

Fiabilité du Deep Learning avec des algorithmes de simulations d'événements rares: théorie et pratique.

Karim TIT

Soutenance de thèse, à l'IRISA, Rennes, le 22 avril 2024

Directeurs: Teddy Furon (INRIA, D.R.) & Mathias Rousset (INRIA, C.R.)

Membres du jury:

Bruno Tuffin (INRIA, D.R.), président du jury

Agnès Lagnoux (Université Toulouse Jean Jaurès, M.C.)

Arnaud Guyader (Sorbonne Université, Pr.)

Stéphane Gerchinovitz, (IRT Saint-Exupéry, C.R.)

Rapporteurs de thèse:

Jean-Marc Bourinet (Sigma Clermont, Pr.)

Jérôme Morio (ONERA, D.R.)



- **Titre** : Fiabilité du Deep Learning avec des algorithmes de simulations d'événements rares: théorie et pratique.
- **Contrat** : CIFRE-Défense, co-financé par l'AID (Agence Innovation de Défense)
- **Dates** : du 1^{er} Février 2021 au 31 Janvier 2024
- **Laboratoire associé** : INRIA Rennes / IRISA
- **Encadrants Académiques**: Teddy Furon (équipe Linkmedia) et Mathias Rousset (équipe Simsmart)
- **Encadrant THALES** : Louis-Marie Traonouez (IAS - La Ruche)

Table des matières

1. Motivation
2. État de l'art en matière de robustesse de l'apprentissage profond
3. Énoncé du problème et résumé des contributions
4. 1^{ère} Contribution: ""
5. 2^{nde} Contribution: "SMC informée par le gradient pour l'estimation de la robustesse"
6. 3^{ème} Contribution: "Échantillonnage d'importance piloté par une attaque adverse"
7. Conclusion

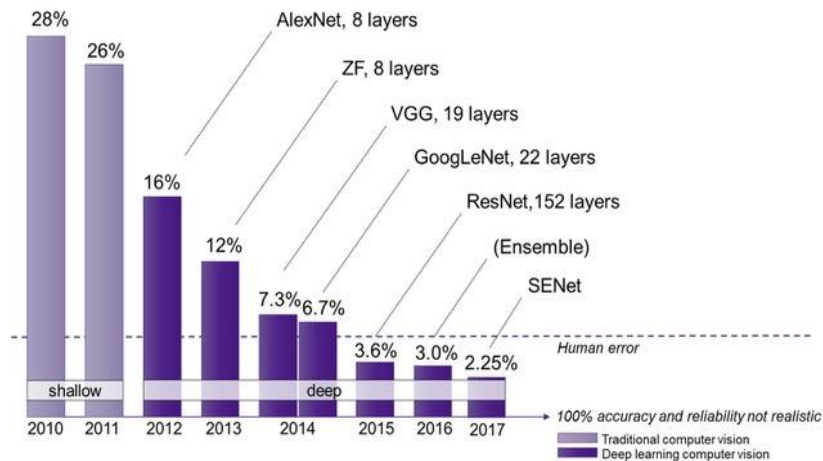


1. Motivation

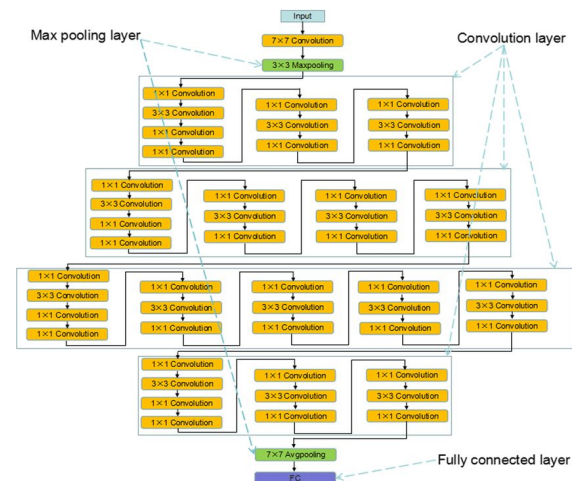


L'apprentissage profond

- Une technique puissante pour le traitement du signal (classification, détection et génération d'images, de signaux sonores, de textes...)
- Des architectures de plus en plus complexes et opaques (GPT-3: 175 milliards de paramètres...)



