STOCHASTIC ODE-GUARD: CERTIFIED AND PATHWISE ROBUSTNESS BOUNDS FOR NEURAL SDE CLASSIFIERS IN INTRUSION DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural intrusion detectors classify network flows from a few dozen traffic statistics. They lose much of their accuracy when an attacker nudges those statistics slightly, and the defences that come with formal guarantees either treat the classifier as a fixed function or add noise to its input. We study *Stochastic ODE-Guard* (SODE-Guard), a neural stochastic differential equation (NSDE) classifier: it maps a flow to an internal representation, lets that representation evolve under a learned random dynamics, and classifies by averaging over the random outcomes. We prove two robustness results for this model class. (i) Theorem A: the averaged classifier, exactly as it is computed with the Euler–Maruyama scheme, changes by at most L times the size of any input change, with L computed from the trained weights. A score gap $M(x)$ between the top two classes therefore guarantees the same decision for every input change smaller than $M(x)/(\sqrt{2}L)$; a Monte-Carlo confidence correction makes the guarantee usable with finitely many samples. (ii) Proposition 2: for a single random outcome and any fixed input change, the probability that the decision flips is at most the probability of starting within β of the decision boundary plus $2L^2\varepsilon^2/\beta^2$; our anti-concentration regulariser is designed to shrink the first term. We also state exactly when a Bismut–Elworthy–Li estimator recovers input gradients of such a model (Proposition 1). On three public flow benchmarks SODE-Guard attains 96.4% clean macro-F1 and 93.1% under PGD-40 at $\epsilon{=}0.03$, against 87.2% for the strongest neural baseline, at 1.7 ms median latency per batch of 128 on one A100.

1 INTRODUCTION

Network intrusion detection and prevention systems (IDPS) inspect traffic and decide, flow by flow, whether to allow or block it. A *flow* is the set of packets exchanged between two endpoints in one session; modern detectors do not read packet contents but summarise each flow by statistics such as its duration, packet counts, byte counts, inter-arrival times and protocol flags. Neural IDPS trained on such summaries exceed 0.97 F1 on benchmarks such as UNSW-NB15 and CIC-IDS2017 (Wu et al., 2022; Halbouni et al., 2022). They are also fragile: small, deliberately chosen changes to the statistics can remove more than fifteen F1 points (Ennaji et al., 2024; Vitorino et al., 2024), and structural attacks on graph-based detectors evade deployed systems (Venturi et al., 2024). An attacker who can pad packets or add delays controls some of these statistics, so the question is practical.

Two families of defences come with formal guarantees. *Lipschitz-bounded networks* (Tsuzuku et al., 2018; Anil et al., 2019; Araujo et al., 2023) limit how fast the output can change with the input; *randomised smoothing* (Cohen et al., 2019; Salman et al., 2019; Yang et al., 2020) classifies many noisy copies of the input and takes a majority vote. The first treats the classifier as a fixed map; the second perturbs the input itself and needs large sample budgets at inference. Continuous-time models offer a third route. Neural ordinary differential equations (ODEs) inherit stability from their flow map (Kang et al., 2021; Guglielmi et al., 2025), and neural *stochastic* differential equations (SDEs) (Tzen & Raginsky, 2019; Li et al., 2020; Kidger et al., 2021) add a learned noise term that can be well posed and carry calibrated uncertainty (Oh et al., 2024; Bergna et al., 2024). What such a model guarantees under adversarial input changes has been left open.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

This paper. SODE-Guard keeps the input flow untouched: it encodes the flow into a 128-dimensional state, lets that state evolve under a learned SDE, and classifies by averaging a linear read-out over several random trajectories (Figure 1). We ask what can be *certified*, that is, proved to hold for every admissible input change, about such a classifier.

Contributions. **(C1)** A worst-case certificate for the trajectory-averaged classifier (Theorem A). It applies to the Euler–Maruyama scheme that is actually computed and uses a constant computed from the trained weights; a Monte-Carlo confidence correction (Corollary 1, Algorithm 1) turns it into a procedure that either returns a guaranteed radius or abstains. **(C2)** A path-wise decision-flip bound (Proposition 2) that avoids a union over the 34 classes and whose leading term is the quantity our anti-concentration regulariser penalises during training. **(C3)** A precise statement of when a Bismut–Elworthy–Li (BEL) estimator with a regularised pseudo-inverse recovers input gradients of an SDE with fewer noise channels than state dimensions (Proposition 1). **(C4)** An evaluation on three public flow benchmarks against six neural baselines under the same threat model. Code, the certifier, all attacks and all data-split protocols are in an anonymised repository (`https://anonymous.4open.science/r/SODE-ExtractGuard-Models`).

Roadmap. Section 2 defines the problem, the threat model and the notion of certified robustness, and recalls the SDE background needed later. Section 4 describes SODE-Guard, Section 5 states and explains the guarantees, Section 6 reports experiments, and Sections 7 and 8 discuss limits and conclude. Proofs are in Section C, and Section A is a glossary of terms.

2 Problem setting and background

Table 1 collects the notation used throughout.

Table 1: Notation.

Symbol	Meaning	Symbol	Meaning
$x \in \mathbb{R}^p$	flow features, $p = 83$, z-scored	K	number of classes (34)
δ, ε	input perturbation, budget	$\|\cdot\|_2$, $\|\cdot\|_\infty$	Euclidean, max-coordinate norm
h	encoder; $X_0 = h(x) \in \mathbb{R}^d$, $d = 128$	W_t	Brownian motion in $\mathbb{R}^m$, $m = 16$
f_θ, g_θ	drift ($\mathbb{R}^d$), diffusion ($\mathbb{R}^{d\times m}$)	T, Δt, n	horizon 1, step 0.05, $n = 20$ steps
X_k	state after k EM steps	$\psi_\eta(z) = W_\psi z + b_\psi$	linear head giving K logits
$F(x)$	mean logits $\mathbb{E}[\psi_\eta(X_n)]$	$\hat{y}(x)$	predicted class $\arg\max_k F_k(x)$
$M(x)$	margin $F_{\hat{y}}(x) - \max_{k\neq\hat{y}} F_k(x)$	$G(x)$	margin of one random path
K_f, K_g, L_h	Lipschitz constants of f_θ, g_θ, h	L	Lipschitz constant of F
$r(x)$	certified radius $M(x)/(\sqrt{2}L)$	N_{mc}, n_{c}	paths: inference (8), certification
λ_0	diffusion floor, 10^{-3}	α	failure probability of a certificate

2.1 Flows, detectors and margins

Each flow is described by $p = 83$ statistics extracted by a CICFlowMeter-compatible tool. Every statistic is *z-scored*: we subtract its training-set mean and divide by its training-set standard deviation, so one unit means one standard deviation. A detector maps x to K real scores (*logits*); the softmax turns them into class probabilities and the predicted class is the largest one. The *margin* is the gap between the largest and the second-largest logit; a large margin means a confident decision. Classes are the benign class and 33 attack types, so accuracy is reported as *macro-F1*: the F1 score (the harmonic mean of precision and recall) computed for each class separately and then averaged, which prevents the frequent benign class from dominating.

2.2 Threat model

An adversary replaces x by $x + \delta$ with $\|\delta\| \le \varepsilon$. Two norms are used. The ℓ_∞ norm $\|\delta\|_\infty = \max_i |\delta_i|$ bounds the change of every single statistic; the ℓ_2 norm $\|\delta\|_2 = (\sum_i \delta_i^2)^{1/2}$ bounds the

total change. For 83 features they are related by $\|\delta\|_\infty \le \|\delta\|_2 \le \sqrt{83}\,\|\delta\|_\infty$, so an ℓ_∞ budget of 0.03 (three hundredths of a standard deviation per feature) lies inside an ℓ_2 ball of radius 0.273. We assume a *white-box* adversary who knows the model and its weights. The standard attack is projected gradient descent (PGD, Madry et al., 2018): starting from a random point in the budget, it repeats $x^{(j+1)} = \Pi_\varepsilon\big(x^{(j)} + a\,\mathrm{sign}\,\nabla_x \ell(x^{(j)}, y)\big)$ for 40 steps (PGD-40), where ℓ is the cross-entropy loss, a a step size and Π_ε the projection back into the budget. Because SODE-Guard is random, a fair attacker averages gradients over many random draws (expectation over transformation, EOT, Athalye et al., 2018); we return to this in Section 6. We perturb the extracted features, as prior NIDS work does; whether a feature change can be realised by real packets is a separate question discussed in Section 7.

