Adaptive Randomized Smoothing with Certified Robustness for Mitigating Tabular Adversarial Attacks

Nour Alhussien[1 [0009−0001−4803−2744]], Bradley Boswell[2 [0009−0001−1506−2533]], Gagan Agrawal[[0000−0002−2609−1428]], Ahmed Aleroud[2 [0000−0002−2609−1428]], and Gokila Dorai[2 [0000−0001−5825−7034]]

[1] University at Albany, Albany, NY 12222, USA
nalhussien@albany.edu
[2] Augusta University, Augusta, GA 30912, USA
{brboswell,aaleroud,gdorai}@augusta.edu
[3] University of Georgia, Athens, GA 30602, USA
gagrawal@uga.edu

Abstract. Deep learning models for tabular classification remain susceptible to adversarial perturbations—small, structured changes to input features that can induce incorrect predictions. While various defenses have been proposed, randomized smoothing has emerged as a certifiable and theoretically grounded technique for building robust classifiers. However, standard randomized smoothing can increase false positives and degrade accuracy on clean, non-perturbed inputs, limiting practical utility in real-world applications. To address this limitation, we introduce a novel methodology that integrates entropy-scaled smoothing during training and margin-confidence proxy-based resampling during certification. This enhanced framework produces a certifiably robust classifier that retains resilience to adversarial perturbations while substantially lowering false positive rates. We empirically evaluate our adaptive randomized smoothing method on eight attacks, two classifiers (a deep neural network and TabNet) and across three benchmark datasets. We validate the effectiveness of our approach in both binary and multi-class classification settings, demonstrating its applicability across different task complexities. In addition to standard adversarial settings, we rigorously test our defense against constrained, realistic attack scenarios to ensure practical robustness. Our results demonstrate improved resilience and reduced false positives, while maintaining reasonable operational efficiency with a controllable efficiency-accuracy tradeoff; fewer samples accelerate certification but may slightly reduce accuracy.

Keywords: Randomized smoothing · Model robustness · Adversarial attacks · Tabular learning

1 Introduction

Tabular data is central to many real-world decision systems arising across diverse domains such as finance, healthcare, e-commerce, and others. This has motivated

the use of Machine Learning (ML) and Deep Learning (DL) models that can learn complex feature interactions beyond traditional rule-based methods. Recently, specialized DL architectures for tabular learning have demonstrated strong performance while retaining interpretability and practical deployment appeal [6]. One security-critical instance of tabular classification is the Network Intrusion Detection System (NIDS), where ML/DL models trained on network-flow features have been widely studied and demonstrated advantages over signature-based approaches in detecting complex attack patterns [23, 34, 8, 20].

Despite their empirical success, ML/DL models are vulnerable to adversarial attacks [35, 18, 24]. In tabular settings, adversarial manipulation often corresponds to small, carefully crafted changes to a subset of input features that can flip a model's prediction while remaining feasible under *domain constraints* (e.g., value ranges and feature dependencies) [10, 14, 3]. This concern is also reflected in the network security literature, where ML/DL-based intrusion detectors have been shown to be vulnerable to adversarial perturbations in network-flow records [40, 2]. Such vulnerabilities are particularly problematic in high-stakes deployments, where even subtle feature manipulations can lead to downstream harm.

A variety of defenses have been proposed to mitigate adversarial attacks [38, 43, 15, 30, 1, 4]. Key considerations include computational overhead, accuracy on clean non-perturbed inputs, perturbation assumptions or cross-domain applicability, and whether the defense provides formal robustness guarantees. Considering these aspects, randomized smoothing has emerged as a scalable method that transforms any base classifier into a smoothed classifier with certified robustness under the L_2 norm [15]. It works by injecting Gaussian noise into the inputs and predicting via a majority vote across multiple noisy instances. Recent certified defenses such as BARS and MARS extend randomized smoothing by redesigning the noise distribution through learned feature-space transformations and NIDS-focused mechanisms [39, 21]. Yet, even with these advances, a central challenge remains for general tabular learning methods; how to allocate noise and sampling based on prediction certainty without sacrificing clean accuracy.

In particular, applying standard randomized smoothing to tabular data raises challenges because tabular inputs are often mixed-type, structured, and constraint-driven; indiscriminate noise injection can distort sample realism and reduce clean accuracy. In operational tabular deployments, this clean accuracy degradation can result in increased Type 1 errors, increasing the burden of downstream reviews and reducing trust in the model [7, 16]. A key driver of these issues is that standard randomized smoothing uses a fixed noise level and a constant number of Monte Carlo samples for all inputs [15], regardless of model confidence or sample complexity.

To address these limitations, we propose a framework with two key innovations. First, we adaptively modulate the level of noise based on model uncertainty, quantified using entropy: we apply less noise to high-entropy (uncertain) samples and more noise to low-entropy (confident) samples. This helps preserve clean-input accuracy while enhancing robustness where it is needed. Second, we introduce a margin-confidence proxy metric—derived from the top-2

softmax probabilities; when the model's prediction confidence is low (i.e., the margin between the top two classes is small), our defense dynamically increases the number of sampled predictions to improve decision reliability. This adaptive sampling reduces unnecessary computation for confident predictions while maintaining robustness where it matters most. Overall, our method improves adversarial robustness, retains high accuracy on clean data, and achieves computational efficiency by allocating resources where they have the most impact. While intrusion detection serves as a motivating example in this work, our focus is on improving the robustness of tabular classifiers, across both binary and multi-class settings, independent of any specific classification domain.

Our primary contributions include:

- A novel Adaptive Randomized Smoothing (ARS) algorithm that leverages entropy-based adaptive noise and margin–confidence proxy certification to improve model robustness against a range of adversarial attacks.
- A theoretical justification showing that our adaptive sampling strategy is a variance-targeted allocation mechanism, preserving the statistical validity of certification while improving robustness guarantees.
- A comprehensive evaluation of our adaptive randomized smoothing approach against eight state-of-the-art adversarial attacks under two distinct threat models (six white-box and two black-box), tested on both a conventional Deep Neural Network (DNN) and an attention-based architecture (TabNet).

Overall, adaptive smoothing improves certified accuracy over the original randomized smoothing baseline in most scenarios without sacrificing clean-data accuracy, incurs only moderate dataset-dependent overhead, and offers a promising path toward accurate, provably robust tabular ML/DL systems against diverse adversarial attacks. The remainder of this paper is structured as follows: Section 2 reviews the related work in this domain. Section 3 outlines the attacker threat model used to evaluate our defense. Section 4 presents our novel defense framework. Section 5 discusses the details of our experiment design. Section 6 reports the results and analysis. Lastly, we state our conclusions and discuss future work in Section 7. To ensure the reproducibility of our results, we have made our implementation publicly available.[4]

2 Related Work

We review the commonly used defense techniques for improving the robustness of ML/DL classifiers on tabular data against adversarial attacks. Additionally, we highlight representative results from the NIDS literature as a security-critical tabular application where adversarial robustness and false positives are particularly consequential.

Adversarial training is a widely used defense mechanism that has been extensively studied, often specific to NIDS settings [30, 37, 1]. The approach was

[4] https://github.com/Nour-Alhussien/AdaptiveRandomizedSmoothing.git

formalized in [24] as a min-max optimization problem, where strong adversarial examples are generated via projected gradient descent and incorporated during training. Similarly, [32] combines adversarial training with Gaussian data augmentation and high-confidence prediction mechanisms to defend against real-time adversarial manipulation of packet-flow features. While adversarial training can improve robustness, it typically increases computational cost and can reduce accuracy on clean inputs; moreover, it does not inherently provide formal robustness guarantees. This limitation is reinforced in [9], which shows that many defenses could be bypassed by adaptive adversaries. An alternative line of defense involves adversarial purification through reconstruction. For example, AdvPurRec leverages diffusion denoising probabilistic models to remove adversarial perturbations by projecting adversarial inputs onto a distribution that better aligns with the decision boundary of the original model [4].

Certified defenses are another class of methods that aims to provide provable robustness guarantees against adversarial perturbations within specified bounds. Exact verification methods have used techniques such as mixed integer linear programming [36] or satisfiability modulo theories [22] to certify invariance within a radius around a given input. Certified training methods incorporate robustness guarantees directly into the training process; for example, [41] proposes a method based on convex relations, while [31] uses semi-definite relaxations to provide theoretical robustness guarantees. Despite their strong formal properties, these approaches often face scalability challenges for large models and complex tabular tasks due to restrictive constraints on model architectures and the computational burden of certification during training.

Recent work has adapted smoothing to heterogeneous tabular features, particularly in network traffic analysis where feature scales and semantics vary widely. BARS (Boundary-Adaptive Randomized Smoothing) [39] proposes a distribution transformer that learns a noise distribution tailored to heterogeneous tabular features via gradient-based optimization, and reports improved certified robustness on intrusion detection benchmarks such as Kitsune [25], CADE [42], and ACID [17] [39]. In contrast to BARS, our method focuses on a tabular-general mechanism that adaptively modulates the effective training-time noise based on model uncertainty, aiming to improve certified robustness while preserving model accuracy across different tabular domains, including NIDS datasets and URL phishing detection. MARS (Multi-order Adaptive Randomized Smoothing) [21] further explores adaptive noise design for intrusion detection by incorporating feature sensitivity and multi-order transformations, and reports robustness improvements under both adversarial and natural perturbations in the NIDS setting. Overall, these studies highlight the promise of smoothing-based certification for security-critical tasks in specific domain (NIDS), while motivating tabular-general methods that improve the robustness–accuracy trade-off under formal guarantees. Table 1 provides an overview of the recent defense strategies adopted in ML/DL-based deployments.