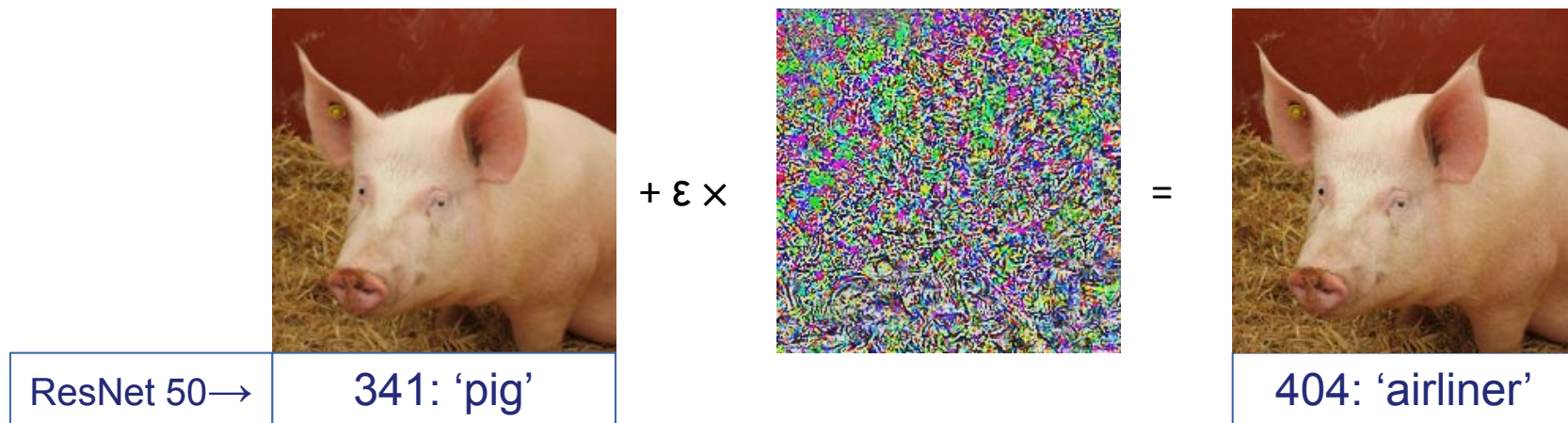
Compétition de classification ImageNet



Architecture ResNet50

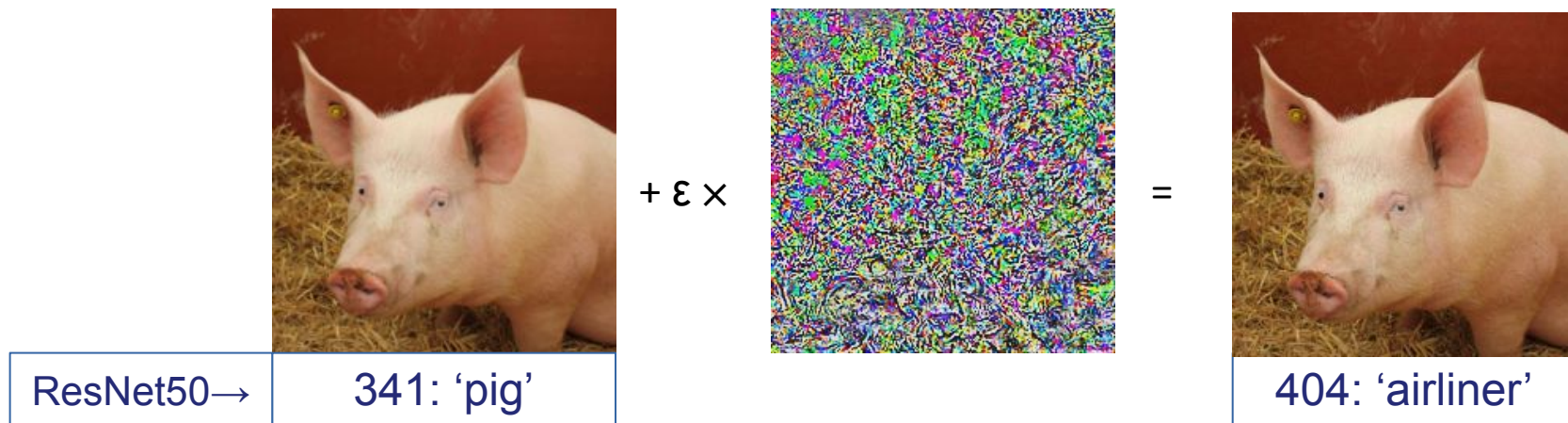
■ Précision élevée \nRightarrow Fiabilité ou sécurité

En 2014, Szegedy, Goodfellow et al. (plus de 3500 citations), montrent la fragilité des DNN, illustrée ci-dessous pour l'architecture ResNet50.



When pigs can fly : les attaques ciblées

- Avec assez d'informations sur les paramètres/l'architecture du réseau de neurones ou en utilisant des méthodes sophistiquées: possibilité d'attaques ciblées



Généralisation \neq Fiabilité \neq Sécurité

- Généralisation : fonctionner comme prévu sur des données **propres** inédites
- Reliability: to operate as expected noisy data
- Security: to operate as expected on data purposely perturbed by attackers

Robustesse aux incertitudes

- Les réseaux de neurones sont aussi vulnérables aux perturbations aléatoires



True image: cauliflower



Adv image: artichoke

VGG16 leuré par un bruit uniform aléatoire

Robustness of classifiers: from adversarial to random noise, Fawzi et al., NeurIPS 2016

Generalization

original



Prediction "Lawn_mower"
Distortion 0

Reliability

noise



"projector"
73.2

JPEG



"joystick"
14.5

Security

black-box



"vacuum"
4.5

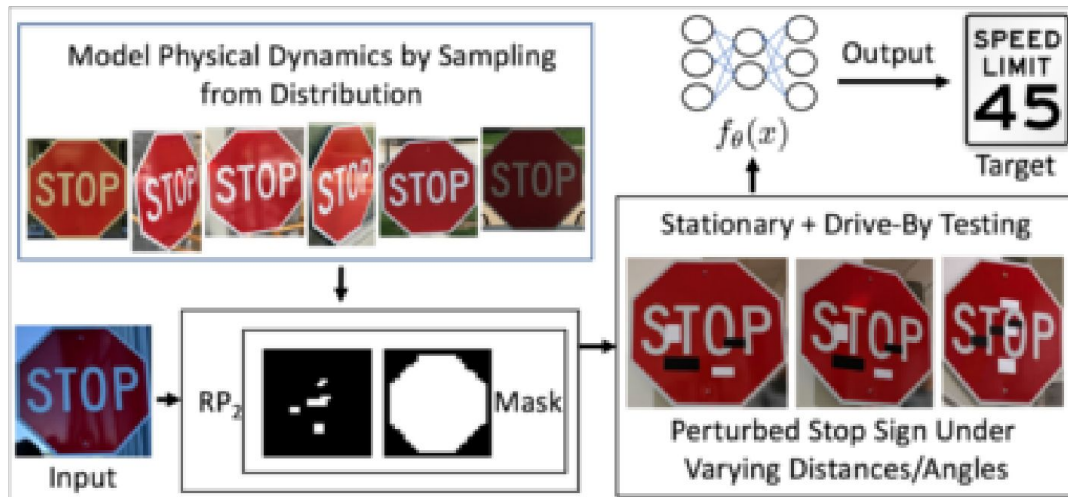
white-box



"rifle"
0.14

Pourquoi est-ce important ?