2.3 Certified robustness

Attacks give *empirical* robustness: they show that one attacker failed. A certificate is a proof that no attacker within the budget can succeed.

Definition 1 (Certified radius and certified accuracy)**.** A number $r(x) \ge 0$ is a *certified ℓ_2 radius* of a classifier $\hat{y}$ at x if $\hat{y}(x+\delta) = \hat{y}(x)$ for every $\|\delta\|_2 \le r(x)$. The *certified accuracy* at radius r is the fraction of test flows that are classified correctly and have certified radius at least r.

A function F is *L-Lipschitz* if $\|F(x) - F(x')\|_2 \le L\|x - x'\|_2$ for all x, x': the output moves at most L times as far as the input. The Lipschitz constant of a linear map $z \mapsto Wz$ is its *spectral norm* $\|W\|$, the largest singular value of W; a composition of maps has Lipschitz constant at most the product of the individual constants. *Spectral normalisation* (Miyato et al., 2018) divides each weight matrix by an estimate of its spectral norm during training so that $\|W\| \le 1$. If F is L-Lipschitz, then every difference of two logits is $\sqrt{2}L$-Lipschitz, and a margin $M(x)$ cannot be closed by any δ with $\|\delta\|_2 < M(x)/(\sqrt{2}L)$ (Tsuzuku et al., 2018). Our first result applies this principle to a predictor defined as an average over random trajectories.

2.4 Stochastic differential equations in brief

A *Brownian motion* $W_t \in \mathbb{R}^m$ models accumulated Gaussian noise: over a short time Δt its increment ΔW is Gaussian with mean zero and covariance $\Delta t\, I_m$, and increments over disjoint intervals are independent. An Itô SDE

$$dX_t = f_\theta(X_t, t)\,dt + g_\theta(X_t, t)\,dW_t, \qquad X_0 = x_0, \tag{1}$$

moves the state $X_t \in \mathbb{R}^d$ in the direction of the *drift* f_θ and adds noise shaped by the *diffusion* g_θ, a $d \times m$ matrix. When f_θ, g_θ are Lipschitz with at most linear growth, (1) has a unique solution (Øksendal, 2003, Thm. 5.2.1). On a computer it is solved by the *Euler–Maruyama* (EM) scheme on the grid $t_k = k\Delta t$, $k = 0, \dots, n = T/\Delta t$:

$$X_{k+1} = X_k + f_\theta(X_k, t_k)\,\Delta t + g_\theta(X_k, t_k)\,\Delta W_k, \qquad \Delta W_k \sim \mathcal{N}(0, \Delta t\, I_m) \text{ independent.} \tag{2}$$

One run of (2) with one draw of the increments is a *path*. Expectations over paths are estimated by *Monte-Carlo* averaging over independent paths; the estimate's error shrinks like one over the square root of the number of paths. All our results are stated for (2), the scheme that is actually run, rather than for the idealised continuous SDE.

3 Related work

Certified robustness. Lipschitz-bounded networks (Tsuzuku et al., 2018; Anil et al., 2019; Wang & Manchester, 2023; Araujo et al., 2023) certify radii that scale with the inverse Lipschitz constant. Randomised smoothing (Cohen et al., 2019) and its anisotropic extensions (Yang et al., 2020; Eiras et al., 2022) give high-probability ℓ_2 certificates; PixelDP (Lecuyer et al., 2019) links Gaussian smoothing to differential privacy. Theorem A is a Lipschitz certificate, but for a predictor defined as an expectation over latent SDE paths, with the constant obtained from a mean-square stability estimate of the discretised SDE (Lemma 1). Reliable evaluation of randomised defences needs expectation over transformation (Athalye et al., 2018), parameter-free attack ensembles (Croce & Hein, 2020) and gradient-masking checks (Carlini et al., 2019). **Neural ODEs/SDEs.** SODEF

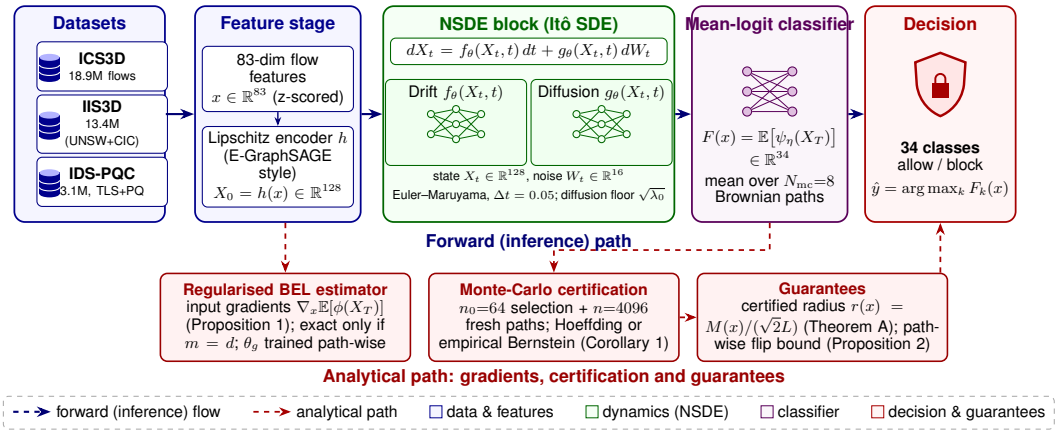


Figure 1: SODE-Guard. Solid arrows: inference. A flow's 83 statistics are encoded into a state X_0, evolved by the learned SDE, read out by a linear head and averaged over $N_{\mathrm{mc}} = 8$ random paths. Dashed arrows: the analytical components. The certificate concerns the averaged predictor F; certification uses $n_{\mathrm{c}} = 4096$ fresh paths.

(Kang et al., 2021) uses Lyapunov-stable equilibria of a neural ODE; Guglielmi et al. (2025) quantify destabilising weight perturbations; stable neural SDEs (Oh et al., 2024) and latent graph neural SDEs (Bergna et al., 2024) establish well-posedness on irregular and graph data; the stochastic adjoint (Li et al., 2020) makes training scalable. We use the diffusion for robustness rather than generation (Song et al., 2021). **Intrusion detection.** E-GraphSAGE (Lo et al., 2022) introduced edge-feature graph networks for NIDS, RTIDS (Wu et al., 2022) adapted transformers, and CNN-LSTM hybrids (Halbouni et al., 2022) and IDS-GraphMamba (Mosaiyebzadeh et al., 2025) refined spatio-temporal modelling. Continuous-time and stochastic detectors with uncertainty estimates have also been proposed (Anaedevha et al., 2026; Anaedevha & Trofimov, 2025); none derives a certificate from the diffusion. **Stochastic networks.** Weight-space uncertainty confers some adversarial robustness (MacKay, 1992; Liu et al., 2019; He et al., 2019); our guarantees concern latent-trajectory noise with fixed weights.