Table 1. Defense Techniques for ML/DL-based Deployments Against Adversarial Attacks

Defense Category	Approach	Key Features	Limitations
Adversarial Training	Min-max optimization [24]	Projected gradient descent for strong adversarial examples	Computational overhead, reduced clean accuracy
	Gaussian augmentation [32]	High confidence prediction mechanisms for real-time defense	Limited formal robustness guarantees
	AdvPurRec [4]	Diffusion denoising probabilistic models for purification	Can be bypassed by adaptive adversaries [9]
Certified Defense	Exact verification [36]	Mixed integer linear programming [22]	Does not scale to complex network sizes
	Certified training [31]	Convex relations [41], semidefinite relaxations	Significant constraints on architecture and training
	BARS [39]	Boundary-adaptive randomized smoothing with distribution transformer	Learns a global transformed noise distribution, limited near decision boundaries
	MARS [21]	Multi-order adaptive randomized smoothing for heterogeneous features	portability beyond NIDS may require re-engineering

3 Attacker Threat Model

We consider untargeted evasion attacks against tabular deep learning classifiers enhanced with randomized smoothing. Each record is represented as a tabular feature vector $x = (x_{\text{cont}}, x_{\text{cat}})$, where continuous features $x_{\text{cont}} \in \mathbb{R}^d$ may be perturbed within feature-specific ranges, while categorical and binary features $x_{\text{cat}} \in \{0,1\}^m$ remain fixed. For a correctly labeled instance (x, y), the adversary produces a perturbed sample $\hat{x}$ such that $\hat{x} \in \mathcal{A}(x)$ and $h(\hat{x}) \neq y$, where h denotes the victim model. The admissible perturbation set is defined as

$$\mathcal{A}(x) = \{\, \hat{x} : \hat{x}_{\text{cat}} = x_{\text{cat}},\ \|\hat{x}_{\text{cont}} - x_{\text{cont}}\|_2 \leq \delta,\ \hat{x}_{\text{cont}} \in [l, u] \,\}, \tag{1}$$

with $[l, u]$ denoting valid ranges and δ the attack perturbation budget. We instantiate a range of constrained-aware adversarial attacks, including gradient-based (FGSM, PGD, DeepFool, CW), saliency-based (JSMA), a tabular-specific (LowProFool), and query-based attacks (ZOO, HopSkipJump), all adapted to enforce $\delta_{\text{cat}} = 0$ for categorical and binary features. The adversary in the white-box setting has full access to model parameters and defense mechanism, whereas in the black-box setting only query access to outputs is assumed. We do not consider training-time poisoning, backdoor insertion, or denial-of-service threats.

4 Certified Robustness via Adaptive Randomized Smoothing for Tabular Models

We propose a proactive defense mechanism that offers certified robustness guarantees for neural classifiers within a provable radius R around an input x. The method builds upon original randomized smoothing (ORS) [15], but introduces two contributions specifically motivated by the characteristics of tabular data. Unlike images, where features share a common pixel-value domain, tabular inputs comprise heterogeneous features with varying scales, strict value boundaries, and mixed types (continuous, categorical, and binary). Applying a fixed, uniform noise level across all samples disproportionately distorts features near their valid boundaries, degrading clean accuracy. Our first contribution, entropy-guided adaptive noise injection, addresses this by modulating noise magnitude according to the predictive certainty of each sample, reducing unnecessary distortion on uncertain, boundary-proximate inputs while reinforcing robustness on confident predictions. Our second contribution, margin-confidence adaptive certification sampling, targets the variable confidence levels that heterogeneous feature types produce. Mixed-type inputs yield a wider spread of prediction margins than homogeneous image inputs, making a fixed sampling budget inefficient. Together, these mechanisms are not a direct transplant from the vision domain but are new design choices tailored to the structural characteristics of tabular classification. Table 2 presents the formal notations used in our defense approach.

4.1 Randomized Smoothing Preliminaries

Randomized smoothing constructs a smoothed classifier g from a base classifier f by assigning to an input x the class most likely to be predicted when x is perturbed by Gaussian noise:

$$g(x) = \arg\max_{c} \ \mathbb{P}(f(x+\epsilon) = c), \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I). \tag{2}$$

To certify a prediction, n noisy samples are drawn, the base classifier f is evaluated, and the empirical counts are computed for the most frequent (c_A) and second most frequent (c_B) classes. A statistical hypothesis test determines whether c_A is significantly more likely than c_B, based on a Clopper–Pearson (Beta) confidence interval:

$$\underline{p_A} = \text{BetaInv}(1-\alpha, n_A, n-n_A+1), \tag{3}$$

where α is the significance level, n is the total number of samples, and n_A is the number of samples classified as c_A. The lower bound $\underline{p_A}$ represents the minimum proportion of times f is expected to predict c_A under noise with confidence level $1-\alpha$.

The certified radius is then computed as

$$R = \frac{\sigma}{2}\left(\Phi^{-1}(p_A) - \Phi^{-1}(p_B)\right), \tag{4}$$

Table 2. Notations

Notation	Description
$x, \hat{x}, \tilde{x}, y$	Original, adversarial, noisy example, class
x_{cont}, x_{cat}	Continuous, categorical/binary features
δ	Adversarial perturbation
f, g	Base classifier, smoothed classifier
σ	Noise hyperparameter
p_A, p_B	Probability of the most frequent class and the runner-up class
$\underline{p_A}$	Lower bound of p_A estimated via Monte Carlo sampling
$\Phi^{-1}(\cdot)$	Inverse Gaussian CDF
ϵ	Gaussian noise
α	Significance level (failure probability) of the confidence interval
λ	Consistency regularization weight
γ	Adaptive sampling scaling parameter
CE, CI	Cross-entropy, confidence interval
R	Certified radius
$H(x)$	Entropy
n_0	Number of noisy samples used to select the top class
n_{base}	Number of noisy samples used to estimate class probabilities and compute the robustness certificate

where σ is the noise standard deviation, p_A is the probability that the smoothed classifier predicts the most likely class A, p_B is the probability of the runner-up class, and $\Phi^{-1}(\cdot)$ is the inverse Gaussian CDF. If $\underline{p_A} \leq 0.5$, the classifier abstains from prediction to preserve correctness guarantees.

If a classifier consistently predicts the same label under Gaussian perturbations, then any adversarial perturbation bounded within radius R cannot change the prediction. This implies invariance under perturbations satisfying $\|x - \hat{x}\| \leq R$, where $\hat{x}$ is an adversarial example.

4.2 Entropy-Guided Adaptive Noise Injection

Standard randomized smoothing applies a fixed noise level σ to all training samples, regardless of model uncertainty. We instead propose entropy-guided noise scaling, where noise magnitude is modulated according to the Shannon entropy of the predictive distribution:

$$H(p(x)) = -\sum_{i=1}^{k} p_i(x) \log p_i(x), \quad 0 \leq H(p) \leq \log k, \tag{5}$$

with $p_i(x)$ the softmax probability for class i, and k the number of classes. Low entropy indicates high model confidence, while high entropy indicates uncer-

tainty. The adaptive scaling factor is

$$\text{scale}(x) = 1 - \frac{H(p(x))}{\log k}. \tag{6}$$

The perturbed input during training is then

$$\tilde{x} = x + \sigma \cdot \text{scale}(x) \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \tag{7}$$

Here, σ controls the base noise magnitude and ϵ is the sampled Gaussian noise. Together, Equations 5–7 define the complete quantitative rule governing per-sample noise adjustment. The Shannon entropy of the predictive distribution (Eq. 5) is normalized to yield a continuous scaling factor in $(0, 1]$ (Eq. 6), which modulates the base noise level σ to produce the per-sample perturbation (Eq. 7). Concretely, a fully confident prediction ($H = 0$) yields $\mathit{scale}(x) = 1$, applying the full noise magnitude σ, whereas a maximally uncertain prediction ($H = \log k$) yields a reduced but non-zero scaling factor due to the imposed noise floor, preventing noise from being completely suppressed. Intermediate entropy values produce proportionally scaled noise. Since $0 < \mathit{scale}(x) \leq 1$, we guarantee that $\|\sigma \cdot \mathit{scale}(x)\| \leq \sigma$, ensuring that injected noise never exceeds that of standard smoothing and that certification guarantees remain valid. This strategy assigns stronger perturbations to low-entropy (high-confidence) inputs to enhance robustness, while reducing noise for high-entropy (borderline) inputs, thereby preserving clean accuracy.

We note that, consistent with ORS [15], ARS does not correct samples that the base classifier already misclassifies with high confidence. For such samples, the low entropy of the (incorrect) prediction produces a high scaling factor, meaning ARS applies a large noise magnitude. Despite this, if the base classifier consistently predicts the wrong class under repeated noise perturbations, the smoothed classifier may still certify the incorrect label. However, the certification procedure provides a partial safeguard. A sample is certified only if the smoothed classifier predicts the same class with statistical confidence $1 - \alpha$. If the noisy predictions are sufficiently split between classes, the lower bound $p_{\underline{A}}$ falls below 0.5, and the certifier abstains rather than issuing a certificate for an incorrect label. This limitation is inherent to randomized smoothing; certification guarantees robustness of the predicted class, not its correctness. Therefore, if a model is consistently misclassified under noise, ORS may certify an incorrect label.