- Déploiement des méthodes de 'Deep Learning' dans le monde physique
- Aide à la décision dans des domaines critiques (défense, santé, véhicules autonomes...)



Pourquoi est-ce important ?

Man crushed to death by robot in South Korea

3 days ago



By Emily Atkinson

BBC News

A man has been crushed to death by a robot in South Korea after it failed to differentiate him from the boxes of food it was handling, reports say.

OPEN

■ Pourquoi est-ce important ?

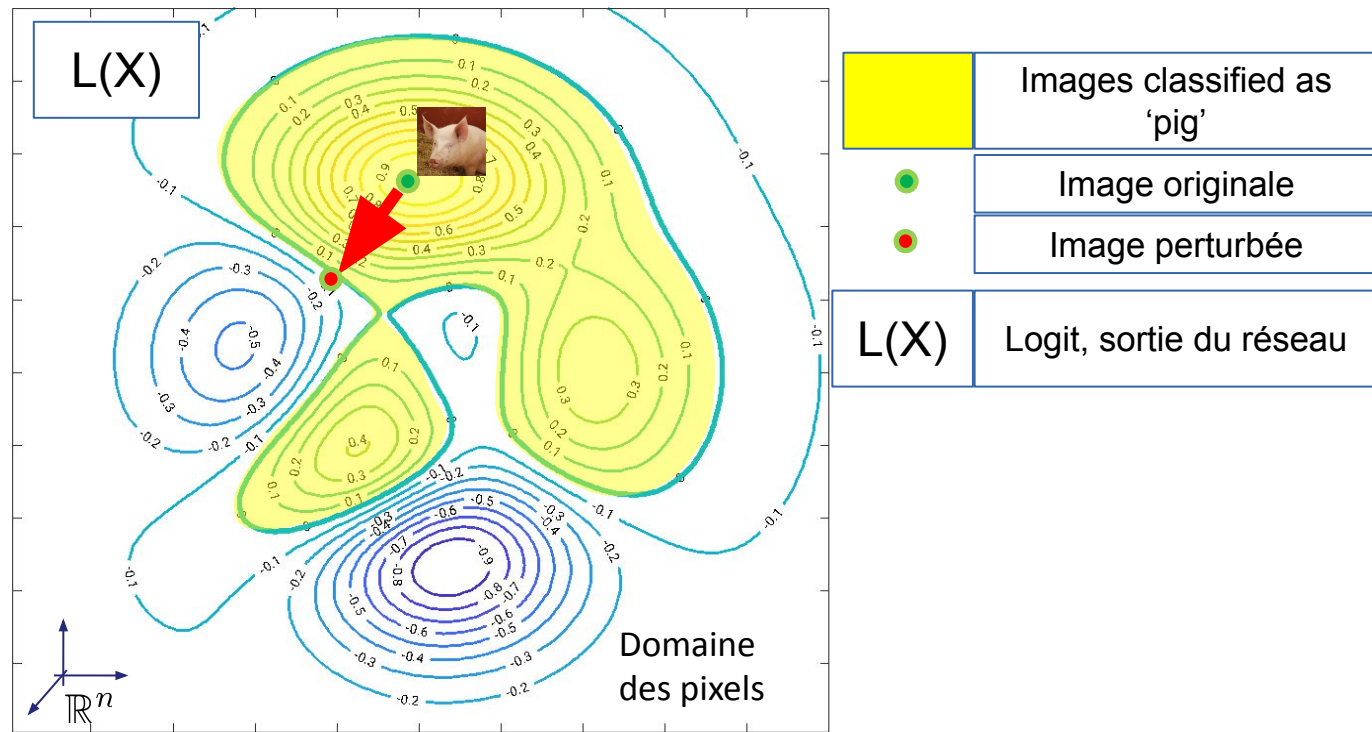
- Déploiement de systèmes de Deep Learning dans le "monde réel"
- Une quantification précise de la robustesse des systèmes d'IA est indispensable pour que l'IA soit digne de confiance
- Les réglementations internationales et les lignes directrices en matière d'audit s'appuieront sur la certification de l'IA.



2. État de l'art en robustesse de l'apprentissage profond



Attaques de type 'descente de gradient'

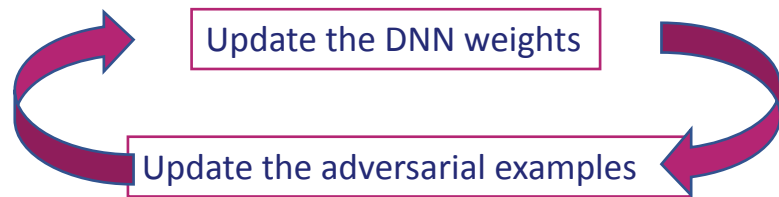


Adversarial training

Idea:

- DNN is not trained for the worst
- Include adversarial examples in the training set