4 The SODE-Guard Classifier

Assumption 1 (Certifiable architecture)**.** *The encoder h, drift f_θ, diffusion g_θ and head ψ_η are built from linear layers and GELU activations, whose slope is at most $c_\phi = \max_u |\mathrm{GELU}'(u)| \approx 1.129$ (Hendrycks & Gimpel, 2016). f_θ, g_θ are three-layer perceptrons; h is a linear projection followed by three residual blocks $z \mapsto \frac{1}{2}\mathrm{GELU}(Uz + c) + \frac{1}{2}z$, with no normalisation layers; every linear layer is spectral-normalised. At every step the diffusion is shifted by the constant $\sqrt{\lambda_0} I_{d \wedge m}$, $\lambda_0 = 10^{-3}$. When one flow is perturbed, the other $B - 1$ flows of its micro-graph are held fixed. L_h, K_f and K_g denote the Lipschitz constants of h, of f_θ in the state, and of g_θ in the state (measured in the Frobenius norm $\|\cdot\|_F$, the root of the sum of squared entries).*

The assumption is what makes the constants computable: under it, $K_f, K_g \le c_\phi^2 \prod_\ell \|W_\ell\|$ and $L_h \le c_\phi \|W_0\| \prod_{j=1}^{3} (\frac{1}{2} c_\phi \|U_j\| + \frac{1}{2})$. Two design choices follow from it. LayerNorm is removed because it has no global Lipschitz bound. The input-dependent gate of the original encoder is replaced by a fixed average, because a product of two input-dependent terms is hard to bound. Spectral normalisation estimates $\|W\|$ with a one-step power iteration during training, so after training the certifier recomputes every $\|W_\ell\|$ exactly by singular-value decomposition. *Scope:* the models evaluated in Section 6 were trained with the original encoder, which does not satisfy Assumption 1. We therefore report no certified radii for them; their empirical results do not rely on Theorem A, and training and certifying the certifiable variant is future work.

Representation, dynamics and prediction. The encoder gives $X_0 = h(x) \in \mathbb{R}^{128}$. The SDE evolves this internal state, so the noise never alters the input flow itself. The dynamics (1) use $d = 128$, $m = 16$ noise channels, hidden width 256 and horizon $T = 1$, integrated by (2) with $\Delta t = 0.05$ (20 steps). The head $\psi_\eta(z) = W_\psi z + b_\psi$ maps the final state to $K = 34$ logits, and the classifier

$$F(x) = \mathbb{E}\big[\psi_\eta(X_n) \,\big|\, X_0 = h(x)\big] \in \mathbb{R}^K, \qquad \hat{y}(x) = \arg\max_k F_k(x), \tag{3}$$

the class with the largest *average* logit (equivalently, the largest softmax of the average logits). Deployment estimates F with $N_{\mathrm{mc}} = 8$ paths.

Training. Let m_i be the top-1 minus top-2 gap of the path-averaged logits of example i in a mini-batch of size B, $\hat{s}$ the ℓ_2 norm of the vector of gaps, and $\sigma_\kappa(u) = (1 + e^{-\kappa u})^{-1}$ a smooth step. The *anti-concentration regulariser*

$$\mathcal{L}_{\mathrm{AC}} = \frac{1}{|\mathcal{B}|} \sum_{\beta \in \mathcal{B}} \log\Big(1 + d^\star (\beta/\hat{s})^{1/d^\star} \tfrac{1}{B} \textstyle\sum_{i=1}^{B} \sigma_\kappa(\beta - |m_i|)\Big), \tag{4}$$

with $\mathcal{B} = \{0.01, 0.025, 0.05, 0.1\}$, $\kappa = 50$ and $d^\star = 4$, counts (smoothly) the fraction of examples whose margin lies within β of the decision boundary and penalises it. The name and the $(\beta/\hat{s})^{1/d^\star}$ weight come from the Carbery–Wright anti-concentration inequality (Carbery & Wright, 2001), which bounds how much probability a polynomial of Gaussian variables can put near zero. The weight is a design choice; no guarantee below depends on it. The loss is $\mathcal{L}_{\mathrm{CE}} + 0.1\,\mathcal{L}_{\mathrm{AC}}$, with the weight 0.1 chosen on the IIS3D validation split. All parameters are trained by backpropagation through (2) with Adam (learning rate 5×10^{-4}, weight decay 10^{-5}, batch 128, cosine annealing, 40 epochs) and five seeds $\{42, 137, 271, 1729, 2026\}$ (Algorithm 2).

Input gradients without differentiating the decision. Attacks and diagnostics need $\nabla_x \mathbb{E}[\phi(X_n)]$ for functions ϕ that are discontinuous, such as "this path misclassifies". The Bismut–Elworthy–Li (BEL) formula (Bismut, 1981; Elworthy & Li, 1994; Hsu, 2002) gives such gradients by re-weighting samples of ϕ instead of differentiating it, but it needs an invertible diffusion, whereas ours has $m = 16 < d = 128$. We use the regularised pseudo-inverse $g_\lambda^+ = (g^\top g + \lambda I_m)^{-1} g^\top$ and the projection $\Pi_\lambda = g\, g_\lambda^+$ onto (approximately) the directions the noise can reach.

Proposition 1 (Regularised BEL estimator)**.** *Let f, g have two bounded derivatives, $J_s = \partial X_s^x / \partial x$ be the sensitivity of the path to its starting point, and, for a direction v, $\mathcal{W}_v = \frac{1}{t}\int_0^t \langle g_\lambda^+(X_s) J_s v, dW_s\rangle$. (i) If $m = d$, g is uniformly invertible and $\lambda = 0$, then $\partial_v \mathbb{E}[\phi(X_t^x)] = \mathbb{E}[\phi(X_t^x)\,\mathcal{W}_v]$ for every bounded measurable ϕ (Elworthy & Li, 1994). (ii) If $m \le d$, $\lambda > 0$ and ϕ is continuously differentiable, $\mathbb{E}[\phi(X_t^x)\,\mathcal{W}_v] = \mathbb{E}\big[\langle \nabla\phi(X_t^x),\, J_t \frac{1}{t}\int_0^t J_s^{-1} \Pi_\lambda(X_s) J_s v\, ds\rangle\big]$, which equals $\partial_v \mathbb{E}[\phi(X_t^x)]$ when $\Pi_\lambda J_s v = J_s v$ for all s. The bias is governed by $\|(I - \Pi_\lambda) J_s v\|$, which vanishes as $\lambda \to 0$ if and only if $J_s v$ lies in the range of $g(X_s)$.*

In words: the estimator is exact only when the noise can push the state in the direction being differentiated. With $m < d$ that is not guaranteed, so we do *not* train with it. SODE-Guard's parameters are trained with ordinary path-wise gradients, and Proposition 1 is used only for input gradients of discontinuous quantities in our attacks.

5 Robustness guarantees

All statements concern the EM scheme (2). Full proofs are in Section C. The key device is *synchronous coupling*: to compare the model at x and at $x + \delta$, run both from their own starting states but with the *same* noise increments ΔW_k. Any difference in the final states is then caused by δ alone.

Lemma 1 (Mean-square stability)**.** *Under Assumption 1, let X_k and Y_k follow (2) from $X_0 = h(x)$ and $Y_0 = h(x + \delta)$ with the same increments. Then*

$$\mathbb{E}\|Y_n - X_n\|^2 \le L_h^2 \|\delta\|^2 \exp\big((2K_f + K_g^2 + K_f^2 \Delta t)\, T\big).$$

Algorithm 1 CERTIFY(x): certified ℓ_2 radius of the mean predictor

Require: flow x; constant L from exact spectral norms; radius B fixed on training data; n_0, n_c, α

1: Draw fresh random seeds from the operating system's cryptographic generator
2: $\hat{y} \leftarrow \arg\max_k$ of the mean of $P_B(\psi_\eta(X_n))$ over n_0 paths {selection}
3: Run n_c new paths; for each $k \neq \hat{y}$ compute $\hat{M}_k$ and t_k {estimation}
4: $\underline{M} \leftarrow \min_{k \neq \hat{y}}(\hat{M}_k - t_k)$
5: **if** $\underline{M} > 0$ **then**
6: **return** $(\hat{y},\ \underline{M}/(\sqrt{2}L))$ {holds with prob. $\geq 1 - \alpha$}
7: **else**
8: **return** ABSTAIN
9: **end if**

Why it holds. In one step the drift can stretch the gap between the two paths by a factor $1 + K_f \Delta t$, and the diffusion adds, on average, variance $K_g^2 \Delta t$ times the squared gap. Multiplying over the n steps and using $1 + u \leq e^u$ gives the exponential. The diffusion therefore does *not* contract paths; it adds K_g^2 to the growth rate. Noise is not free.