4.3 Confidence and Margin-Based Adaptive Certification Sampling

In standard randomized smoothing [15], the number of Monte Carlo samples n during certification is fixed across all inputs, regardless of classifier confidence. This is inefficient; inputs far from the decision boundary require fewer samples, while borderline cases require more.

We introduce a proxy difficulty metric based on the top-2 softmax probabilities. Let

$$m(x) = p_{(1)}(x) - p_{(2)}(x), \tag{8}$$
$$c(x) = p_{(1)}(x), \tag{9}$$

where $p_{(1)}$ and $p_{(2)}$ are the top-1 and top-2 predicted probabilities. The *margin* $m(x)$ measures separation between the best two classes; small margins indicate proximity to the decision boundary. The *confidence* $c(x)$ captures absolute certainty in the top class. Using margin alone can be misleading, since two samples may share the same margin but differ in absolute confidence. Thus, we combine both signals: low $m(x)$ and low $c(x)$ indicate harder inputs that require additional sampling, whereas high values retain the base sampling budget.

We define a continuous scaling factor:

$$\text{scale}_n(x) = \max\left(1.0, (1 - m(x)) \cdot (1 - c(x)) \cdot \gamma\right), \tag{10}$$

where $\gamma > 1$ expands the range of the difficulty score beyond 1 for uncertain inputs, and the floor 1.0 ensures that every input receives at least n_{base} samples. The total sampling budget per input is then

$$n(x) = \lceil n_{\text{base}} \cdot \text{scale}_n(x) \rceil . \tag{11}$$

This policy allocates more samples to inputs with small margins and low confidence, which—by Lemma 1—correspond to higher variance in the estimate of $p_A(x)$, while confident inputs retain the baseline sampling budget.

Theoretical Justification

Lemma 1. *For a fixed total sampling budget, the allocation that minimizes the average half-width of binomial confidence intervals is proportional to the Bernoulli variance $p_A(1 - p_A)$. Since $p_A(1 - p_A)$ is maximized when $p_A \approx 0.5$, more samples should be allocated to inputs with smaller margins and lower confidence.*

Proof. Under the normal approximation to the binomial proportion, the half-width of a $(1 - \alpha)$ CI is

$$\text{HW}(x) \approx z_{1-\alpha/2} \sqrt{\frac{p_A(x)(1 - p_A(x))}{n(x)}}, \tag{12}$$

so minimizing average half-width under a budget constraint yields $n(x) \propto p_A(x)(1 - p_A(x))$ by the method of Lagrange multipliers.

Suppose we estimate the probability of predicting class A on input x by sampling from the randomized smoothing distribution. Let $p_A(x)$ denote the true

probability and $n(x)$ the number of allocated samples. For m inputs $\{x_1, \ldots, x_m\}$ and total budget N, we solve

$$\min_{n(x_1),\ldots,n(x_m)} \frac{1}{m} \sum_{i=1}^{m} \mathrm{HW}(x_i), \tag{13}$$

subject to

$$\sum_{i=1}^{m} n(x_i) = N, \qquad n(x_i) \geq 0 \ \ \forall i. \tag{14}$$

The optimal allocation is

$$n(x_i) \propto p_A(x_i)\big(1 - p_A(x_i)\big), \tag{15}$$

so inputs with higher variance receive proportionally more samples.

The ARS method is inherently multi-class, as both the entropy-based noise scaling and certification procedure operate over general k-class distributions. We focus on binary settings to isolate robustness behavior in security-critical scenarios, where adversarial robustness and false positives are most critical. To further validate generality, we include additional experiments on multi-class settings in Appendix A.

5 Evaluation Details

Datasets and pre-processing. To provide a comprehensive evaluation of the performance of our ARS algorithm, we selected three benchmark tabular datasets spanning multiple application contexts. Specifically, we evaluated our approach on two intrusion-detection benchmarks, UNSW-NB15 [27], CSE-CIC-IDS2018 (CICIDS2018) [33], and the web page phishing detection dataset (URL) [19]. The UNSW-NB15 and CICIDS2018 datasets provide a comprehensive coverage of network-based attacks including DoS, DDoS, botnet traffic, and attempts at network infiltration. The URL dataset complements this evaluation by providing a non-intrusion detection tabular setting focused on phishing detection, where inputs are derived from URL and website characteristics. Evaluating across these three datasets enables a broader assessment of our method across different tabular domains and feature distributions. To prepare the datasets for model training, we preprocessed the data by applying min-max normalization to the numerical features and one-hot encoding to the categorical features. We then split the datasets into training and testing sets where 80% of the data was used for training and the remaining 20% was used for testing.

Initial model training and evaluation. Using the preprocessed data, we trained both a Tabular Neural Network (TabNet) [6] and a Vanilla DNN. Both models were trained to perform binary classification on each dataset such that they can distinguish between Benign and Attack traffic. We additionally evaluate ARS in a multi-class setting using the original ten-class UNSW-NB15 label structure in Appendix A. We selected TabNet for its specialized architecture

designed specifically for handling tabular data. The DNN serves as a baseline model to evaluate the effectiveness of our algorithm in different deployment contexts, and has shown a strong performance on intrusion detection benchmarks [5, 14].

Both models were trained using cross-entropy loss and the Adam optimizer with an initial learning rate of 0.001 to ensure consistent training conditions across both model architectures. We trained the TabNet model for 10 epochs, whereas the DNN was trained over 30 epochs. The DNN architecture consisted of three hidden layers with 128, 64, and 32 neurons, respectively. After each layer, we applied batch normalization and ReLU activation functions. Additionally, we applied dropout regularization with a rate of 0.25 after the first hidden layer to prevent overfitting of the model. Both models were trained to achieve high classification accuracy on non-perturbed datasets, yielding three pre-trained TabNet models and three pre-trained DNNs. This step is essential as it establishes a baseline performance, ensuring that our initial classifiers achieve strong discriminative ability between Benign and Attack traffic prior to evaluating their robustness against various adversarial perturbations.

Randomized smoothing setup. To fairly compare ORS and ARS methods, we used the same base classifier (DNN/TabNet) and optimizer hyperparameter settings [5] in both methods, except the noise augmentation magnitude. During the training, ORS injects fixed-σ Gaussian noise, while ARS uses entropy-guided noise $\sigma(x) = \sigma \cdot scale(x)$, where $scale(x)$ is an entropy-based scaling factor normalized by the number of classes, lower-bounded by a noise floor of 0.2, and modulated by a ramp schedule. Noise injection is activated after half of the training epochs and increased linearly over 5 ramp epochs. To stabilize learning under noisy inputs, we included a consistency regularization λ that penalizes differences between the model's predictions on clean and noisy inputs that penalizes differences between the model's predictions on clean and noisy inputs weighted by $\lambda = 0.1 \times$ ramp. A sensitivity analysis of these hyperparameters is provided in Appendix B.

During certification, both methods use the same fixed noise level σ, pilot samples n_0, and nominal estimation budget n_{base} at significance level α. However, they differ in how the estimation samples are allocated: ORS uses a fixed number of Monte Carlo samples ($n = n_{\text{base}}$) to estimate class probabilities and compute confidence bounds, whereas ARS dynamically scales the estimation sample size on a per-input basis ($n \geq n_{\text{base}}$) according to a confidence and margin proxy, allocating more samples to uncertain points and fewer to confident ones. See Table 3 for detailed smoothing/training hyperparameter settings.

Experimental environment. Experiments were conducted on a workstation running Windows 11, equipped with an Intel Core Ultra 9 285K processor, 64 GB of RAM, and an NVIDIA RTX 5090 GPU. The implementation was developed in Python using PyTorch, with adversarial examples generated using the Adversarial Robustness Toolbox (ART) [28].

[5] We used a StepLR scheduler that halves the learning rate every 10 epochs to stabilize convergence.

Table 3. Experimental Hyperparameters: Smoothing/Training (Left) and Adversarial Attack Settings (Right).

Smoothing / Training		**Adversarial Attacks**		
Hyperparameter	**ORS. | ARS.**	**Atk**	**Param**	**Value**
Optimizer	AdamW	PGD	`eps`	0.15
Learning rate	10^{-3}	PGD	`eps_step`	0.01
Weight decay	10^{-4}	DF	`max_iter`	10
Scheduler	StepLR(10,γ=0.5)	FGSM	`eps`	0.15
Batch size	64	LPF	`lambd`	0.5
Epochs	30	ZOO	`conf`	0.8
Training noise	fixed σ | $\sigma(x)$	HSJ	`max_iter`	64
Cert. noise σ	0.3	CW	`max_halving`	5
Pilot samples n_0	200	CW	`max_doubling`	5
Samples n	1000 | ≥ 1000	JSMA	θ	0.1
Significance α	0.001	JSMA	γ_s	1.0
λ, noise floor, ramp epoch	0.1, 0.2, 5			

Adversarial attacks. We generated eight attack-dataset combinations using the Adversarial Robustness Toolbox (ART) [28], a commonly used Python library for ML security. We evaluated each of the pre-trained model-dataset combinations using different threat models, including six white-box attacks and two black-box attacks, to demonstrate their effectiveness and adaptability. We presented the adversarial attack hyperparameter used to generate adversarial examples (AEs) for each attack in Table 3. All unlisted parameters with respect to each attack were initialized using the default ART values. We selected these attacks due to their widespread use in adversarial machine learning research [5, 14, 43, 32].