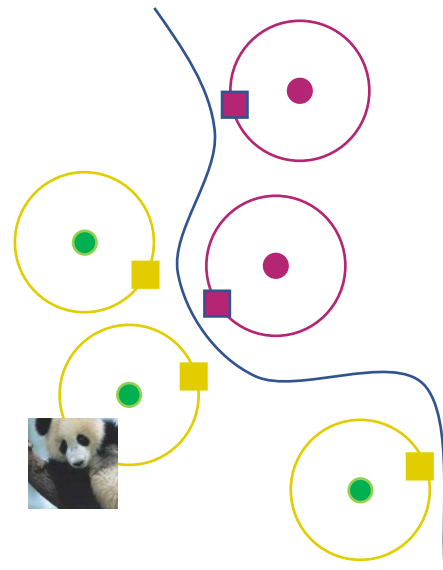
Difficulty



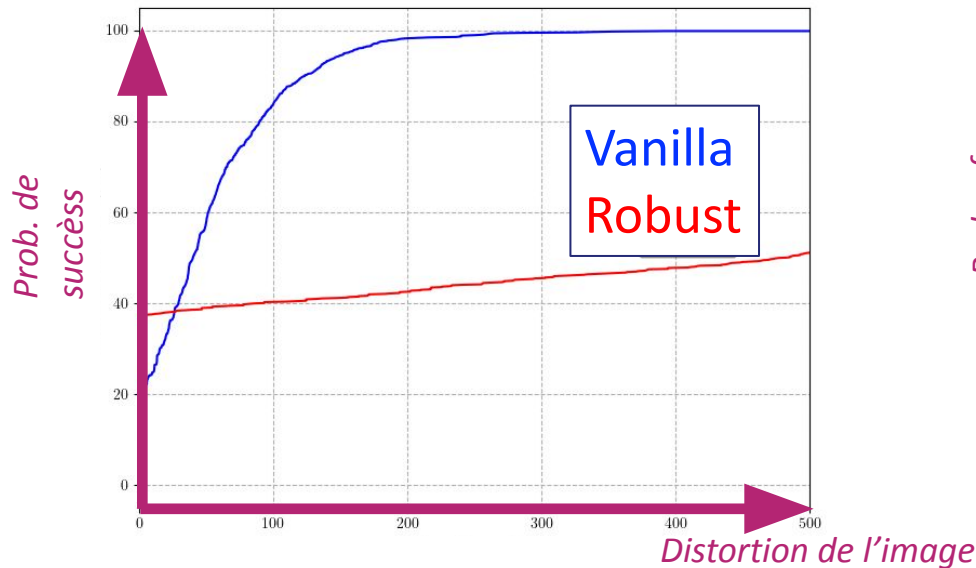
- Only possible with a simple attack like FGSM
- The DNN is not prepared for the **worst** attack
- No theoretical **guarantees, even against FGSM**

Plenty of tricks

- Gradually increase the number / strength of adversarial examples
- Larger training set
- Specific batch normalization

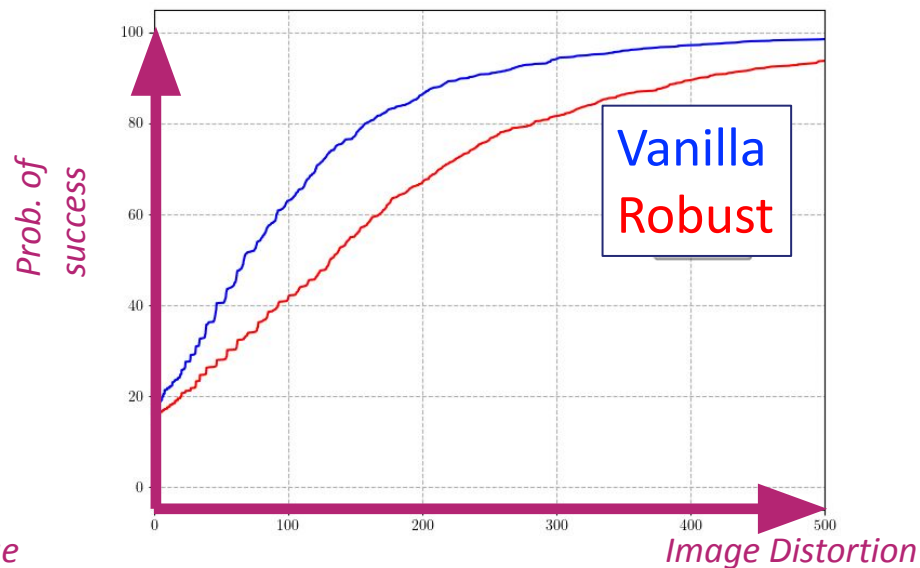


ResNet-50



« Towards Deep Learning Models Resistant to Adversarial Attacks », Madry et al. (MIT), 2018

EfficientNet-b0



« Adversarial Examples Improve Image Recognition », Xie et al. (Google), 2019

Types of defense

- Front-end: detector and/or denoiser
- Training: adversarial training, penalty
- Obfuscating:
 - include randomness/secrecy to prevent white-box scenario
 - include strong linearities to prevent gradient propagation

Biggest mistakes

- « Adversarial images against a vanilla DNN are now correctly classified »
An attack is a process, not a set of adversarial images
- No or poor self-assessment
 - **False feeling of security**
 - **No guarantees**
 - « On Adaptive Attacks to Adversarial Example Defenses », Tramer et al. , 2020
« We demonstrate that thirteen defenses recently published at ICLR, ICML and NeurIPS can be circumvented »

Robustness guarantees

- Consider a neural network $N: \mathbb{R}^d \rightarrow \mathbb{R}^C$
- Define a property \mathcal{P}
 - Input region of the data space $\phi \subset \mathbb{R}^d$
 - Target region of the output space $\psi \subset \mathbb{R}^C$
- The goal is to prove that

$$\forall x \in \phi, N(x) \in \psi$$

- In that case, we say that the network satisfies property \mathcal{P}

3 Characteristics of a certifier (be it formal or statistical):

- Soundness
a network not satisfying \mathcal{P} is never certified
- Completeness
any network satisfying \mathcal{P} is always certified
- Scalability
the cost grows reasonably with the size of the network

An important theoretical result (proven in *Katz et al.*, CAV 2017)
Sound and complete certification is NP-complete

- En utilisant une formulation mathématique des réseaux de neurones, on peut garantir que la sortie d'un NN se situe dans une certaine région (par exemple, la classification est correcte). Guaranteed to classify to label 0