Theorem A (Certified radius of the SDE-mean classifier)**.** *Under Assumption 1, let P_B project a logit vector onto the Euclidean ball of radius B, let $F^B(x) = \mathbb{E}[P_B(\psi_\eta(X_n)) \mid X_0 = h(x)]$, and let*

$$L := \|W_\psi\|\, L_h\, \exp\big((K_f + \tfrac{1}{2}K_g^2 + \tfrac{1}{2}K_f^2 \Delta t)\, T\big).$$

Then $\|F^B(x+\delta) - F^B(x)\|_2 \leq L\|\delta\|_2$ for all x, δ. If $\hat{y} = \arg\max_k F_k^B(x)$ and $M(x) = F_{\hat{y}}^B(x) - \max_{k \neq \hat{y}} F_k^B(x) > \sqrt{2}L\varepsilon$, then $\arg\max_k F_k^B(x + \delta) = \hat{y}$ for every $\|\delta\|_2 \leq \varepsilon$. Hence $r(x) = M(x)/(\sqrt{2}L)$ is a certified ℓ_2 radius and $r(x)/\sqrt{83}$ a certified ℓ_∞ radius. With every spectral norm at most one, $L \leq c_\phi(\frac{1}{2}c_\phi + \frac{1}{2})^3 \exp(c_\phi^2 + \frac{1}{2}c_\phi^4 + \frac{1}{2}c_\phi^4 \Delta t) \approx 11.4$.

Proof idea. Average the coupled difference over paths. Averaging cannot increase distances (Jensen's inequality), the projection P_B cannot increase distances, and the head scales them by at most $\|W_\psi\|$; Lemma 1 bounds the rest. A logit gap then moves by at most $\sqrt{2}L\|\delta\|$. The projection P_B keeps every sampled logit vector bounded, which the Monte-Carlo step below needs. We fix B on training data before certification, and whenever B exceeds all logit norms it has no effect ($F^B = F$). Algorithm 1 also records the fraction of path logits that P_B changes.

Worked example. With the analytic bound $L = 11.4$, $\sqrt{2}L = 16.1$. A flow whose average logits separate the top two classes by $M(x) = 4$ is certified up to ℓ_2 radius $4/16.1 = 0.25$, that is, ℓ_∞ radius $0.25/\sqrt{83} = 0.027$ in z-score units. A margin of 6 certifies ℓ_∞ radius 0.041, which covers the $\epsilon = 0.03$ budget used in our attacks. The constant computed from trained weights can be smaller than the analytic bound, because trained layers need not reach spectral norm one.

The margin $M(x)$ involves an expectation that we can only estimate, so the certificate is applied with a confidence correction.

Corollary 1 (Monte-Carlo certification)**.** *Choose $\hat{y}$ from n_0 paths. From n_c further independent paths compute the pairwise margins $Z_k^{(i)} = (e_{\hat{y}} - e_k)^\top P_B(\psi_\eta(X_n^{(i)})) \in [-\sqrt{2}B, \sqrt{2}B]$, their means $\hat{M}_k$ and one-sided confidence widths t_k at level $\alpha/(K-1)$: Hoeffding $t_k = 2B\sqrt{\log((K-1)/\alpha)/n_c}$, or empirical Bernstein (Maurer & Pontil, 2009) with sample variance $\hat{V}_k$, $t_k = \sqrt{2\hat{V}_k \log(2(K-1)/\alpha)/n_c} + \frac{14\sqrt{2}B\log(2(K-1)/\alpha)}{3(n_c-1)}$. With probability at least $1 - \alpha$ over the paths, if $\underline{M} = \min_{k \neq \hat{y}}(\hat{M}_k - t_k) > 0$, then $\hat{y} = \arg\max F^B(x)$ and $\hat{r}(x) = \underline{M}/(\sqrt{2}L) \leq r(x)$.*

Choosing the sample size. The empirical-Bernstein width has a variance term that decays like $n_c^{-1/2}$ and a range term that decays like n_c^{-1}. For $B = 10$, $K = 34$ and $\alpha = 10^{-3}$, and a path-margin standard deviation of 0.5, the width is about 1.5 logits at $n_c = 512$ and about 0.2 at $n_c = 4096$; Hoeffding's width at 4096 is 1.0. Algorithm 1 therefore uses $n_0 = 64$, $n_c = 4096$, $\alpha = 10^{-3}$ and empirical Bernstein. This is separate from the $N_{mc} = 8$ paths used for fast inference.

The second result concerns one random path rather than the average.

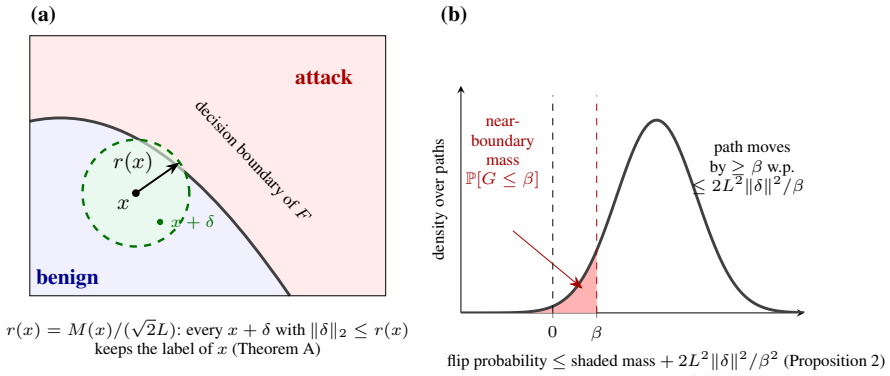


Figure 2: Schematic of the two guarantees (not data). (a) Theorem A: a margin $M(x)$ of the averaged logits certifies every input within ℓ_2 distance $r(x) = M(x)/(\sqrt{2}L)$. (b) Proposition 2: across random paths the clean margin $G(x)$ has a distribution; a single path can flip only if it starts within β of the boundary (shaded) or if the perturbation moves it by at least β.

Proposition 2 (Path-wise decision-flip bound)**.** *Fix x, a class $k^\star$ (for example $\hat{y}(x)$), and a perturbation δ chosen independently of the noise. Let $G(x) = \psi_\eta^{(k^\star)}(X_n) - \max_{k \neq k^\star} \psi_\eta^{(k)}(X_n)$ be the margin of one path and let the perturbed path use the same increments. Under Assumption 1, for every $\beta > 0$,*

$$\mathbb{P}\big[\arg\max_k \psi_\eta^{(k)}(Y_n) \neq k^\star\big] \;\le\; \mathbb{P}\big[G(x) \le \beta\big] + \frac{2L^2\|\delta\|_2^2}{\beta^2}.$$

Reading the bound (Figure 2b). A single path can change its decision in only two ways: it was already close to the boundary (within β), or the perturbation moved it by at least β. The first probability involves the clean input only; we bound it with a Clopper–Pearson upper confidence limit, which is exact for binomial counts, from fresh paths, with a union bound over a small grid of β. The second follows from Lemma 1 and Markov's inequality. Because G is a minimum over classes, no union over the 33 competing classes is needed. The first term is exactly the near-boundary mass that $\mathcal{L}_{\mathrm{AC}}$ penalises during training.

Remark 1 (What is and is not certified)**.** Theorem A is worst-case: it covers *every* perturbation within $r(x)$, for the averaged predictor F^B. With Corollary 1 it holds with probability $1-\alpha$ over the certification paths. It does not cover the fast 8-path classifier, whose agreement with F^B can be measured on the same paths. Proposition 2 is probabilistic over the path and holds for each fixed δ, including one chosen by an adversary who does not see the path, but not for an adversary who adapts δ to the realised noise.