Projected Gradient Descent (PGD): A white-box attack that maximizes the model's loss while keeping perturbations close to the original example, constrained within l_p norm [24]. PGD starts with a sample, applies a random perturbation within the l_p ball, and iteratively takes gradient steps to increase loss, projecting back to the l_p ball after each step until convergence.

Fast Gradient Sign Method (FGSM): A white-box attack that finds minimal perturbations for misclassification [18]. FGSM computes the loss gradient with respect to input data, then takes the sign of this gradient, multiplied by a small value (ϵ) to determine the direction of perturbation. Using the gradient sign instead of the full gradient makes FGSM a fast attack.

Jacobian-based Saliency Map (JSMA): A white-box attack that targets salient features to misclassify towards a target class [29]. JSMA initializes a Jacobian matrix of partial derivatives of output class probabilities with respect to input features. Using a saliency map, JSMA identifies and modifies the most influential features iteratively until the target class probability becomes sufficiently high or a maximum number of features is modified.

Carlini and Wagner (CW): A white-box optimization-based attack seeking minimal perturbations for maximal misclassification [11]. CW iteratively refines AEs to enhance misclassification confidence while maintaining imperceptibility. The high transferability of CW AEs allows them to deceive models with different architectures.

LowProFool (LPF): A white-box attack developed to generate AEs for tabular data by iteratively applying perturbations to an original sample until the perturbed sample is misclassified by the classifier [10]. The algorithm also incorporates feature importance to guide the perturbation process, prioritizing features based on their significance to the classifier. Feature importance is calculated using Pearson correlation, where greater perturbations are applied to less important features and smaller perturbations to more important ones. This approach ensures that the AEs are both effective and minimally distorted.

DeepFool (DF): A white-box attack that iteratively finds minimal perturbations required to cross the decision boundary and induce misclassifcation [26]. At each iteration, it applies the calculated perturbation and repeats the process until the sample crosses the decision boundary with the goal of adaptively minimizing the final perturbation.

Zeroth Order Optimization (ZOO): A black-box attack that approximates gradients when they are unavailable or expensive [13]. ZOO uses output probabilities to estimate gradients via finite differences, iteratively creating adversarial examples until success or a maximum iteration count is reached.

HopSkipJump (HSJ): A black-box decision-based attack that begins with an original sample and perform a binary search to find the decision boundary [12]. HSJ attacks estimate gradient direction and perturb samples accordingly. This process is facilitated by geometric progression, which adjusts the step size during the search: larger steps are used when the perturbed point is far from the boundary, and smaller steps when it is close. This approach enhances query efficiency by avoiding unnecessary computational resource expenditure.

For each attack-dataset combination, we measure the classification accuracy of both the DNN and TabNet models before and after implementing any evasion attack with respect to each model and source dataset. In Table 4, we present the results which serve as a baseline to evaluate, on one hand, the model's vulnerability to different attack categories, and, on the other hand, the extent to which our defense enhances model robustness against each attack scenario. In addition to reporting the baseline model performance, we also provide the dataset statistics, including the number of features before and after applying one-hot encoding to the categorical features as well as the total number of records for each of the datasets.

Evaluation metrics. we used three metrics to evaluate our adaptive randomized smoothing technique.

Certified Accuracy (CertAcc) measures the proportion of inputs for which the smoothed classifier is both correct and provably robust within a certified radius. A sample contributes to *CertAcc* only if all of the following conditions are met: 1) The smoothed prediction is correct, i.e., $g(x_i) = y_i$; 2) The certified radius is

Table 4. Dataset Statistics and Baseline Model Accuracy.

Model	UNSW-NB15	CICIDS2018	URL
DNN	99.7%	99.9%	95.0%
TabNet	99.5%	99.9%	92.6%
Features (pre)	43	69	63
Features (post)	204	71	63
Records	235,050	1,156,152	11,358

"Features (pre)" denotes the original number of features in the dataset. "Features (post)" denotes the number of features after preprocessing; one-hot encoding of categorical features.

strictly positive, $R(x_i) > 0$; 3) The Monte Carlo hypothesis test confirms, with confidence level $1 - \alpha$, that the top class is statistically dominant over the other class. Therefore, *CertAcc* measures prediction correctness *together with* a formal robustness guarantee.

Accuracy (Acc) measures the accuracy of the *base classifier* f on adversarial samples, i.e., the proportion of inputs for which the clean (unsmoothed) model predicts the correct label. This metric reflects standard predictive performance without any smoothing or certification.

Average Certification Time (Avg_ cert_ time) measures the computational cost of randomized smoothing. For each fold and each attack, we recorded the total wall-clock time T_{cert} (in seconds) required to certify all samples, and computed the per-sample certification time in milliseconds according to the following equation:

$$\text{cert\_ms\_per\_sample} = 1000 \times \frac{T_{\text{cert}}}{N_{\text{test}}}, \tag{16}$$

where N_{test} is the number of test samples in that fold. We report *Avg_ cert_ time* as the mean of these per-sample certification times across the ($K = 5$) cross-validation folds.

6 Results and Analysis

The following experiments were designed to evaluate the robustness of our ARS algorithm through answering the following Research Questions (RQs):

RQ1: To what extent does the proposed ARS defense improve model accuracy across different evasion attack types compared to undefended models?

RQ2: How does the proposed ARS technique compare to the ORS in terms of model accuracy under different adversarial attacks?

RQ3: To what extent does the ARS defense preserve model accuracy on clean examples compared to the ORS method?

RQ4: What is the overhead introduced by the ARS technique compared to the ORS method with respect to training time?

RQ5: How does ARS compare with the state-of-the-art randomized smoothing techniques in terms of robustness across diverse evasion attacks?

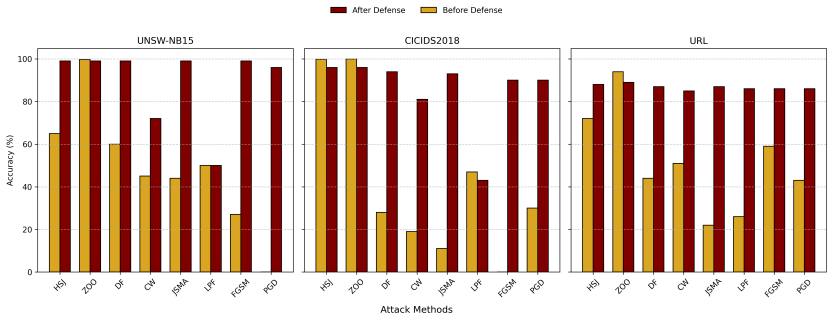

Fig. 1. Certified Accuracy for DNN Model Before and After Defense Across Each Attack Method

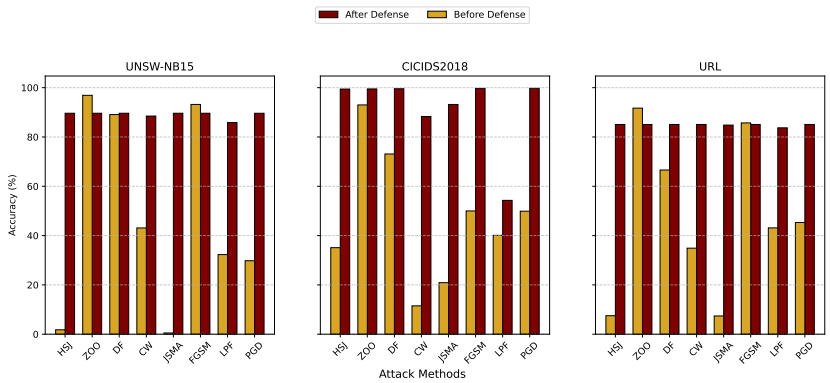

Fig. 2. Certified Accuracy for TabNet Model Before and After Defense Across Each Attack Method

RQ1. Improvement in model accuracy against various evasion attack scenarios. To answer RQ1, we implemented our defense on both the TabNet and DNN architectures and evaluated their resilience against eight state-of-the-art evasion attacks across three benchmark datasets. Fig. 1 and Fig. 2 present the results for the DNN and TabNet models, respectively, comparing model accuracy before and after applying ARS. Both figures clearly demonstrate substantial improvements in adversarial accuracy, which confirms the effectiveness of our approach in mitigating diverse attack strategies.

The results further reveal that, for our DNN model, different attack algorithms exploit model vulnerabilities in distinct ways, leading to varying levels of success in reducing accuracy. Gradient-based white-box attacks such as PGD, FGSM, CW, JSMA and DF are particularly aggressive because they directly leverage gradient information to maximize the misclassification probability. Consequently, these attacks tend to craft highly optimized perturbations that effectively degrade model performance. Nonetheless, our defense successfully coun-

tered these attacks, recovering model certified accuracy by an average of 68%, 65%, 41%, 67%, and 49% for PGD, FGSM, CW, JSMA, and DF respectively across the three datasets.

In contrast, black-box query-based attacks such as ZOO or decision-based methods like HSJ are generally less aggressive. Since they cannot access gradients, these attacks rely on repeated model queries to approximate the decision boundary and optimize surrogate objectives. This approximation process typically produces weaker perturbations compared to white-box methods, making such attacks less effective in practice. As a result, ARS maintains high adversarial accuracy under black-box settings, which demonstrates robustness even when the attacker adapts to realistic query-based constraints.