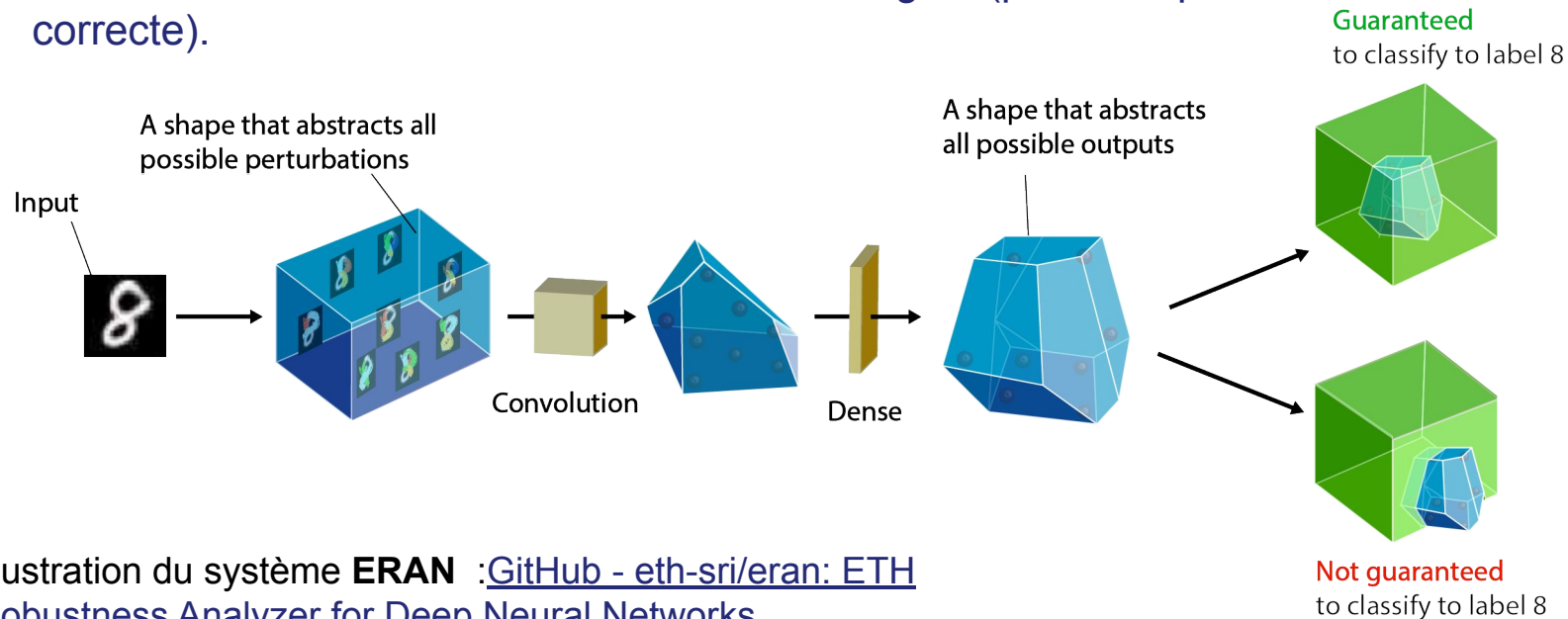


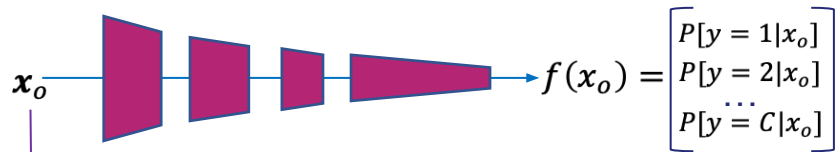
Illustration du système **ERAN** : [GitHub - eth-sri/eran](https://github.com/eth-sri/eran): [ETH Robustness Analyzer for Deep Neural Networks](https://ethz.ch/en/research-hubs/ethz-hub-ai/research-groups/ethz-hub-ai-research-group-on-robustness/)



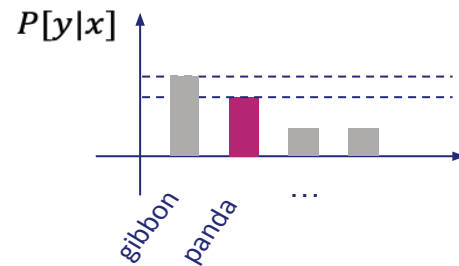
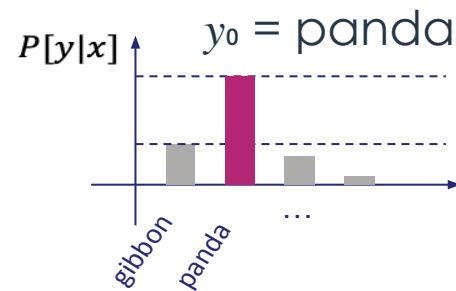
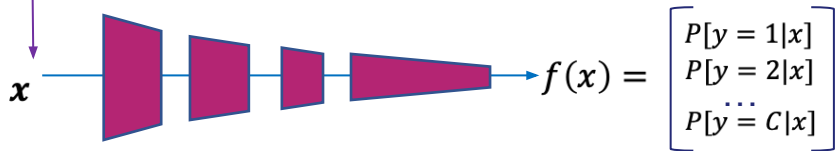
3. Problem Statement and Summary of Contributions