Remark 2 (Relation to smoothing and Lipschitz networks)**.** Randomised smoothing adds Gaussian noise to the *input* of a deterministic classifier and certifies radius $\sigma\Phi^{-1}(p_A)$ from the probability p_A of the top class; the radius does not depend on the classifier's Lipschitz constant. SODE-Guard keeps the input clean, and its radius is a Lipschitz radius whose constant contains the SDE growth factor $\exp((K_f + K_g^2/2)T)$ instead of a product over a deep feed-forward stack. Because the diffusion enlarges this factor, noise in the dynamics costs certified radius. The diffusion's benefit is the empirical robustness measured below, not a larger worst-case radius. A Carbery–Wright bound on the near-boundary term of Proposition 2 is possible in principle but is vacuous at useful β (Section D), so we estimate that term directly.

6 Experiments

Data. ICS3D (Anonymous, 2025a) contains 1.89×10^7 records of cloud-security logs, an Edge-IIoT split and container traces. IIS3D (Anonymous, 2025b) contains 1.34×10^7 flows from CSE-CIC-IDS2018, CIC-IoT-2023 and UNSW-NB15, harmonised onto one 83-feature schema and 34 classes. IDS-PQC (Anonymous, 2026) contains 3.10×10^6 TLS-1.3 and post-quantum handshake

Method	Clean	PGD-40	
		ϵ=0.01	ϵ=0.03
E-GraphSAGE	0.951	0.871	0.793
RTIDS	0.953	0.864	0.781
CNN-LSTM	0.946	0.842	0.738
IDS-GraphMamba	0.958	0.882	0.811
Ens-7B	0.961	0.901	0.847
SDE-TGNN	0.961	0.911	0.872
SODE-Guard (ours)	**0.964**	**0.946**	**0.931**

Table 2: Macro-F1, mean of the three per-dataset scores (Table 4); five seeds; random split; ℓ_∞ PGD-40 on z-scored features. Baselines: E-GraphSAGE (Lo et al., 2022), RTIDS (Wu et al., 2022), CNN-LSTM (Halbouni et al., 2022), IDS-GraphMamba (Mosaiyebzadeh et al., 2025).

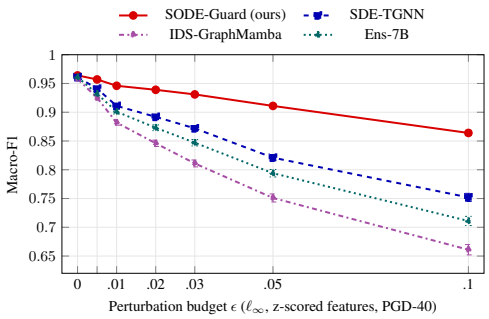


Figure 3: Macro-F1 versus ℓ_∞ budget under PGD-40 (same averaging as Table 2; error bars: bootstrap 95% intervals over five seeds).

flows. We use stratified random 70/15/15 train/validation/test splits. The repository also provides temporal, host-disjoint and scenario-disjoint splits and deduplication and leakage checks (Section F).

Baselines. We compare six neural detectors under the same threat model: E-GraphSAGE (Lo et al., 2022), RTIDS (Wu et al., 2022), CNN-LSTM (Halbouni et al., 2022), IDS-GraphMamba (Mosaiyebzadeh et al., 2025), and two strong detectors with publicly released code, a seven-branch ensemble (Ens-7B) and a temporal-graph neural SDE (SDE-TGNN), configured as in the anonymised repository. Signature and LLM-policy systems face a different threat model and are reported only for context (Section E).

Metrics and protocol. We report macro-F1 on the test split, the mean of five seeds, with bias-corrected (BCa) bootstrap 95% confidence intervals. We also report the expected calibration error (ECE): the average gap between predicted confidence and observed accuracy over 15 confidence bins, so lower means better-calibrated probabilities. McNemar's test compares two methods on the same flows by counting the flows that exactly one of them misclassifies. All ϵ are ℓ_∞ budgets in z-score units. PGD-40 against SODE-Guard differentiates through exactly the predictor that is scored, the 8-path average.

6.1 Clean and adversarial performance

SODE-Guard has the best clean macro-F1 and, at $\epsilon = 0.03$, leads SDE-TGNN by 5.9 points (Table 2); per-dataset results in Table 4 show the same order on every benchmark. McNemar's test against SDE-TGNN at $\epsilon = 0.03$ gives $p < 10^{-8}$. Figure 3 extends the comparison to seven budgets. SODE-Guard degrades most slowly and keeps 0.864 at $\epsilon = 0.1$, where SDE-TGNN keeps 0.752. The confidence bands are narrower than ± 0.01 everywhere. A curve this flat can also be a symptom of gradient masking, where the attack fails because it receives uninformative gradients rather than because the model is robust. PGD-40 results are therefore upper bounds on robust accuracy until adaptive attacks are run (Section 7).

6.2 Ablation

Table 3 removes one component at a time. Removing the diffusion ($g_\theta \to 0$, which turns the model into a neural ODE) costs 6.7 PGD-40 points but only 0.4 clean points, the largest single effect under attack. Removing spectral normalisation or the diffusion floor costs 11.9 and 5.5 points, and removing both costs 15.0. The regulariser $\mathcal{L}_{\mathrm{AC}}$ adds 3.4 points. Randomness helps empirically even though, by Lemma 1, it enlarges the worst-case constant.

Efficiency. On one A100 with batch 128 the median (P95) latency is 1.7 (3.1) ms for SODE-Guard with $N_{\mathrm{mc}} = 8$, against 1.4 (2.6) ms for SDE-TGNN and 0.8 (1.4) ms for Ens-7B. That is a single-stream throughput of about 7.5×10^4 flows/s (batch size divided by median latency). SODE-Guard

Table 3: Ablation over the three benchmarks (random split; mean of five seeds). Δ is the change in clean macro-F1 in points; ECE is expected calibration error (lower is better).

Configuration	Clean	Δ	PGD-40 (ϵ=0.03)	ECE
Full SODE-Guard	0.964	—	0.931	0.028
− diffusion ($g_\theta \to 0$, neural ODE)	0.960	-0.4	0.864	0.041
− E-GraphSAGE edge features	0.949	-1.5	0.882	0.046
− spectral normalisation on f_θ, g_θ	0.957	-0.7	0.812	0.064
− multi-path inference ($N_{\mathrm{mc}} = 1$)	0.958	-0.6	0.901	0.052
− regulariser $\mathcal{L}_{\mathrm{AC}}$	0.962	-0.2	0.897	0.034
− diffusion floor λ_0	0.958	-0.6	0.876	0.051
− cosine annealing	0.961	-0.3	0.919	0.032
− virtual-Brownian-tree cache	0.964	± 0	0.926	0.029
− spectral norm and λ_0 together	0.951	-1.3	0.781	0.087

also has the lowest ECE (0.028; Table 6). Certification with $n_{\mathrm{c}} = 4096$ paths is an offline audit and does not run at line rate.

7 Discussion and limitations

What the theory explains. Theorem A guarantees stability of the averaged predictor. Its radius is small in ℓ_∞ units because it divides by $\sqrt{83}$ and because the constant contains the SDE growth factor. It explains why the averaged model *can* be certified, not why the model resists PGD as well as it does, and it has not yet been applied to the trained models of Section 6. Proposition 2 links training ($\mathcal{L}_{\mathrm{AC}}$ reduces the near-boundary mass) to single-path stability, but holds only for perturbations chosen without seeing the noise.

Adaptive attacks. Randomised defences can look robust because of gradient masking (Athalye et al., 2018). Our PGD-40 attacks differentiate through the scored 8-path average, but they do not replace large-sample EOT, AutoAttack-style ensembles, a direct attack on the mean predictor and the checks of Carlini et al. (2019). The repository implements all of them; until they are reported, PGD-40 numbers are upper bounds on robust accuracy.