When evaluating these attacks on our TabNet model, we found that ARS primarily improved the certified accuracy in cases where the baseline model fails most severely, while occasionally introducing small drops for attacks where the baseline was already strong. On the UNSW-NB15 dataset, the baseline is already high for ZOO/DF/FGSM (89%-97%), but it collapses for HSJ/JSMA (0-2%) and CW/LPF/PGD (30-45%). After applying the defense, accuracy becomes consistently high across all attacks (86-90%), with only minor decreases on a couple of the already-strong baseline cases. On CICIDS2018, the baseline was particularly vulnerable to CW, JSMA, and HSJ where we observed the accuracy drop below 40%. However, after retraining the model with ARS we observed near perfect certified accuracy (99%) for HSJ, ZOO, DF, FGSM, and PGD. The main remaining weakness is LPF, which had moderate improvement after ARS with accuracy improving by 13%. Finally, on the URL datasets, ARS raised the accuracy of the weakest baseline cases (HSJ/JSMA at 7% and CW/LPF/PGD at roughly 35-45%) up to a consistent 84-85%, while slightly reducing performance for attacks where the baseline was already high (ZOO/FGSM).

RQ2. Comparison between our ARS technique and ORS. To investigate RQ2, we evaluated the robustness of our proposed ARS technique against the ORS. Both defenses are tested under eight evasion attack algorithms, each implemented in two distinct scenarios: (i) *non-constrained* attacks, where adversarial perturbations are generated without respecting dataset-specific feature constraints, and (ii) *constrained-aware* attacks, where categorical and binary features are preserved via masking, and only continuous features are perturbed. This scenario preserves feature semantics and validity, ensuring that perturbations remain realistic. Testing under these conditions provides a more faithful assessment of our defense's robustness in real world scenarios.

We reported four key evaluation metrics; Acc-NC and Acc-C, where they denoted model accuracy under non-constrained and constraint-aware adversarial attacks, respectively, both measured before applying any defense. CertAcc-ARS and CertAcc-ORS denoted the certified accuracy after applying our ARS and the ORS defense, respectively.

Table 5 reports the experimental results for the DNN architecture. We observed in some attack cases that Acc-C is consistently higher than Acc-NC. This outcome stems from the constrained-aware setting, where the adversarial per-

Table 5. DNN Adversarial Accuracy (%) Comparison Across Each Dataset and Attack

Dataset	Metric	PGD	FGSM	CW	JSMA	DF	LPF	ZOO	HSJ
UNSW-NB15	Acc-NC	0	27	45	44	60	49	99.7	65
	Acc-C	0	46	82	58	89	50	99.7	99.7
	CertAcc-ARS	96	99	72	99	99	50	99	99
	CertAcc-ORS	96	99	79	99	99	54	99	99
CICIDS2018	Acc-NC	30	0	19	11	28	47	99.9	99.8
	Acc-C	46	41	36	28	74	50	99.9	50
	CertAcc-ARS	90	90	81	93	94	43	96	96
	CertAcc-ORS	81	81	81	81	81	30	82	82
URL	Acc-NC	43	59	51	22	44	26	94	72
	Acc-C	40	56	47	26	45	28	95	73
	CertAcc-ARS	90	91	85	87	87	87	89	88
	CertAcc-ORS	90	90	85	87	86	87	87	87

turbations that violate the original tabular data constraints are discarded and replaced with the original values. As a result, the effective perturbation space is reduced, making constrained-aware adversarial examples less aggressive but more realistic [3].

Our ARS defense consistently outperforms or performs equally to the ORS across nearly all attack types and datasets, with the exception of CW and LPF on UNSW-NB15. In these cases, the ORS slightly outperforms the adaptive variant when defending against these attacks by 7% and 4%, respectively. LPF, being a targeted and highly effective tabular attack, requires training with substantially higher levels of injected noise to mitigate its impact. However, such aggressive noise injection is not incorporated into the adaptive smoothing method, as it comes with the major drawback of significantly degrading model accuracy on clean, unperturbed examples.

Across all three datasets, ARS records substantial improvements particularly on CICIDS2018. Specifically, JSMA, DF, LPF, HSJ, and ZOO achieve 12%, 13%, 13%, 14%, and 14% accuracy improvements, respectively, when compared to ORS, the ARS method demonstrates model accuracy improvement by 9% on gradient descent attacks, PGD and FGSM. We also note that black-box query-based attacks ZOO and HSJ were largely ineffective against the DNN on CICIDS2018 datasets. Despite this, the ORS method incurred a significant reduction in model accuracy by 17%, thus compromising robustness in the absence of strong adversarial threat. In contrast, our ARS approach preserved model accuracy while maintaining comparable robustness which underscores its ability to balance defense strength and clean model accuracy.

In Table 6, we report the experimental results for the TabNet architecture. When examining the comparative performance under the non-constrained (Acc-NC) and the constraint-aware (Acc-C) attack settings, we found similar results

Table 6. TabNet Adversarial Accuracy (%) Comparison Across Each Dataset and Attack

Dataset	Metric	PGD	FGSM	CW	JSMA	DF	LPF	ZOO	HSJ
UNSW-NB15	Acc-NC	30	93	43	0.1	89	33	96	2
	Acc-C	99	99.3	44	82	99.5	68	99.5	99.5
	CertAcc-ARS	90	90	89	90	90	86	90	90
	CertAcc-ORS	90	90	89	89	90	89	89	89
CICIDS2018	Acc-NC	50	50	12	21	73	41	93	35
	Acc-C	50	50	12	30	73	41	93	39
	CertAcc-ARS	99.7	99.7	89	93	99.6	54	99.5	99.5
	CertAcc-ORS	94	94	89	94	94	93	94	94
URL	Acc-NC	45	85	35	7	66	43	92	7
	Acc-C	45	85	35	7	66	43	92	7
	CertAcc-ARS	85	85	85	85	85	84	85	85
	CertAcc-ORS	85	85	85	84	84	84	85	85

to the DNN model, where the constraint-aware attacks generally yielded higher or equal accuracy across each of the datasets. When assessing the certified accuracies of the original smoothing and our method, we observe that ARS generally outperforms ORS across most of the datasets and attack types, with some exceptions. On UNSW-NB15, ARS achieved comparative performance on all attacks except for LPF where we observed a reduction in accuracy of 3%. On CICIDS2018, ARS outperformed ORS for PGD, FGSM, DF, ZOO, and HSJ with accuracy improvements ranging between 5.5%-5.7%. However, ARS underperformed when defending against the JSMA and LPF attacks where we observe an accuracy reduction of 1% and 39%, respectively, compared to ORS. The 39% drop under LPF is specific to the TabNet–CICIDS2018 combination and does not generalize across datasets. On UNSW-NB15 the gap is only 3%, on URL it is less than 1%, and for the DNN the maximum reduction is 7% across all three datasets (Tables 5–6). LPF concentrates perturbations on low-importance features, producing adversarial examples that fall in high-entropy regions of the feature space, where our entropy-guided scaling reduces noise magnitude. TabNet's attention mechanism amplifies this effect on CICIDS2018 by assigning soft, distributed attention weights across features, which increases predictive entropy on LPF-perturbed inputs and consequently reduces the effective noise applied during smoothing. Regarding CW, the slight underperformance of ARS on UNSW-NB15 (7% below ORS for the DNN) arises because CW produces minimal, tightly optimized perturbations that remain close to the decision boundary. These perturbations yield moderate entropy values that receive intermediate noise scaling, which is insufficient to shift predictions back across the boundary. Lastly, on the URL dataset, we see comparable performance overall

when compared to the ORS with accuracies ranging within 1% between methods.

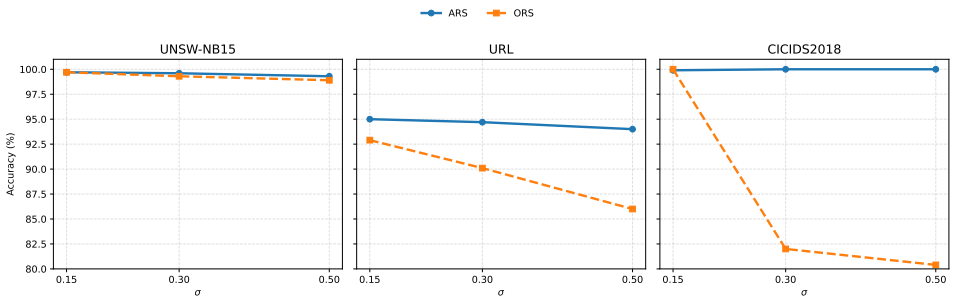

Fig. 3. DNN Model Accuracy on Clean Data for ARS and ORS Across Different σ.

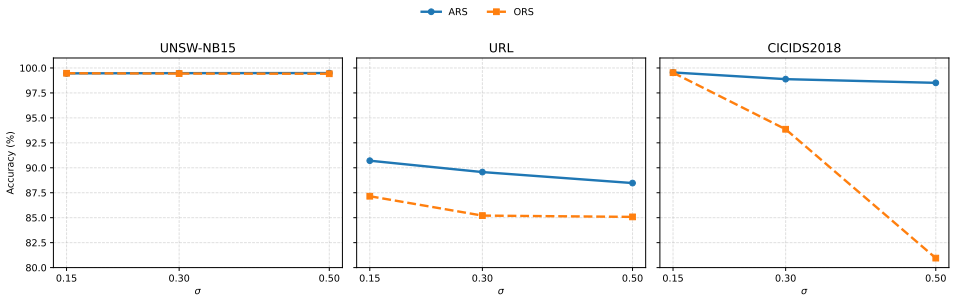

Fig. 4. TabNet Model Accuracy on Clean Data for ARS and ORS Across Different σ.