Problem Statement



+ noise



Probits = « predicted » probabilities

Estimer la fiabilité des prédictions de modèles DL

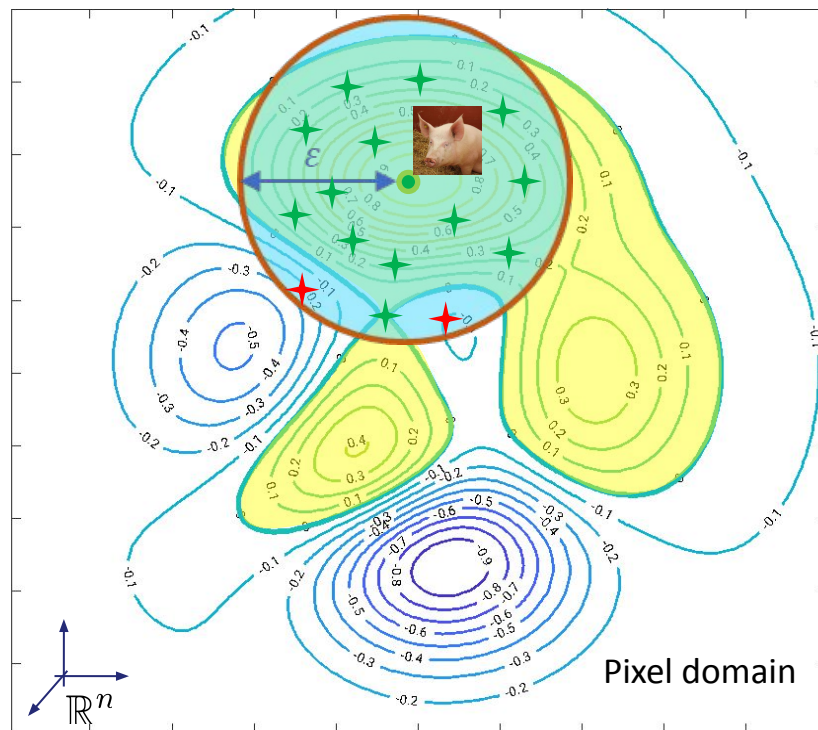
*Is there any perturbation $d\mathbf{x}$ of norm $\|d\mathbf{x}\| < \epsilon$ which leads to a failure:
 $\text{predict}(\mathbf{x}) \neq \text{predict}(\mathbf{x} + d\mathbf{x})$?*

- Approches formelles utilisant des solveurs SMT (Satisfiability Modulo Theory)
 - Problème NP-complet, fortement limité par la taille des réseaux étudiés
- Etudier une formulation alternative, probabiliste, conditionnée par un modèle de perturbation aléatoires \mathbf{U} :

$$\mathbb{P}(\text{predict}(\mathbf{x}) \neq \text{predict}(\mathbf{x} + \mathbf{U}))$$

- Probabilité qui peut être très faible

Evaluation statistique de fiabilité (Monte Carlo)

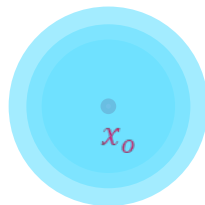


	Images bien classifiées'
	Image originale
	Image bruitée bien classifiée
	Image bruitée mal classifiée
	Distribution des incertitudes

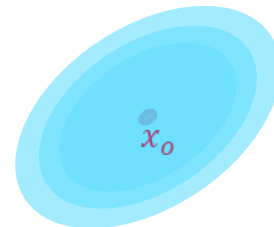
$$\Pr[Echec] = \frac{\# Echecs}{\# Echantillons}$$

Dans l'exemple $\Pr[echec] = \frac{2}{15}$

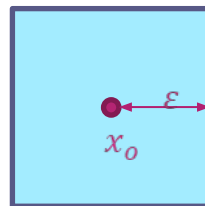
Modélisation statistique des incertitudes



$$U = \sigma G$$

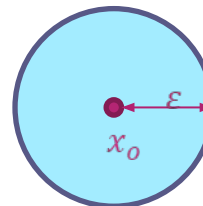


$$U = \Lambda^{1/2} G$$



ℓ_∞ norm

$$U = \varepsilon (2\Phi(G) - 1)$$

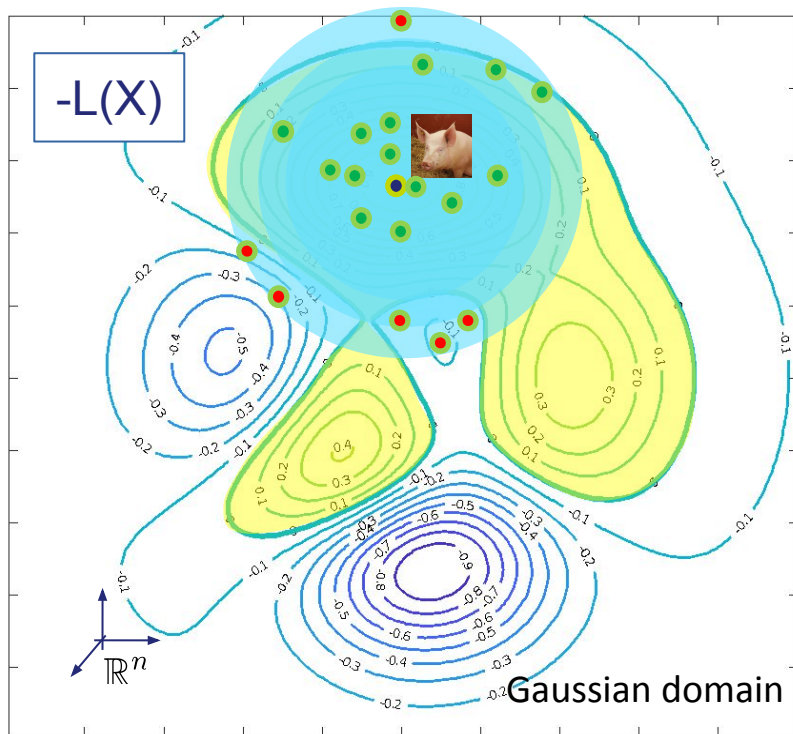


ℓ_2 norm

$$U = \varepsilon G_{1:d} / \|G\|$$

Statistical Reliability Estimation

$$P_f = \mathbb{P}_{X \sim \mathcal{N}(0,1)}[\operatorname{argmax} f(x_0 + X) \neq \text{pig}]$$

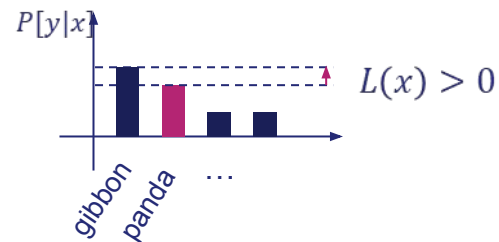


	Images classified as 'pig' by ResNet50
	Original (clean) image
	Misclassified Gaussian samples
	Correctly classified Gaussian samples
$L(X)$	Logit, output of NN
P_f	Probability of failure