Feature space versus traffic. Attacks and certificates act on extracted statistics. Some changes cannot be produced by real packets (e.g. a negative byte count), and real traffic changes move many statistics at once; the repository's feasibility projector restricts attacks to realisable feature vectors, and packet-level evaluation is future work.

Data and baselines. IIS3D harmonises three public corpora onto one schema, which can make classification easier than in deployment; temporal and host-disjoint splits and per-corpus results would strengthen external validity. Two of the six neural baselines were configured by us. **Numerics.** On the encrypted-flow tail of IDS-PQC we observed occasional instability that required gradient clipping at unit norm.

8 Conclusion

We studied a neural SDE classifier for intrusion detection that averages a linear read-out over random latent trajectories. We proved a worst-case certified radius for the averaged predictor, with a constant computed from the trained weights and a Monte-Carlo confidence correction, and a class-union-free bound on single-path decision flips whose near-boundary term our regulariser targets. The analysis shows that noise in the dynamics enlarges the worst-case constant, while empirically the same noise buys robustness: SODE-Guard improves PGD-40 macro-F1 over the strongest neural baseline by 5.9 points at $\epsilon = 0.03$. Future work covers certifying trained models, adaptive attacks, path-adaptive adversaries, tighter constants, packet-level attacks and model-extraction detection.

REPRODUCIBILITY STATEMENT

Architectures, hyper-parameters and seeds are in Section 4 and Algorithm 2; all assumptions are stated (Assumption 1); complete proofs are in Section C. The anonymised repository (`https://anonymous.4open.science/r/SODE-ExtractGuard-Models`) contains the model, the certifier (Algorithm 1, Proposition 2), the regularised BEL estimator, all attacks, the split and deduplication protocols, baseline configurations, the sources of every figure, and one script per experiment that writes the corresponding table rows. Dataset download pointers are in the repository; links are withheld for review.

ETHICS STATEMENT

This work is defensive and aims to make intrusion detectors harder to evade. The attacks used are standard and public. The datasets contain flow statistics only, with no payloads or personal identifiers. A certificate for the averaged predictor does not certify the fast deployed classifier, and Proposition 2 does not cover path-adaptive adversaries; the detector should not be the sole safeguard.

AI USE STATEMENT

We used large-language-model assistants to reformat the manuscript to the ICLR template and copy-edit it; to also revise parts of the theoretical exposition and of the proofs in Sections 2, 4 and 5 and Section C, after initial draft by the authors, which the authors also checked line by line. The large-language-model was not used in writing any other parts of the manuscript, the released code, including the certifier and the adaptive attacks, which the authors tested with the unit tests in the repository. All experimental results were produced by the authors' own training and evaluation runs. Every citation was checked against its source. We take responsibility for the content of this work.

REFERENCES

Roger Nick Anaedevha and Alexander G. Trofimov. Stochastic multimodal transformer with uncertainty quantification for robust network intrusion detection. In *Advances in Neural Computation, Machine Learning, and Cognitive Research IX*, volume 1241 of *Studies in Computational Intelligence*. Springer, 2025. doi: 10.1007/978-3-032-07690-8_35.

Roger Nick Anaedevha, Alexander G. Trofimov, and Yuri V. Borodachev. Temporal adaptive neural ordinary differential equations with deep spatio-temporal point processes for real-time network intrusion detection. *Complex & Intelligent Systems*, 2026. doi: 10.1007/s40747-026-02291-7.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *Proc. ECCV*, pp. 484–501, 2020.

Cem Anil, James Lucas, and Roger Grosse. Sorting out Lipschitz function approximation. In *Proc. ICML*, volume 97 of *PMLR*, pp. 291–301, 2019.

Anonymous. ICS3D: Integrated cloud security flow benchmark, 2025a. Link withheld for review; data available via the anonymised repository.

Anonymous. IIS3D: Integrated intrusion-detection flow benchmark, 2025b. Link withheld for review; data available via the anonymised repository.

Anonymous. IDS-PQC: Encrypted and post-quantum handshake flow benchmark, 2026. Link withheld for review; data available via the anonymised repository.

Alexandre Araujo, Aaron Havens, Blaise Delattre, Alexandre Allauzen, and Bin Hu. A unified algebraic perspective on Lipschitz neural networks. In *Proc. ICLR*, 2023.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proc. ICML*, volume 80 of *PMLR*, pp. 274–283, 2018.

Richard Bergna et al. Latent graph neural stochastic differential equations. *arXiv preprint arXiv:2408.16115*, 2024.

Jean-Michel Bismut. Martingales, the Malliavin calculus and hypoellipticity under general Hörmander's conditions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 56(4): 469–505, 1981. doi: 10.1007/BF00531428.

Anthony Carbery and James Wright. Distributional and L^q norm inequalities for polynomials over convex bodies in $\mathbb{R}^n$. *Mathematical Research Letters*, 8(3):233–248, 2001. doi: 10.4310/MRL.2001.v8.n3.a1.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Jeremy Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In *Proc. ICML*, volume 97 of *PMLR*, pp. 1310–1320, 2019.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proc. ICML*, volume 119 of *PMLR*, pp. 2206–2216, 2020.

Francisco Eiras, Motasem Alfarra, M. Pawan Kumar, Philip H. S. Torr, Puneet K. Dokania, Bernard Ghanem, and Adel Bibi. ANCER: Anisotropic certification via sample-wise volume maximization. *Transactions on Machine Learning Research*, 2022.

K. David Elworthy and Xue-Mei Li. Formulae for the derivatives of heat semigroups. *Journal of Functional Analysis*, 125(1):252–286, 1994. doi: 10.1006/jfan.1994.1124.

Sabrine Ennaji et al. Adversarial challenges in network intrusion detection systems: Research insights and future prospects. *arXiv preprint arXiv:2409.18736*, 2024.

Nicola Guglielmi et al. Minimal weight perturbation for robust neural ODEs. *arXiv preprint arXiv:2501.10740*, 2025.

Asmaa Halbouni, Teddy Surya Gunawan, Mohamed Hadi Habaebi, Murad Halbouni, Mira Kartiwi, and Robiah Ahmad. CNN-LSTM: Hybrid deep neural network for network intrusion detection system. *IEEE Access*, 10:99837–99849, 2022. doi: 10.1109/ACCESS.2022.3206425.

Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *Proc. CVPR*, pp. 588–597, 2019. doi: 10.1109/CVPR.2019.00068.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Elton P. Hsu. *Stochastic Analysis on Manifolds*, volume 38 of *Graduate Studies in Mathematics*. American Mathematical Society, 2002.

Qiyu Kang, Yang Song, Qinxu Ding, and Wee Peng Tay. Stable neural ODE with Lyapunov-stable equilibrium points for defending against adversarial attacks. In *Proc. NeurIPS*, 2021.

Patrick Kidger, James Foster, Xuechen Li, Harald Oberhauser, and Terry Lyons. Neural SDEs as infinite-dimensional GANs. In *Proc. ICML*, volume 139 of *PMLR*, pp. 5453–5463, 2021.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *Proc. IEEE Symposium on Security and Privacy*, pp. 656–672, 2019. doi: 10.1109/SP.2019.00044.

Xuechen Li, Ting-Kam Leonard Wong, Ricky T. Q. Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *Proc. AISTATS*, volume 108 of *PMLR*, pp. 3870–3882, 2020.

Xuanqing Liu, Yao Li, Chongruo Wu, and Cho-Jui Hsieh. Adv-BNN: Improved adversarial defense through robust Bayesian neural network. In *Proc. ICLR*, 2019.

Llama Team, AI @ Meta. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Wai Weng Lo, Siamak Layeghy, Mohanad Sarhan, Marcus Gallagher, and Marius Portmann. E-GraphSAGE: A graph neural network based intrusion detection system for IoT. In *Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS)*, pp. 1–9, 2022. doi: 10.1109/NOMS54207.2022.9789878.

