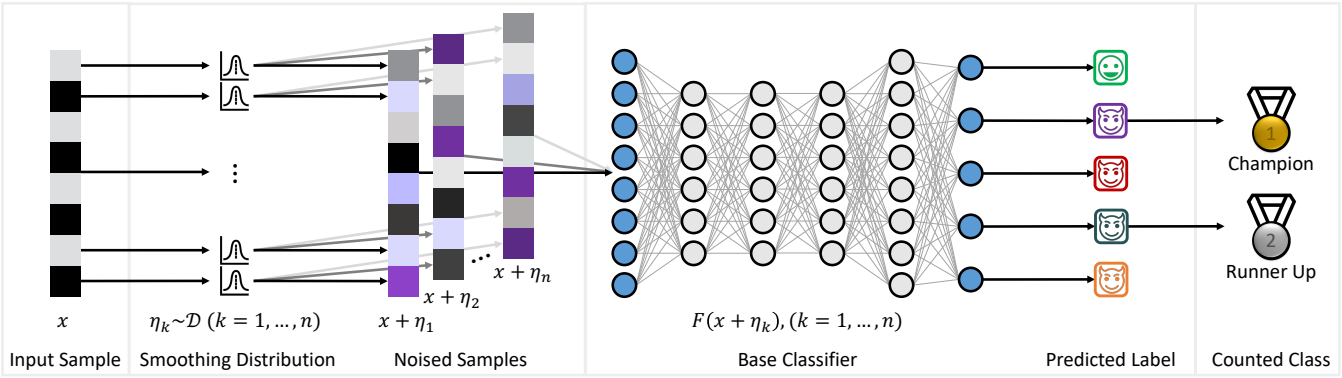

Fig. 2: Architecture of the Smoothed NID. ① Prediction Procedure: $n = n_{small}$, identify the champion class c_A that is predicted most times among n noised samples. ② Certification Procedure: $n = n_{large}$, count the number of noised samples predicted as c_A to estimate $P_A = \mathbb{P}(F_{smooth}(x) = c_A)$ and $P_B = \mathbb{P}(F_{smooth}(x) = c_B)$.

and n_A, the certified radius is then calculated using the zero-order output and first-order gradient of the smoothed classifier.

2) Smoothing Distribution Parameters Optimization: The difference between a smoothed NID 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. Parameters ϑ optimized dimensionally for the multivariate distribution facilitate an adaptive approach to the classification boundary in feature space. 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 optimizing (2).

$$\min_{\vartheta} \mathbb{E}_{x \sim D_{train}} \mathbb{I}[F(x + \eta_\vartheta) \neq F(x)] L(f(x + \eta_\vartheta), F(x)) - \mathbb{I}[F(x + \eta_\vartheta) = F(x)] L(f(x + \eta_\vartheta), F(x)), \tag{2}$$

where $\vartheta = \vartheta_{shape} \times 1$ and $\mathbb{I}$ is the indicator function.

Distribution Scale Optimization. The optimization goal of the distribution scale, as defined in (3), 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.

$$\max_{\vartheta} R = \frac{\sigma}{2}(\Phi^{-1}(\underline{P_A}) - \Phi^{-1}(\overline{P_B})) = \max_{\vartheta} \frac{\vartheta}{2}(\Phi^{-1}(\mathbb{P}_{\eta \sim \mathcal{D}_{std}} (F(x + \vartheta\eta) = c_A)) - \Phi^{-1}(\mathbb{P}_{\eta \sim \mathcal{D}_{std}}(F(x + \vartheta\eta) = c_B))) \tag{3}$$

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.

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 most probable 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 [17], using $\texttt{LowerConfidenceBound}(n_{large}, n_A, \alpha)$, 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 $\overline{P_B} = 1 - \underline{P_A}$. Like [17], the certified radius R_{zero} based only on the zero-order information is calculated according to (4):

$$R_{zero} = \frac{\sigma}{2}(\Phi^{-1}(\underline{P_A}) - \Phi^{-1}(\overline{P_B})) \tag{4}$$

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 surrounding the input sample x_a or input sample x_b in Fig. 1.