Monte Carlo Estimation :

$$\hat{P}_f = \frac{1}{N} \sum_i^N \mathbf{1}_{f(x_i) \neq \text{pig}}$$

Problem Statement



$$L(x) := \max_{y \neq y_o} f_y(x) - f_{y_o}(x)$$

This quantity tells how close the uncertainties are to delude the classifier

$$G \sim \mathcal{N}(0, I) \longrightarrow U = \mathcal{T}(G) \longrightarrow X = x_o + U \longrightarrow p = \mathbb{P}[L(X) > 0] < p_c ?$$

Estimer par simulation des probabilités très faibles

- Estimateur de Monte Carlo basique **inefficace**
- Techniques d'estimation d'évènements rares:
 - **Importance sampling** : simulation suivant une distribution biaisée favorisant l'évènement rare, et correction des biais *a posteriori*.
 - **Importance splitting** : décomposition de l'évènement rare en une séquence d'évènements de plus en plus rares.
 - **Hamiltonian/Langevin Monte Carlo (HMC/LMC)** : utilisation de l'information du gradient pour accélérer la convergence de l'estimateur de fiabilité
 - **Adversarial-Attack Driven Importance Sampling** : utilisation de méthodes sophistiquées d'attaques adversariales combinées avec l'importance sampling

Last Particle Importance Splitting

- Variante de l'algorithme importance splitting fournissant des garanties théoriques pour obtenir une certification statistique de fiabilité
- Implémentation et application sur les cas d'étude MNIST, Acas Xu, ImageNet
- Publication à NeurIPS 2021
- Application sur des données de trajectoires ADS-B
- Publication à la Conference on AI for Defense (CAID) 2021

■ Simulation d'évènements rares basés sur l'utilisation du gradient (LMC)

- Implémentation de la méthode utilisant le gradient avec le framework PyTorch
- Application à des problèmes de classifications (MNIST/ImageNet)
- Publication à AISTATS 2023
- Développement et implémentation d'une méthode hybride LMC/Last Particle
- Preprint en cours sur l'algorithme hybride + preuve de consistance/biais

■ Importance Sampling via Attaques adversariales

- Implémentation d'importance sampling utilisant des attaques adversariales
- Lien entre Robustesse adversarial et fiabilité statistique
- Application à des problèmes de classifications (MNIST/ImageNet)
- Soumission prochaine dans une conférence de Machine Learning/Statistiques

4. Last Particle Algorithm for Reliability Hypothesis Tests



- **Statistical certification**

- Assume a statistical distribution of the input

$$X \sim \mathcal{U}(\mathcal{I})$$

- Define probability of failure

$$p = \mathbb{P}[f(X) \notin \mathcal{O}]$$

- Hypothesis Testing

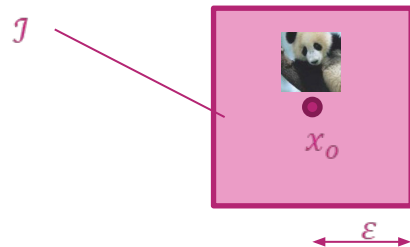
- $H_0: p > p_c$

Do not certify

- $H_1: p < p_c$

Certify

p_c critical level set by the user



- Last Particule Simulation

Simulation and Estimation of Extreme Quantiles and Extreme Probabilities,
A. Guyader, N. Hengartner, and E. Matzner-Løber, 2011