David J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. ICLR*, 2018.

Andreas Maurer and Massimiliano Pontil. Empirical Bernstein bounds and sample variance penalization. In *Proc. COLT*, 2009.

Raghu Meka, Oanh Nguyen, and Van Vu. Anti-concentration for polynomials of independent random variables. *Theory of Computing*, 12(11):1–17, 2016. doi: 10.4086/toc.2016.v012a011.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Proc. ICLR*, 2018.

Fatemeh Mosaiyebzadeh et al. IDS-GraphMamba: A Markov-enhanced graph Mamba framework for real-time intrusion detection in IoMT edge networks. *Computer Networks*, 268:108989, 2025. doi: 10.1016/j.comnet.2025.108989.

David Nualart. *The Malliavin Calculus and Related Topics*. Springer, 2nd edition, 2006. doi: 10.1007/3-540-28329-3.

YongKyung Oh, Dongyoung Lim, and Sungil Kim. Stable neural stochastic differential equations in analyzing irregular time series data. In *Proc. ICLR*, 2024.

Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer, 6th edition, 2003. doi: 10.1007/978-3-642-14394-6.

Open Information Security Foundation. Suricata 7.0 release notes. `https://suricata.io/release-notes-7-0/`, 2024.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sébastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Proc. NeurIPS*, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proc. ICLR*, 2021.

The Snort Project. Snort 3 with SnortML neural exploit-detection engine. `https://github.com/snort3/snort3`, 2024.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In *Proc. NeurIPS*, 2018.

Belinda Tzen and Maxim Raginsky. Neural stochastic differential equations: Deep latent Gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*, 2019.

Andrea Venturi, Dario Stabili, and Mirco Marchetti. Problem-space structural adversarial attacks for network intrusion detection systems based on graph neural networks. *arXiv preprint arXiv:2403.11830*, 2024.

João Vitorino et al. An adversarial robustness benchmark for enterprise network intrusion detection. *arXiv preprint arXiv:2402.16912*, 2024.

Ruigang Wang and Ian R. Manchester. Direct parameterization of Lipschitz-bounded deep networks. In *Proc. ICML*, 2023.

Zihan Wu, Hong Zhang, Penghai Wang, and Zhibo Sun. RTIDS: A robust transformer-based approach for intrusion detection system. *IEEE Access*, 10:64375–64387, 2022. doi: 10.1109/ACCESS.2022.3182333.

Greg Yang, Tony Duan, J. Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *Proc. ICML*, volume 119 of *PMLR*, pp. 10693–10705, 2020.

A GLOSSARY

Term	Meaning in this paper
Flow	packets exchanged between two endpoints in one session, summarised by 83 statistics
IDPS / NIDS	(network) intrusion detection and prevention system: classifies flows as benign or as an attack type
Logit, softmax, margin	raw class score; normalisation of scores into probabilities; gap between the two largest logits
Macro-F1	F1 (harmonic mean of precision and recall) per class, averaged with equal weight over classes
ECE	expected calibration error: mean gap between predicted confidence and accuracy over confidence bins
Adversarial perturbation	a small, deliberately chosen change δ to the input that aims to change the decision
ℓ_2, ℓ_∞ budget	bound on the Euclidean length, or on the largest single coordinate, of δ
PGD, EOT	projected gradient descent attack; expectation over transformation, i.e. averaging gradients over the model's randomness
APGD, Square	step-size-adaptive PGD (Croce & Hein, 2020); a query-based attack that needs no gradients (Andriushchenko et al., 2020)
Gradient masking	an attack fails because gradients are uninformative, not because the model is robust
Certified radius	distance within which the decision provably cannot change (Definition 1)
Lipschitz constant	largest ratio of output change to input change
Spectral norm / normalisation	largest singular value of a matrix; rescaling weights so that it is at most one
Brownian motion	continuous-time accumulated Gaussian noise with independent increments
Drift, diffusion	deterministic and random parts of an SDE (Equation (1))
Euler–Maruyama	step-by-step simulation of an SDE (Equation (2))
Path	one simulated trajectory, driven by one draw of the noise
Synchronous coupling	comparing two inputs by driving both with the same noise
Monte-Carlo estimate	average over independent random paths
Hoeffding, empirical Bernstein	confidence bounds for averages of bounded variables (the second uses the observed variance)
Clopper–Pearson	exact confidence bound for a probability estimated from binomial counts
Bismut–Elworthy–Li	a formula for gradients of expectations of SDEs that avoids differentiating the function (Proposition 1)
Anti-concentration	a bound on how much probability a random quantity can place near a value, here near the decision boundary
Randomised smoothing	certification by majority vote over Gaussian-noised copies of the input

B TRAINING ALGORITHM

Algorithm 2 SODE-Guard training

Require: $\mathcal{D}_{\mathrm{tr}}$, $T = 1$, $\Delta t = 0.05$, paths P, floor λ_0, weight $\lambda = 0.1$

Initialise $h, \theta_f, \theta_g, \eta$ with spectral normalisation on every linear layer
for epoch $= 1, \ldots, 40$ **do**
  for mini-batch $\{(x_i, y_i)\}_{i=1}^{B}$ **do**
    $X_0^{(i)} = h(x_i)$; draw P Brownian paths (virtual Brownian tree)
    Integrate (2) with diffusion floor λ_0; average the logits over paths
    $\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda \mathcal{L}_{\mathrm{AC}}$ with (4)
    Backpropagate through (2) (path-wise gradients for all parameters)
    Adam step with cosine annealing
  end for
end for
After training: recompute every $\|W_\ell\|$ by exact SVD and form L of Theorem A

C PROOFS

Proof of Lemma 1. Let $D_k = Y_k - X_k$. The constant diffusion floor cancels in the difference, so $D_{k+1} = D_k + \Delta f_k\,\Delta t + \Delta g_k\,\Delta W_k$ with $\Delta f_k = f_\theta(Y_k, t_k) - f_\theta(X_k, t_k)$ and $\Delta g_k = g_\theta(Y_k, t_k) - g_\theta(X_k, t_k)$. Since ΔW_k is independent of (X_k, Y_k) with mean zero and covariance $\Delta t\, I_m$, the cross term vanishes and

$$\mathbb{E}\big[\|D_{k+1}\|^2 \mid X_k, Y_k\big] = \|D_k + \Delta f_k \Delta t\|^2 + \Delta t\,\|\Delta g_k\|_F^2 \le \big((1 + K_f\Delta t)^2 + K_g^2\Delta t\big)\|D_k\|^2.$$

Taking expectations and iterating $n = T/\Delta t$ times, with $1 + u \le e^u$, gives $\mathbb{E}\|D_n\|^2 \le \|D_0\|^2 \exp((2K_f + K_g^2 + K_f^2\Delta t)T)$, and $\|D_0\| \le L_h\|\delta\|$. As $\Delta t \to 0$ this recovers the continuous-time Grönwall bound $\exp((2K_f + K_g^2)T)$. □

Proof of Theorem A. Couple the two expectations through the same increments. P_B is non-expansive (it is a projection onto a convex set), and ψ_η is affine with $\|W_\psi\|$. By Jensen's inequality, Cauchy–Schwarz and Lemma 1,

$$\|F^B(x+\delta) - F^B(x)\| \le \mathbb{E}\|P_B(\psi_\eta(Y_n)) - P_B(\psi_\eta(X_n))\| \le \|W_\psi\|\big(\mathbb{E}\|Y_n - X_n\|^2\big)^{1/2} \le L\|\delta\|.$$