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. We can see from Fig. 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. Take a l_2 robust radius as an example. Refer to [35], we obtain the final certified radius R by solving the system of simultaneous equations shown in (5). 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 (5), where z_1 and z_2 are fixed. The search stops when $R_{high} - R_{low} \leq 0$.

$$\begin{aligned} \Phi(z_1 - R) - \Phi(z_2 - R) &= 0.5 \\ \Phi(z_1) - \Phi(z_2) &\leq F_{smooth}(x) = P_A \\ \varphi(z_2) - \varphi(z_1) &\geq \sigma ||\nabla F^{c_A}_{smooth}(x)||_2 \end{aligned} \tag{5}$$

C. Smoothing Distribution Alignment

To provide guarantee calculations for robust regions under various types of l_p norm measures, selecting the appropriate distribution whose sampling region aligns with the l_p-bounded robust region is essential. We explore the question of which probability distribution yields the most random noises into the corresponding l_p-measured certificate region. In the l_2-bounded robust region, noised samples from a Gaussian distribution form areas resembling a circle in 2D feature space, aligning with the l_2 norm. In the l_1-bounded region, samples from a Laplacian distribution form diamond-shaped areas, indicating higher central probability density and consistent with the l_1 norm. In the l_∞-bounded region, samples from a Uniform distribution take on shapes akin to a square, catering to extreme variations and matching the l_∞ norm.

IV. Experimental Evaluation

This section introduces evaluation metrics (§IV-A), experimental setup (§IV-B), comparison of l_2 robustness guarantees tightness with existing schemes (§IV-C), comparison of l_1 and l_p robustness guarantees tightness (§IV-D), impact analysis of distribution alignment (§IV-D), comparison of defense against evasion attacks (§IV-E) and natural corruptions (§IV-F), The code for MARS has been open-sourced at https://github.com/CertNID/MARS.

A. Evaluation Metrics

1) Certified Robustness: Two metrics are adopted to evaluate the certified robustness.

Mean Certified Radius (MCR). It calculates the average certified radii of test samples as $MCR = \sum_{i=1}^{N} R_i/N$, where N is the number of test samples. We count MCR based on all samples in the same class to observe the certified robustness of the detection model across different categories. A larger MCR indicates a tighter lower bound for the robust radius.

Certified Accuracy. Given a radius threshold R_{given}, it calculates the ratio of test samples correctly predicted by F_{smooth} with a certified radius R greater than R_{given} according to (6). Higher certified accuracy indicates more samples passing robustness certification under the threshold R_{given}.

$$CertifiedAcc = \frac{N_{(F_{smooth}(x)=y_{true})\&(R \geq R_{given})}}{N_{TotalCertifyTest}} \tag{6}$$

2) Empirical Robustness: **Robust Accuracy**, the ratio of correctly predicted samples among all test perturbated samples, as shown in (7), is used to evaluate the empirical robustness of NIDs on four evasion attacks (l_2-PGD, l_∞-PGD, l_1-PGD, l_1-EAD) and two natural corruptions (Packet Loss and Latency). For evasions, Robust Accuracy equals Recall (True Positive Rate), as the evasion test set contains only adversarial malicious traffic. For corruptions, we also measure False Positive (Alarm) Rate, False Negative Rate, F1-score, and Precision.

$$RobustAcc = \frac{N_{(F_{smooth}(x^*)=y_{true})}}{N_{TotalPerTest}} \tag{7}$$

3) Regular Performance: **Clean Accuracy** shown in (8), the ratio of correctly predicted samples among all clean test samples, evaluates the regular performance. We also measure the False Alarms, Recall, F1-score, etc.

$$CleanAcc = \frac{N_{(F_{smooth}(x)=y_{true})}}{N_{TotalCleanTest}} \tag{8}$$

B. Experimental Setup

1) Testbed: We implemented the method using PyTorch 2.0.1 and SciPy V1.11.2 [41]. Each experiment ran three times with varied random seeds (42, 43, 44) on an NVIDIA GeForce 3090 GPU with CUDA V11.7, and the averages were shown.