RQ3. Defense impact on clean examples. To answer RQ3, we implemented a comparative analysis between our ARS and the ORS technique with respect to their ability to accurately classify clean non-perturbed examples. Maintaining high model accuracy on clean examples is a crucial requirement for any defense technique. Some defense methods, while attempting to purify adversarial examples, inadvertently degrade the accuracy of clean inputs. This issue is particularly critical in real-world tabular deployments, where even a slight increase in the false positive rate can impose substantial downstream costs such as additional manual review, delayed decisions, and reduced trust in the model.

Fig. 3 and 4 present the performance of both defense techniques in preserving accuracy on clean examples for the DNN and TabNet models. We reported model accuracy on clean examples while varying the noise magnitude used for training and certification, with $\sigma \in \{0.15, 0.30, 0.50\}$. Across all datasets, ARS

consistently preserves high clean accuracy as σ increases, remaining close to the non-smoothed baseline.

For the URL and CICIDS2018 datasets, ARS substantially outperforms ORS. In particular, ORS exhibits a pronounced degradation in clean accuracy as σ increases, with drops of up to 5–10% on URL and 17–20% on CICIDS2018 at $\sigma = 0.30$ and $\sigma = 0.50$. In contrast, ARS maintains near-baseline performance across all noise levels (see Table 4 for the baseline accuracy), demonstrating strong robustness to training time noise injection. In UNSW-NB15, both methods achieve similar clean accuracy in all σ values, indicating that this dataset is less sensitive to noise augmentation. Overall, these results show that entropy-guided adaptive noise injection effectively mitigates the accuracy degradation commonly observed with ORS. Notably, the clean accuracy drops exhibited by ORS in Figures 3 and 4 directly correspond to increased false positive rates. Samples that the unsmoothed model correctly classifies as benign are misclassified as attacks after ORS is applied, constituting Type I errors. For example, on CICIDS2018 at $\sigma = 0.5$, ORS reduces clean accuracy by up to 20% relative to the unsmoothed baseline (Table 4), indicating that a substantial proportion of benign inputs are incorrectly flagged. In contrast, ARS maintains near-baseline clean accuracy across all noise levels and datasets, thereby preserving low false positive rates while still providing certified robustness guarantees. We further validate this finding in the multi-class setting in Appendix C.

RQ4. Efficiency of our adaptive defense method. To answer RQ4, we further compared the computational efficiency of ARS and ORS by measuring the average *Avg_ cert_ time* reported in Fig. 5. For the DNN results on UNSW-NB15, CICIDS2018, and URL, ARS consistently achieves lower certification latency than ORS across all datasets. Specifically, ARS reduces the average certification time from 9.87 ms to 8.20 ms on UNSW-NB15, from 9.87 ms to 7.69 ms on CICIDS2018, and from 10.25 ms to 8.11 ms on URL. This corresponds to relative speedups[6] of approximately 17%, 22%, and 21%, respectively.

In contrast, the TabNet results show a dataset-dependent overhead trend: on UNSW-NB15 and CICIDS2018 ARS requires higher average certification time than ORS (33.5ms vs. 27.7ms and 32.6ms vs. 28.23ms, respectively), while on URL the two methods are nearly identical (51.7ms vs. 52.5ms). This is due to ARS increasing the sampling budget $n(x)$ for inputs with small margin $m(x)$ and low confidence $c(x)$ (Eqs. 10–11), whereas ORS uses a fixed sampling budget. TabNet's sequential attention mechanism produces softer probability distributions than the DNN, particularly on UNSW-NB15 and CICIDS2018, where the heterogeneous feature distributions cause attention weights to spread across multiple features rather than concentrating on a few discriminative ones. This results in a higher proportion of inputs with small margins and low confidence scores, which triggers additional resampling under ARS and increases the average certification time (33.5 ms vs. 27.7 ms on UNSW-NB15 and 32.6 ms vs. 28.23 ms on CICIDS2018). In contrast, on the URL dataset, TabNet's predic-

[6] We reported certification speedup as the relative reduction in the average certification time *Avg_ cert_ time* compared to ORS.

tions are well-separated with consistently high margins, so ARS allocates close to the base budget and its certification time matches ORS more closely (51.7 ms vs. 52.5 ms). This confirms that the overhead introduced by adaptive sampling is not a fixed cost but arises only when the model's predictive uncertainty warrants additional samples, representing a trade-off between certification reliability and computational efficiency.

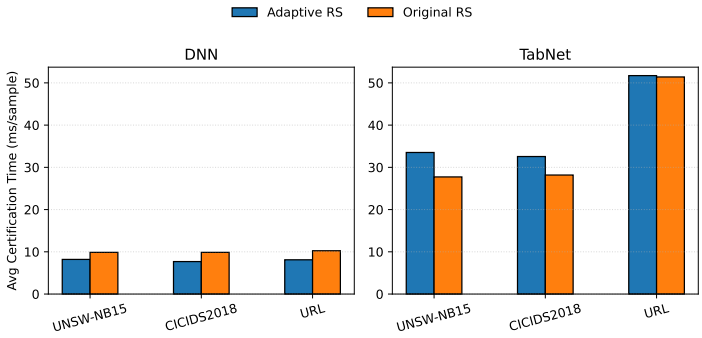

Fig. 5. Average Certification Time per Sample (ms) for ARS and ORS.

RQ5. Comparison with other approaches.[7] We compared the certified accuracy of four defense strategies—ARS, ORS, ORS with clean training (ORS-Z)[8], and BARS—under eight evasion attacks across three datasets using DNN model, see Fig. 6. To ensure a controlled comparison, we evaluate all methods using the same base architecture, data splits, and adversarial loaders, and we align the certification budget by using the same $(\sigma, n_0, n_{\text{base}}, \alpha)$ across defenses; for BARS, we integrate its noise-transformation module into our pipeline while replacing BARS' original IDS-specific backbone with our tabular DNN to ensure a controlled comparison. While existing certification techniques primarily emphasize maximizing the certified radius, our objective is to improve model robustness with formal guarantees, as reflected by certified accuracy under attack. ARS consistently achieves the highest certified accuracy across most attacks and datasets, indicating stronger robustness in practice. In contrast, ORS-Z, which applies randomized smoothing without noise augmented training, exhibits substantial performance degradation under stronger attacks, highlighting the importance of noise aware training when robustness is the primary goal. Although ORS improves robustness relative to ORS-Z, it still suffers noticeable accuracy

[7] We restrict this experiment to the DNN to keep the comparison focused on randomized smoothing variants rather than the effects of changing the model architecture.

[8] ORS-Z corresponds to the standard randomized smoothing evaluation: the base classifier is trained on clean data, and smoothing is applied only at certification time. Noise augmented training is optional and not required by the original randomized smoothing formulation.

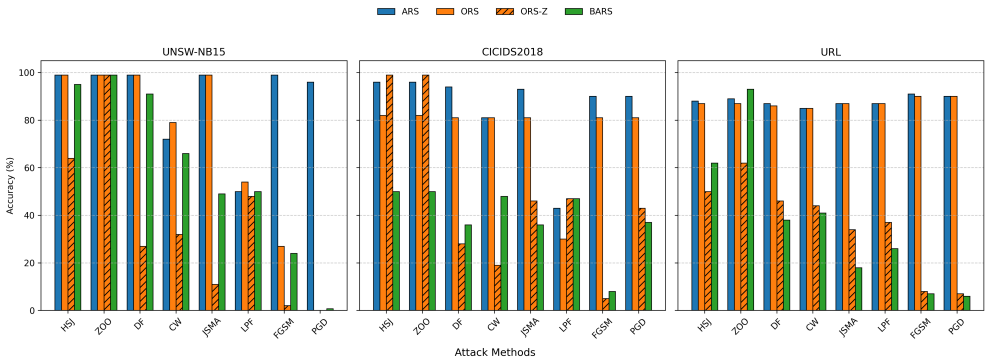

Fig. 6. Certified Accuracy Comparison of Four Defense Strategies Under 8 Evasion Attacks

drops compared to ARS, particularly on feature-sensitive attacks such as FGSM, JSMA, and PGD in UNSW-NB15 and CICIDS2018, and also degrades clean accuracy. BARS and ORS-Z generally preserve clean accuracy but fail to maintain robustness under stronger adversarial perturbations. Overall, these results demonstrate that ARS best preserves the trade-off between certified robustness and clean predictive accuracy among existing randomized smoothing baselines. We extend this comparison to the multi-class setting in Appendix D.

7 Conclusion and Future Work

In this paper, we advance randomized smoothing for tabular classification by introducing an adaptive variant that accounts for model uncertainty and data characteristics. Our method combines entropy-guided noise injection with margin–confidence–based adaptive sampling, leading to improved robustness against a wide range of adversarial attacks while preserving accuracy on clean inputs.