For every $k \ne \hat{y}$, $|(F^B_{\hat{y}} - F^B_k)(x+\delta) - (F^B_{\hat{y}} - F^B_k)(x)| = |(e_{\hat{y}} - e_k)^\top(F^B(x+\delta) - F^B(x))| \le \sqrt{2}L\|\delta\| < M(x) \le (F^B_{\hat{y}} - F^B_k)(x)$, so $(F^B_{\hat{y}} - F^B_k)(x+\delta) > 0$. The ℓ_∞ statement follows from $\|\delta\|_2 \le \sqrt{83}\|\delta\|_\infty$. For the numerical bound: each perceptron has three linear layers and two activations, so $K_f, K_g \le c_\phi^2$; the encoder gives $L_h \le c_\phi(\frac{1}{2}c_\phi + \frac{1}{2})^3$; and $\|W_\psi\| \le 1$. With $c_\phi = 1.129$ and $\Delta t = 0.05$ this gives $L \le 1.36 \times e^{2.13} \approx 11.4$. □

Proof of Corollary 1. The n_c estimation paths are independent of the selection paths, so $\hat{y}$ is fixed given the selection sample. For each $k \ne \hat{y}$ the $Z_k^{(i)}$ are independent and identically distributed with range $2\sqrt{2}B$. The one-sided Hoeffding inequality, or Theorem 4 of Maurer & Pontil (2009), gives $\mathbb{E}Z_k \ge \hat{M}_k - t_k$ with probability at least $1 - \alpha/(K-1)$. A union bound over the $K-1$ competitors gives $M(x) \ge \underline{M}$ with probability at least $1 - \alpha$. If $\underline{M} > 0$, then $\hat{y}$ is the maximiser of F^B, and Theorem A applies with $M(x) \ge \underline{M}$. □

Proof of Proposition 2. The perturbed path's decision differs from $k^\star$ only if $G(x+\delta) \le 0$. For vectors a, b, $|\min_k a_k - \min_k b_k| \le \max_k |a_k - b_k|$, and $\|(e_{k^\star} - e_k)^\top W_\psi\| \le \sqrt{2}\|W_\psi\|$, so $|G(x+\delta) - G(x)| \le \sqrt{2}\|W_\psi\|\,\|Y_n - X_n\|$. Hence $\{G(x+\delta) \le 0\} \subseteq \{G(x) \le \beta\} \cup \{\sqrt{2}\|W_\psi\|\|Y_n - X_n\| \ge \beta\}$. By Markov's inequality on the square and Lemma 1, the second event has probability at most $2\|W_\psi\|^2\mathbb{E}\|Y_n - X_n\|^2/\beta^2 \le 2L^2\|\delta\|^2/\beta^2$. □

Proof of Proposition 1. Part (i) is the Elworthy–Li formula (Elworthy & Li, 1994). For (ii), $u_s = \frac{1}{t}g_\lambda^+(X_s)J_s v$ is adapted, and the Malliavin derivative of X_t in direction u is $D_u X_t = J_t \int_0^t J_s^{-1} g(X_s) u_s\, ds$ (Nualart, 2006, Sec. 2.2). Integration by parts gives $\mathbb{E}[\phi(X_t)\int_0^t \langle u_s, dW_s\rangle] = \mathbb{E}[\langle \nabla\phi(X_t), D_u X_t\rangle]$, and $g\, g_\lambda^+ = \Pi_\lambda$, which gives the stated identity. If $\Pi_\lambda(X_s)J_s v = J_s v$ for all s, then $D_u X_t = J_t v$ and the right-hand side is $\partial_v \mathbb{E}\phi(X_t)$. As $\lambda \to 0$, Π_λ converges to the orthogonal projection onto $\operatorname{range} g(X_s)$. □

The implementation propagates $J_k v$ through the EM map by forward-mode differentiation and uses the discrete weight $\frac{1}{T}\sum_k \langle g_\lambda^+(X_k)J_k v, \Delta W_k\rangle$. For a linear SDE with $m = d$ it reproduces the analytic gradient within Monte-Carlo error, which a unit test checks.

D CARBERY–WRIGHT AND THE NEAR-BOUNDARY TERM

For a polynomial p of degree at most d in standard Gaussians, $\mathbb{P}[|p(Z)| \le \varepsilon\|p\|_{L^2}] \le Cd\,\varepsilon^{1/d}$ (Carbery & Wright, 2001; Meka et al., 2016). On the EM grid the path margin $G(x)$ is a function of $mn = 320$ Gaussians, and it is a polynomial only after chaos truncation. Using the inequality for $\mathbb{P}[|G(x)| \le \beta]$ in Proposition 2 therefore needs (a) a truncation degree and a bound on the truncation residual, and (b) the constant C. Even with $C = 1$ and $d = 4$, the bound is below one

only when $\beta < 0.004\,\|G(x)\|_{L^2}$. At such small β, the movement term $2L^2\|\delta\|^2/\beta^2$ of Proposition 2 is large unless $\|\delta\|$ is tiny. A total-degree Hermite regression in 320 variables is also infeasible (about 4.5×10^8 terms at degree 4). We therefore estimate $\mathbb{P}[G(x) \le \beta]$ with a Clopper–Pearson bound, and the regulariser (4) keeps the Carbery–Wright-shaped weight only as a training heuristic.

E Additional tables

Table 4: Per-dataset macro-F1 (random split; five seeds; BCa 95% CIs for SODE-Guard; PGD-40 at $\epsilon = 0.03$).

	ICS3D		IIS3D		IDS-PQC	
Method	Clean	PGD-40	Clean	PGD-40	Clean	PGD-40
E-GraphSAGE	0.949	0.771	0.955	0.808	0.948	0.799
RTIDS	0.947	0.754	0.961	0.797	0.950	0.791
CNN-LSTM	0.942	0.714	0.953	0.755	0.943	0.744
IDS-GraphMamba	0.952	0.793	0.963	0.824	0.958	0.817
Ens-7B	0.957	0.831	0.968	0.862	0.957	0.849
SDE-TGNN	0.957	0.858	0.967	0.886	0.958	0.872
SODE-Guard	0.961	0.922	0.969	0.938	0.962	0.933
95% CI	[.957,.965]	[.916,.928]	[.964,.973]	[.932,.943]	[.958,.966]	[.927,.939]

Table 5: Reference systems with a different (signature/regex or text-policy) threat model; macro-F1, random split. A gradient-based perturbation is not the natural attack on these systems, so this table gives production context only. Llama Guard 3 is adapted to flows by policy rewriting.

Method	Clean	PGD-40	
		ϵ=0.01	ϵ=0.03
Snort 3 + SnortML (The Snort Project, 2024)	0.812	0.745	0.660
Suricata 7 (Open Information Security Foundation, 2024)	0.836	0.778	0.694
Llama Guard 3 (Llama Team, AI @ Meta, 2024)	0.792	0.728	0.641

Table 6: Efficiency and calibration (A100 80 GB, batch 128). Throughput is the single-stream figure: batch size divided by median latency.

Method	Latency P50	Latency P95	ECE	Throughput (flows/s)
SDE-TGNN	1.4 ms	2.6 ms	0.038	9.1×10^4
IDS-GraphMamba	2.1 ms	3.8 ms	0.054	6.1×10^4
Ens-7B	0.8 ms	1.4 ms	0.031	1.6×10^5
SODE-Guard	1.7 ms	3.1 ms	0.028	7.5×10^4

F Datasets

ICS3D combines cloud-security logs, an Edge-IIoT split and Kubernetes/Docker container traces. IIS3D maps CSE-CIC-IDS2018, CIC-IoT-2023 and UNSW-NB15 onto a common 83-feature schema and a harmonised 34-class label space. IDS-PQC contains TLS-1.3 and post-quantum-cryptography handshake traffic with downgrade and key-encapsulation replay attacks. Features are extracted by a CICFlowMeter-compatible pipeline; missing values are median-imputed and all features are z-scored with training-split statistics. The anonymised repository contains the preprocessing code, the split protocols (random, temporal, host-disjoint and scenario-disjoint), a BLAKE2b-fingerprint deduplication and cross-split leakage report, and download pointers for the three corpora.