2) Classifiers: We evaluated two SOTA NIDs. CADE [2] is a concept drift NID capable of training on $n-1$ classes and inferring on n classes to detect unknown anomalies. ACID [1] is a NID that integrates unsupervised and supervised learning for multi-classification.

3) Datasets: Following and extending the BARS [18] settings, we evaluated the performance of MARS using two subdatasets, CSE-CIC-IDS-2018-CADE and CSE-CIC-IDS-2018-ACID, derived from the CSE-CIC-IDS-2018 [40].

CSE-CIC-IDS-2018-CADE: For CADE, we used samples from four classes in the CSE-CIC-IDS-2018 dataset: Benign, SSH-Bruteforce, DoS-Hulk, and Infiltration. Since CADE supports the detection of concept drift, the test set must include categories that the model has seen and unseen during the training phase. We divide CSE-CIC-IDS-2018-CADE into two datasets: DoS-Hulk-Drift and Infiltration-Drift. In DoS-Hulk-Drift, DoS-Hulk appears only in the test set. In Infiltration-Drift, Infiltration appears exclusively in the test set.

CSE-CIC-IDS-2018-ACID: For ACID, we used samples from the Benign class and three other malicious categories in the CSE-CIC-IDS-2018 dataset, including FTP-Bruteforce, DDoS-HOIC, and Bot.

4) Baseline Methods: To ensure fair comparison, we selected three SOTA robustness certification methods known for their good architecture-level scalability. These methods rely on randomized smoothing, allowing for applicability across various NID structures.

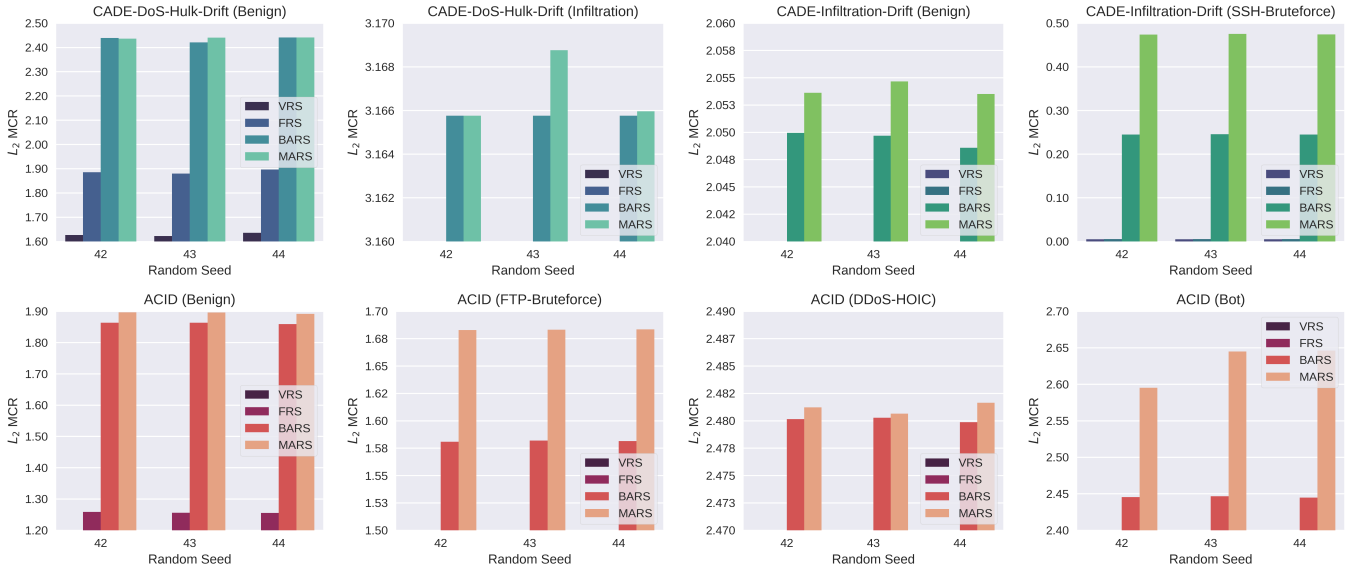

Fig. 3: Class-specific l_2-measured Mean Certified Radius (MCR).