Although our primary analysis focuses on binary classification, we further validate the effectiveness of our approach in multi-class settings, demonstrating its applicability across tasks of varying complexity. The proposed framework is model-agnostic and compatible with different architectures, including standard deep neural networks and tabular-specific models such as TabNet. This flexibility is particularly important for real-world tabular data, which often exhibit heterogeneous feature types and complex dependencies.

Looking forward, an important direction for future work is extending adaptive randomized smoothing to account for both adversarial perturbations and distribution shifts. In many real-world tabular applications, such as network traffic analysis, data distributions evolve over time due to concept drift. Existing certification methods typically assume static distributions and may fail to provide reliable guarantees under such conditions. Developing certification frameworks that jointly address adversarial robustness and distributional changes is

therefore a key step toward building trustworthy and deployable tabular ML/DL systems.

References

1. Maged Abdelaty, Sandra Scott-Hayward, Roberto Doriguzzi-Corin, and Domenico Siracusa. Gadot: Gan-based adversarial training for robust ddos attack detection. In *2021 IEEE Conference on Communications and Network Security (CNS)*, pages 119–127. IEEE, 2021.
2. James Aiken and Sandra Scott-Hayward. Investigating adversarial attacks against network intrusion detection systems in sdns. In *2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 1–7. IEEE, 2019.
3. Nour Alhussien, Gagan Agrawal, and Ahmed Aleroud. Augmented tabular adversarial evasion attacks with constraint satisfaction guarantees. In *International Conference on Availability, Reliability and Security*, pages 365–386. Springer, 2025.
4. Nour Alhussien and Ahmed Aleroud. Advpurrec: Strengthening network intrusion detection with diffusion model reconstruction against adversarial attacks. In *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1638–1646. IEEE, 2024.
5. Nour Alhussien, Ahmed Aleroud, Abdullah Melhem, and Samer Y Khamaiseh. Constraining adversarial attacks on network intrusion detection systems: transferability and defense analysis. *IEEE Transactions on Network and Service Management*, 21(3):2751–2772, 2024.
6. Sercan O. Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning, 2020.
7. Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. Dos and don'ts of machine learning in computer security, 2021.
8. Lirim Ashiku and Cihan Dagli. Network intrusion detection system using deep learning. *Procedia Computer Science*, 185:239–247, 2021.
9. Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, 2018.
10. Vincent Ballet, Jonathan Aigrain, Thibault Laugel, Pascal Frossard, Marcin Detyniecki, et al. Imperceptible adversarial attacks on tabular data. In *NeurIPS 2019 Workshop on Robust AI in Financial Services: Data, Fairness, Explainability, Trustworthiness and Privacy (Robust AI in FS 2019)*, 2019.
11. Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.
12. Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, pages 1277–1294. IEEE, 2020.
13. Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.
14. Alesia Chernikova and Alina Oprea. Fence: Feasible evasion attacks on neural networks in constrained environments. *ACM Transactions on Privacy and Security*, 25(4):1–34, 2022.

15. Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing, 2019.
16. Jacopo Cortellazzi, Feargus Pendlebury, Daniel Arp, Erwin Quiring, Fabio Pierazzi, and Lorenzo Cavallaro. Intriguing properties of adversarial ml attacks in the problem space [extended version], 2024.
17. Alec F. Diallo and Paul Patras. Adaptive clustering-based malicious traffic classification at the network edge. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021.
18. Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.
19. Abdelhakim Hannousse and Salima Yahiouche. Towards benchmark datasets for machine learning based website phishing detection: An experimental study. *Engineering Applications of Artificial Intelligence*, 104:104347, 2021.
20. Vanlalruata Hnamte, Hong Nhung-Nguyen, Jamal Hussain, and Yong Hwa-Kim. A novel two-stage deep learning model for network intrusion detection: Lstm-ae. *Ieee Access*, 11:37131–37148, 2023.
21. Mengdie Huang, Yingjun Lin, Xiaofeng Chen, and Elisa Bertino. Mars: Robustness certification for deep network intrusion detectors via multi-order adaptive randomized smoothing. In *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 767–774, 2024.
22. Guy Katz, Clark Barrett, David Dill, Kyle Julian, and Mykel Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks, 2017.
23. Nattawat Khamphakdee, Nunnapus Benjamas, and Saiyan Saiyod. Improving intrusion detection system based on snort rules for network probe attack detection. In *2014 2nd International Conference on Information and Communication Technology (ICoICT)*, pages 69–74. IEEE, 2014.
24. Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.
25. Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. Kitsune: An ensemble of autoencoders for online network intrusion detection, 2018.
26. Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016.
27. Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*, pages 1–6, 2015.
28. Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. Adversarial robustness toolbox v1.2.0. *CoRR*, 1807.01069, 2018.
29. Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.
30. Marek Pawlicki, Michał Choraś, and Rafał Kozik. Defending network intrusion detection systems against adversarial evasion attacks. *Future Generation Computer Systems*, 110:148–154, 2020.
31. Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *6th International Conference on Learning Representa-*

tions, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018.
32. Khushnaseeb Roshan, Aasim Zafar, and Shiekh Burhan Ul Haque. Untargeted white-box adversarial attack with heuristic defence methods in real-time deep learning based network intrusion detection system. *Computer Communications*, 218:97–113, 2024.
33. Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *4th International Conference on Information Systems Security and Privacy (ICISSP)*, pages 108–116, Portugal, 2018.
34. Nathan Shone, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. A deep learning approach to network intrusion detection. *IEEE transactions on emerging topics in computational intelligence*, 2(1):41–50, 2018.
35. Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.
36. Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming, 2019.
37. Muhammad Usama, Muhammad Asim, Siddique Latif, Junaid Qadir, and Ala-Al-Fuqaha. Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 78–83, 2019.
38. Jianyu Wang, Jianli Pan, Ismail AlQerm, and Yuanni Liu. Def-ids: An ensemble defense mechanism against adversarial attacks for deep learning-based network intrusion detection. In *2021 international conference on computer communications and networks (ICCCN)*, pages 1–9. IEEE, 2021.
39. Kai Wang, Zhiliang Wang, Dongqi Han, Wenqi Chen, Jiahai Yang, Xingang Shi, and Xia Yin. Bars: Local robustness certification for deep learning based traffic analysis systems. In *Network and Distributed System Security (NDSS) Symposium*, San Diego, CA, USA, 2023. Internet Society.
40. Arkadiusz Warzyński and Grzegorz Kołaczek. Intrusion detection systems vulnerability on adversarial examples. In *2018 Innovations in Intelligent Systems and Applications (INISTA)*, pages 1–4. IEEE, 2018.
41. Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5283–5292. PMLR, 2018.
42. Limin Yang, Wenbo Guo, Qingying Hao, Arridhana Ciptadi, Ali Ahmadzadeh, Xinyu Xing, and Gang Wang. CADE: Detecting and explaining concept drift samples for security applications. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2327–2344. USENIX Association, August 2021.
43. Chaoyun Zhang, Xavier Costa-Perez, and Paul Patras. Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms. *IEEE/ACM Transactions on Networking*, 30(3):1294–1311, 2022.

A ARS Adaptation to Multi-class Classification

To evaluate the generalizability of ARS beyond binary classification, we extended our evaluation to the multiclass setting using the UNSW-NB15 dataset with its original ten-class label structure, as detailed in Table 7. For this experiment, we adopted a stronger DNN architecture to better handle the increased complexity of the ten-class task. Specifically, the DNN classifier consists of three fully connected hidden layers with 256, 128, and 64 neurons, respectively, followed by a ten-class output layer; ReLU activations are used after each hidden layer, and dropout with rate 0.2 is applied after the first two hidden layers to improve generalization. Using this architecture, we evaluated ARS and ORS against all eight evasion attacks. Table 8 reports the results.

On seven of the eight attacks, ARS and ORS achieve certified accuracy within 3 percentage points of each other: ARS holds a slight advantage on PGD and DF (+1 percentage point each), ORS leads on FGSM, JSMA, and HSJ (+3, +2, and +1 percentage points, respectively), and the two methods are tied on ZOO and CW. The cases where ORS slightly outperforms ARS reflect the increased difficulty of entropy-based noise scaling in the multi-class setting, where the wider entropy range causes the scaling factor to distribute noise more conservatively. The most notable difference appears on LPF, where ARS outperforms ORS by 17 percentage points (47% vs. 30%). This contrasts with the binary setting, where ORS generally held an advantage on LPF, and suggests that the wider entropy range in the multi-class setting may better differentiate LPF-perturbed inputs from clean ones, allowing entropy-guided noise scaling to allocate noise more effectively. These results confirm that ARS generalizes to multi-class tasks without architectural modification.