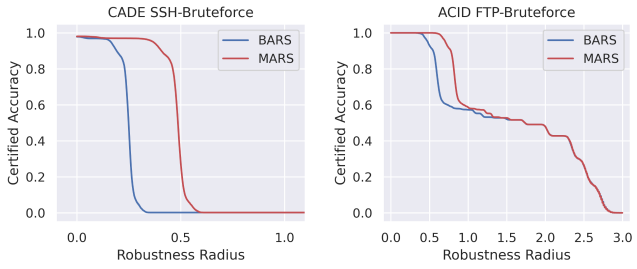

Fig. 4: Certified Accuracy of l_2 Robustness Guarantee of Various NIDs with Different Certification Methods.

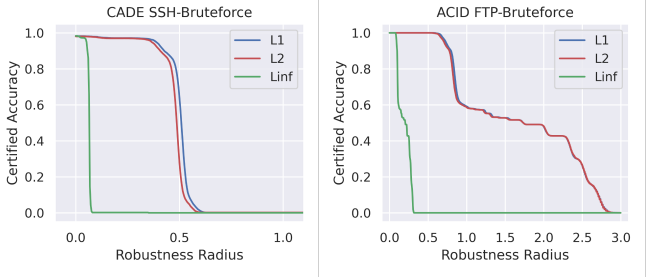

Fig. 5: Certified Accuracy of l_p Robustness Guarantees of Various NIDs under the Same Smoothing Distribution.

C. Comparing l_2 Robustness Guarantee with SOTA Methods

We first compare the tightness of the l_2 robustness guarantees provided by MARS with VRS [17] and FRS [35] for the image domain, and BARS [18] for the network traffic domain. We calculated the MCR and certified accuracy (defined in §IV-A) for each category on ACID and CADE.

Setup. To be comparable with VRS and FRS that 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 failure probability α for radius calculation is set to 0.001. To be consistent with the evaluation setup in BARS, MCR (see Fig. 3) and certified accuracy (see Fig. 4) are measured by category.

Results. The results show that MARS always outperforms the SOTA method. Especially on the CADE-Infiltration-Drift dataset and the CSE-CIC-IDS-2018-ACID dataset, MARS shows 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 zero, 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.

D. Comparing Various l_p Robustness Guarantees

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 [35], since neither VRS nor BARS supports l_1 and l_∞ certification. Although FRS incorporates the l_2, l_1, and l_∞ guarantees, it relies exclusively on the Gaussian distribution for smoothing. For fair comparison, we compare MARS with FRS specifically under Gaussian smoothing distribution.

Results. Fig. 6 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, Fig. 5 shows 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.

E. Evaluating Robustness against Different l_p Evasion Attacks

We employ Projected Gradient Descent (PGD) [9] and Elastic-Net Attack to DNN (EAD) [42] as white-box threat models to generate adversarial examples $x^* = x + \delta$.

Setup. Following the BARS settings, we selected one of the malicious categories, Bot, as a representative to test whether

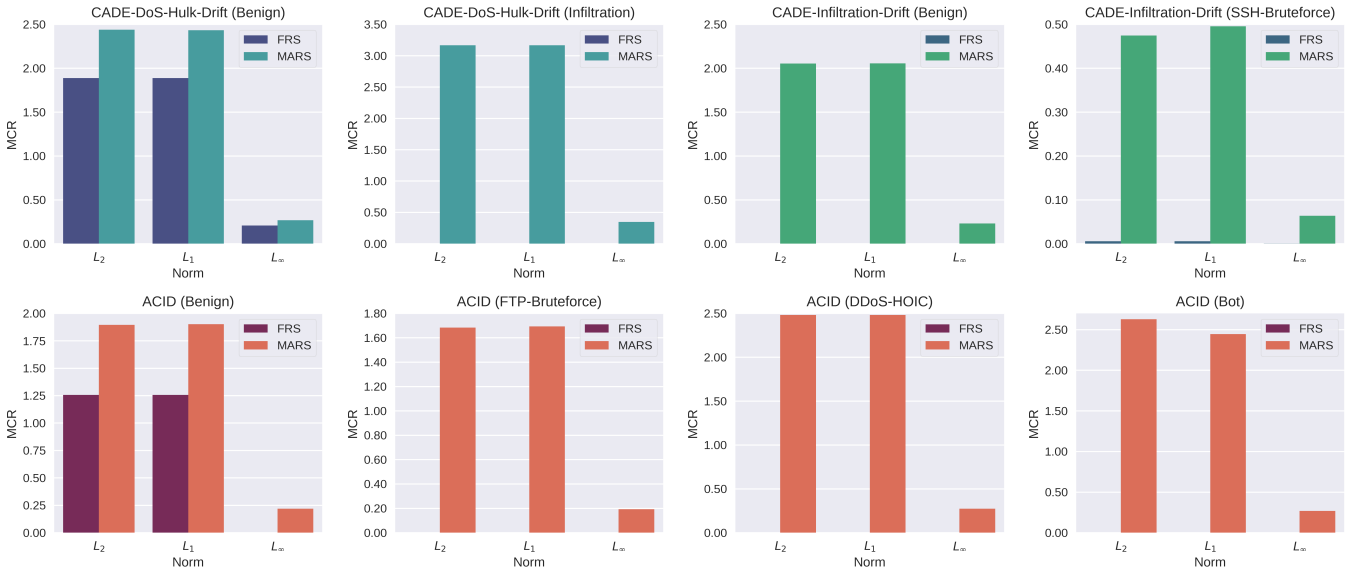

Fig. 6: l_p-measured Mean Certified Radius (MCR) under the Same Gaussian Distribution.

TABLE I: Robustness of ACID against Evasion Attacks

Method	CleanAcc/Recall on Clean Bot (%)	RobustAcc/Recall on Adversarial Bot (%)		
		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 [18]	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**

the ACID model with certified defense can correctly identify adversarial Bot samples, and calculated the robust accuracy and clean accuracy. 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_{step} that determines the maximum allowed perturbation at each iteration is set to 0.75. For l_∞-PGD, ϵ is 0.2 and ϵ_{step} is 0.1. The maximum number of iterations is set to 20 for all attacks. ***Results.*** Since we only measure the detection accuracy of the model against adversarial malicious samples, robust accuracy here is equivalent to the Recall rate. Compared to BARS, our defense boosts the detection accuracy against evasion attacks by 1.70% for l_2-PGD, 7.17% for l_∞-PGD, and 10.11% for l_1-EAD (see Table I). Compared to the base NID 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.

F. Evaluating Robustness against Varied Natural Corruptions

Natural corruption 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 Packet Loss, as defined in §II-2.

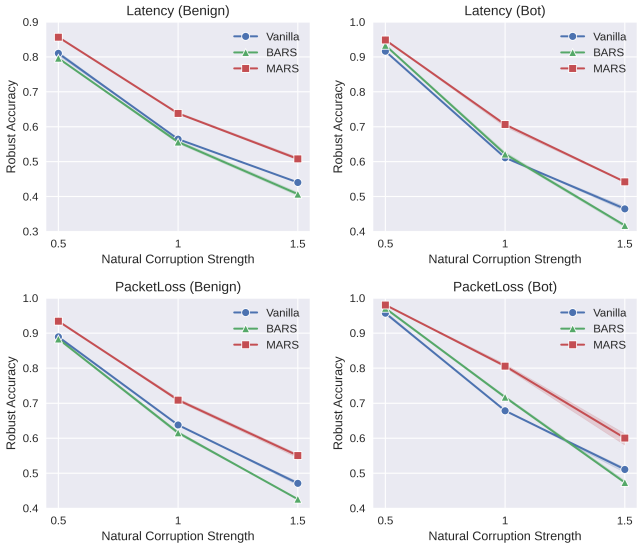

Fig. 7: Robustness of ACID against Natural Corruptions.