Table 7. Dataset Statistics for UNSW-NB15 in Multi-class Setting

Attack Category	**Training**	**Testing**	**Total**
Analysis	2,000	677	2,677
Backdoor	1,746	583	2,329
DoS	12,264	4,089	16,353
Exploits	33,393	11,132	44,525
Fuzzers	18,184	6,062	24,246
Generic	40,000	18,871	58,871
Normal	56,000	37,000	93,000
Reconnaissance	10,491	3,496	13,987
Shellcode	1,133	378	1,511
Worms	130	44	174
Total	**175,341**	**82,332**	**257,673**

Table 8. DNN Adversarial Accuracy (%) Comparison on Multi-Class UNSW-NB15

Dataset	Metric	PGD	FGSM	JSMA	DF	ZOO	CW	LPF	HSJ
UNSW-NB15	Acc-NC	34	35	33	55	74	59.5	16	58
	Acc-C	61	62	54	63	60	15.5	63	63
	CertAcc-ARS	64	60	57	64	64	60	47	63
	CertAcc-ORS	63	63	59	63	64	60	30	64

B Additional Experimental Setup Details

We performed a sensitivity analysis over $\lambda \in \{0.05, 0.1, 0.2\}$, noise floor $\in \{0.1, 0.2, 0.3\}$, and ramp epochs $\in \{3, 5, 7\}$. As shown in Fig. 7, certified accuracy varies by at most 1.8 percentage points across all tested configurations, confirming that ARS is robust to hyperparameter selection. The chosen configuration (λ=0.1, noise floor=0.2, ramp=3) provides the best trade-off between robustness and certified accuracy. These experiments were conducted on the URL dataset.

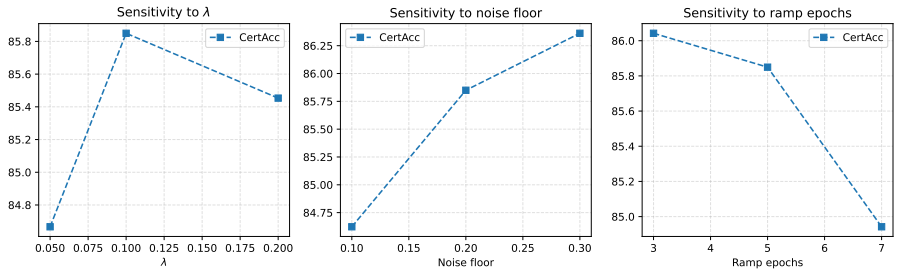

Fig. 7. Sensitivity Analysis of Key ARS Hyperparameters for URL.

C Clean Accuracy and Adaptive Noise Behavior in the Multi-class Setting

To complement the binary classification results for RQ3, we evaluate the impact of ARS and ORS on clean accuracy in the multi-class UNSW-NB15 setting. We first observe that the DNN achieves an accuracy of 76% in the multi-class setting, compared to 99.7% in the binary setting. This performance gap is primarily attributable to differences in data distribution. In the binary setting, the dataset is balanced, which simplifies the classification task and enables the model to learn more stable decision boundaries. In contrast, the multi-class setting exhibits significant class imbalance, with several attack categories being underrepresented, thereby increasing task difficulty and reducing overall accuracy. This highlights

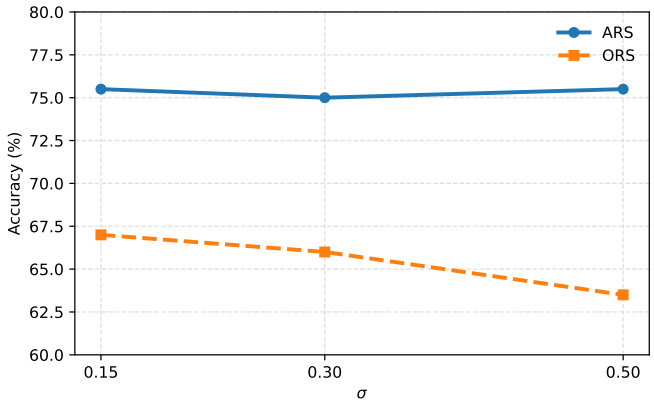

Fig. 8. ARS vs. ORS Clean Accuracy Under Different σ Values on UNSW-NB15.

that performance in the multi-class setting reflects a more challenging and realistic scenario rather than a limitation of the model. Figure 8 shows clean accuracy under varying noise magnitudes $\sigma \in \{0.15, 0.30, 0.50\}$. ARS maintains stable clean accuracy at approximately 75% across all noise levels, whereas ORS degrades from approximately 67% at $\sigma = 0.15$ to approximately 63% at $\sigma = 0.50$. This confirms that the clean accuracy preservation observed in the binary setting (Figures 3–4) generalizes to multi-class classification, and that the accuracy gap between ARS and ORS widens as noise increases, demonstrating the robustness of ARS under more challenging and imbalanced conditions.

To further analyze the behavior of entropy-guided noise injection, we log the effective training-time noise $\boldsymbol{\sigma(x)}$ across epochs for the DNN model. Figure 9 illustrates the evolution of $\boldsymbol{\sigma(x)}$ in ARS using grouped box plots over non-overlapping 5-epoch windows. During the ramp phase (epochs 15–19), the distribution of $\boldsymbol{\sigma(x)}$ gradually expands, rising from approximately 0.03 at epoch 15 to approximately 0.34 at epoch 19. This reflects a controlled introduction of noise that avoids strong perturbations early in training.

In subsequent epochs (20–24), $\boldsymbol{\sigma(x)}$ spans a wide range across samples, with minimum values near the noise floor (0.1) in this experiment for uncertain inputs and maximum values approaching the global $\boldsymbol{\sigma(x)}$ (0.5) for confident predictions. This behavior indicates that the model selectively applies stronger smoothing where it is most appropriate, while remaining conservative for fragile examples.

In the final training stage (epochs 25–29), the distribution of $\boldsymbol{\sigma(x)}$ stabilizes, suggesting that the adaptive mechanism converges to a steady allocation of noise across samples. This adaptive behavior helps explain the improved clean accuracy observed with ARS; the model avoids excessive noise on fragile inputs during training while still enforcing smoothness in confident regions. We further observe that the variance of $\boldsymbol{\sigma(x)}$ increases from a small value early in training (0.006) to a non-trivial level at convergence (0.11), confirming that the model

does not collapse to fixed-noise augmentation as in ORS, but instead consistently allocates noise based on prediction confidence.

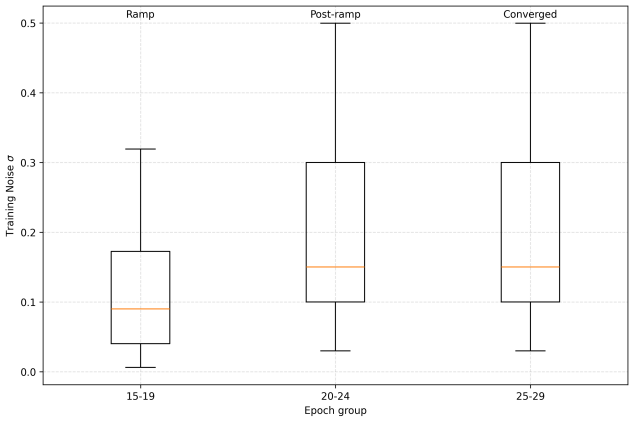

Fig. 9. Grouped Distribution of Adaptive σ over Training for DNN Model.

D Comparison with Other Defense Approaches on Multi-Class Setting for UNSW-NB15

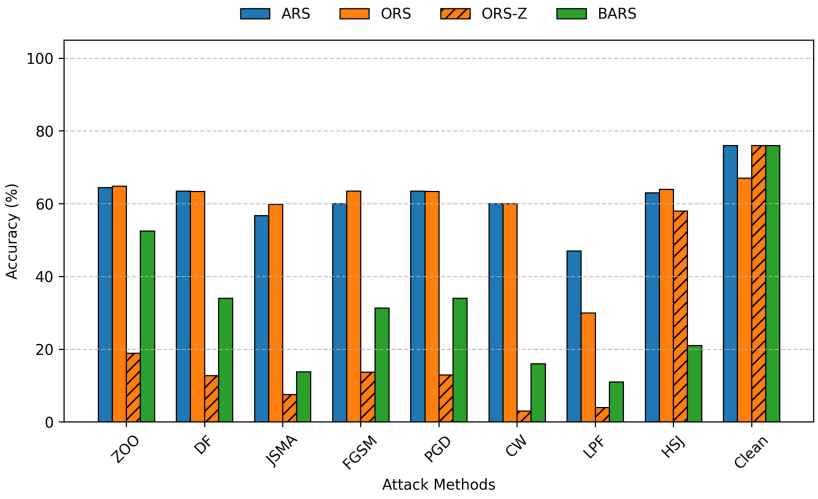

Fig. 10. Certified Accuracy Comparison of Four Defense Approaches Under 5 Evasion Attacks in Multi-Class Setting

To extend the RQ5 analysis to multi-class classification, we compare ARS, ORS, ORS-Z, and BARS under all eight evasion attacks on the multi-class UNSW-NB15 dataset using the DNN model. Figure 10 reports the results alongside clean accuracy. ARS and ORS achieve comparable certified accuracy on most attacks, with both methods reaching approximately 64–65% on ZOO, DF, PGD, and CW, and differences of at most 4 percentage points on FGSM, JSMA, and HSJ. The most notable divergence is on LPF, where ARS outperforms ORS by 17 percentage points (47% vs. 30%). However, ORS achieves this adversarial performance at a significant cost to clean accuracy: ORS degrades clean accuracy to approximately 67%, whereas ARS preserves it at approximately 76%, matching ORS-Z and BARS. Both ORS-Z and BARS suffer substantial robustness degradation under adversarial attacks, with BARS falling below 53% in every case and ORS-Z below 20% on six of eight attacks, with the exception of HSJ (approximately 58%) and ZOO (approximately 19%). These results confirm that, consistent with the binary setting, ARS provides the most favorable trade-off between adversarial robustness and clean accuracy preservation among the evaluated defenses, and that this property generalizes to multi-class tabular classification.