Setup. 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 in $\{0.5, 1.0, 1.5\}$ to mimic the different corruption strengths. ***Results.*** As shown in Fig. 7, MARS outperforms BARS and Vanilla across various corruption intensities and classes. Under the same corruption strength, both vanilla and certified defended models show higher resilience to Packet Loss than to Latency, suggesting that ACID is more sensitive in time-related features and more robust in quantity-related features.

V. Conclusion

We introduced MARS to certify the robust radius of DL-based NIDs. Compared to BARS, our use of first-order gradient tightens bounds (12.23% radius increase) and improves

certified accuracy. Also, MARS boosts detection accuracy on evasions and corruptions (7.17%, 10.11%, 16.65%, 18.23% higher for l_∞-PGD, l_1-EAD, Latency, and Packet Loss, respectively). Future work will explore automatic detection of attack distribution and non-l_p guarantees for structural perturbations.

VI. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61960206014 and in part by the 111 Center under Grant B16037. Mengdie Huang's work was done when she visited Purdue University.

References

[1] A. F. Diallo and P. Patras, "Adaptive clustering-based malicious traffic classification at the network edge," in *IEEE Conference on Computer Communications*, 2021.

[2] L. Yang, W. Guo, Q. Hao, A. Ciptadi, A. Ahmadzadeh, X. Xing, and G. Wang, "CADE: Detecting and explaining concept drift samples for security applications," in *USENIX Security Symposium*, 2021.

[3] D. Han, Z. Wang, Y. Zhong, W. Chen, J. Yang, S. Lu, X. Shi, and X. Yin, "Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors," *IEEE Journal on Selected Areas in Communications*, 2021.

[4] N. Wang, Y. Chen, Y. Hu, W. Lou, and Y. T. Hou, "Manda: On adversarial example detection for network intrusion detection system," in *IEEE Conference on Computer Communications*, 2021.

[5] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "Textbugger: Generating adversarial text against real-world applications," 2019.

[6] W. Jia, Z. Lu, H. Zhang, Z. Liu, J. Wang, and G. Qu, "Fooling the eyes of autonomous vehicles: Robust physical adversarial examples against traffic sign recognition systems," 2022.

[7] L. Schonherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," in *Network and Distributed System Security Symposium*, 2019.

[8] J.-W. Chang, M. Javaheripi, S. Hidano, and F. Koushanfar, "Rovisq: Reduction of video service quality via adversarial attacks on deep learning-based video compression," 2023.

[9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[10] M. Nasr, A. Bahramali, and A. Houmansadr, "Defeating dnn-based traffic analysis systems in real-time with blind adversarial perturbations.," in *USENIX Security Symposium*, 2021.

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[12] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[13] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[14] X. Liu, M. Cheng, H. Zhang, and C.-J. Hsieh, "Towards robust neural networks via random self-ensemble," in *European Conference on Computer Vision*, 2018.

[15] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu, "Improving adversarial robustness via promoting ensemble diversity," in *International Conference on Machine Learning*, 2019.

[16] C. Zhang, X. Costa-Perez, and P. Patras, "Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms," *IEEE/ACM Transactions on Networking*, 2022.

[17] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *International Conference on Machine Learning*, 2019.

[18] K. Wang, Z. Wang, D. Han, W. Chen, J. Yang, X. Shi, and X. Yin, "Bars: Local robustness certification for deep learning based traffic analysis systems.," in *Network and Distributed Systems Security Symposium*, 2023.

[19] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," in *Advances in Neural Information Processing Systems*, 2020.

[20] L. Li, T. Xie, and B. Li, "Sok: Certified robustness for deep neural networks," in *IEEE Symposium on Security and Privacy*, 2022.

[21] G.-H. Lee, Y. Yuan, S. Chang, and T. Jaakkola, "Tight certificates of adversarial robustness for randomly smoothed classifiers," in *Advances in Neural Information Processing Systems*, 2019.

[22] S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, and P. Kohli, "On the effectiveness of interval bound propagation for training verifiably robust models," *arXiv:1810.12715*, 2018.

[23] J. Bitterwolf, A. Meinke, and M. Hein, "Certifiably adversarially robust detection of out-of-distribution data," in *Advances in Neural Information Processing Systems*, 2020.

[24] Z. Shi, Y. Wang, H. Zhang, J. Yi, and C.-J. Hsieh, "Fast certified robust training with short warmup," in *Advances in Neural Information Processing Systems*, 2021.

[25] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," *arXiv:1801.09344*, 2018.

[26] H. Salman, G. Yang, H. Zhang, C.-J. Hsieh, and P. Zhang, "A convex relaxation barrier to tight robustness verification of neural networks," in *Advances in Neural Information Processing Systems*, 2019.

[27] A. Raghunathan, J. Steinhardt, and P. S. Liang, "Semidefinite relaxations for certifying robustness to adversarial examples," in *Advances in Neural Information Processing Systems*, 2018.

[28] K. Xu, H. Zhang, S. Wang, Y. Wang, S. Jana, X. Lin, and C.-J. Hsieh, "Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers," *arXiv*, 2020.

[29] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," in *Advances in Neural Information Processing Systems*, 2018.

[30] K. Xu, Z. Shi, H. Zhang, Y. Wang, K.-W. Chang, M. Huang, B. Kailkhura, X. Lin, and C.-J. Hsieh, "Automatic perturbation analysis for scalable certified robustness and beyond," in *Advances in Neural Information Processing Systems*, 2020.

[31] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C.-J. Hsieh, and J. Z. Kolter, "Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification," in *Advances in Neural Information Processing Systems*, 2021.

[32] Z. Lyu, M. Guo, T. Wu, G. Xu, K. Zhang, and D. Lin, "Towards evaluating and training verifiably robust neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[33] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," in *IEEE Symposium on Security and Privacy*, 2019.

[34] H. Phan, M. T. Thai, H. Hu, R. Jin, T. Sun, and D. Dou, "Scalable differential privacy with certified robustness in adversarial learning," in *International Conference on Machine Learning*, 2020.

[35] J. Mohapatra, C.-Y. Ko, T.-W. Weng, P.-Y. Chen, S. Liu, and L. Daniel, "Higher-order certification for randomized smoothing," in *Advances in Neural Information Processing Systems*, 2020.

[36] M. Fischer, M. Baader, and M. Vechev, "Scalable certified segmentation via randomized smoothing," in *International Conference on Machine Learning*, 2021.

[37] Z. Hao, C. Ying, Y. Dong, H. Su, J. Song, and J. Zhu, "Gsmooth: Certified robustness against semantic transformations via generalized randomized smoothing," in *International Conference on Machine Learning*, 2022.

[38] H. C. Moon, S. Joty, R. Zhao, M. Thakkar, and C. Xu, "Randomized smoothing with masked inference for adversarially robust text classificationsz," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

[39] B. Wang, J. Jia, X. Cao, and N. Z. Gong, "Certified robustness of graph neural networks against adversarial structural perturbation," in *ACM Conference on Knowledge Discovery & Data Mining*, 2021.

[40] UNB, "Ips/ids dataset on aws (cse-cic-ids2018)," 2018.

[41] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, *et al.*, "Scipy 1.0: fundamental algorithms for scientific computing in python," *Nature Methods*, 2020.

[42] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "Ead: elastic-net attacks to deep neural networks via adversarial examples," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.