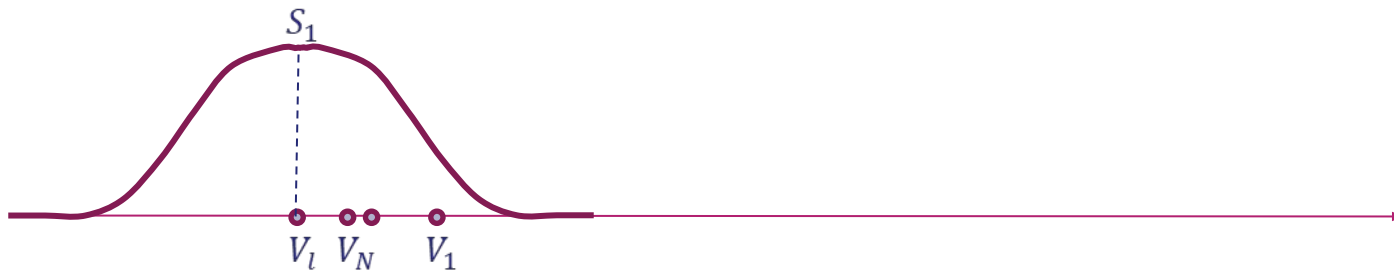


Consider r.v. $V \sim \mathcal{L}_V$ absolutely continuous with c.d.f. F_V

Define $g(x) = -\log(1 - F_V(x))$

g is a non decreasing function

Let $E = g(V)$. Then we have $E \sim \mathcal{E}(1)$



1. Sample N r.v. $V_j \sim \mathcal{L}_V$ i.i.d.

Define $l = \arg \min V_j$ and $S_1 = V_l$ $T_1 = g(S_1)$

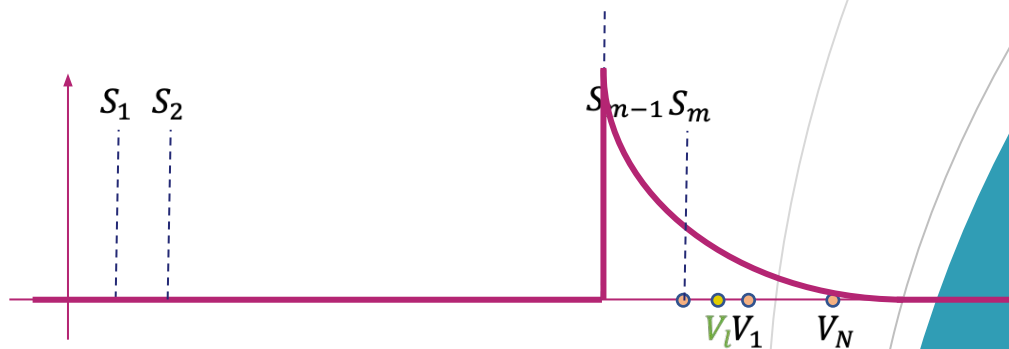
We have $T_1 \sim \mathcal{E}(N)$



2. Resample $V_l \sim \mathcal{L}_{V>S_1}$

Define $l = \arg \min V_j$ and $S_2 = V_l$. $T_2 = g(S_2)$

We have $T_2 \sim \mathcal{E}(N) + T_1 = \Gamma(2, N)$



m. Resample $V_l \sim \mathcal{L}_{V > S_{m-1}}$

Define $l = \arg \min V_j$ and $S_m = V_l$ $T_m = g(S_m)$

We have $T_m \sim \mathcal{E}(N) + T_{m-1} = \Gamma(m, N)$

User gives

- Threshold τ , critical probability p_c , significance level α , N particles
- Question: Do we have $p = \mathbb{P}[V > \tau] < p_c$?

Run Last Particle Simulation for m iterations $\rightarrow S_m$

Decision: Certify that $p < p_c$ if $S_m < \tau$

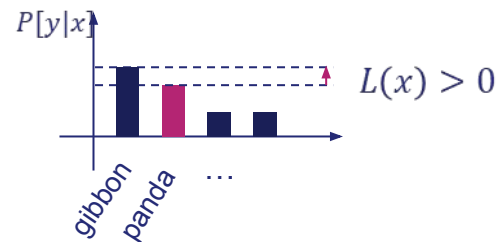
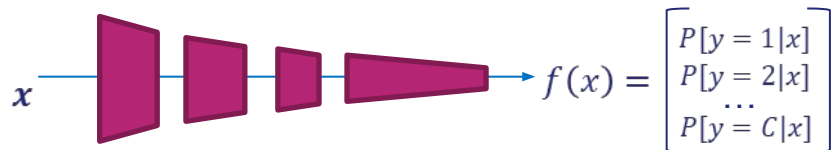
False positive: Suppose that $p > p_c$

$$\mathbb{P}[S_m < \tau] = \mathbb{P}[g(S_m) < g(\tau)] = \mathbb{P}[T_m < -\log p] = \frac{\gamma(m, -N \log p)}{\Gamma(m)} \leq \alpha$$

Problem solved by finding m s. t. $\frac{\gamma(m, -N \log p_c)}{\Gamma(m)} = \alpha$

N	p_c	$\alpha = 0.1$		$\alpha = 0.01$		$\alpha = 0.001$		
		m	\tilde{m}_1	m	\tilde{m}_1	m	\tilde{m}_1	
20	10^{-10}	489	489	512	514	529	532	↪ $\approx \times 3$
20	10^{-30}	1430	1431	1470	1471	1499	1502	
10	10^{-10}	251	251	267	269	280	283	↪ $\approx \times 3$
10	10^{-30}	726	726	754	755	774	777	
2	10^{-10}	56	56	64	65	69	73	↪ $\approx \times 3$
2	10^{-30}	154	155	167	169	177	180	

$$m \approx \tilde{m}_1 = \left\lceil \frac{1}{4} \left(z_\alpha + \sqrt{z_\alpha^2 - 4N \log(p_c)} \right)^2 \right\rceil$$

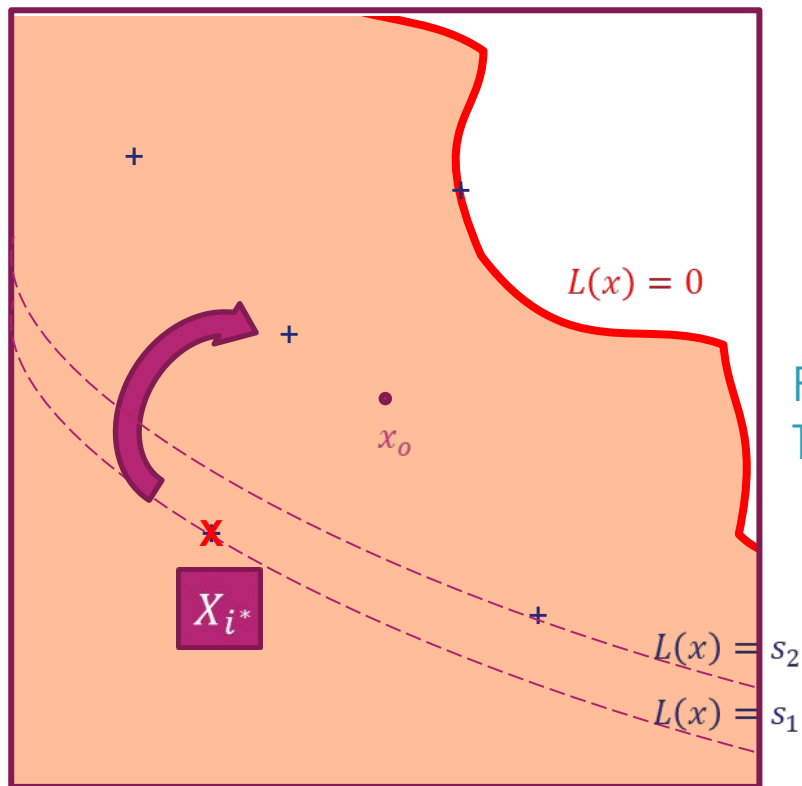


$$L(x) := \max_{y \neq y_o} f_y(x) - f_{y_o}(x)$$

This quantity tells how close the uncertainties are to delude the classifier

$$G \sim \mathcal{N}(0, I) \longrightarrow U = \mathcal{T}(G) \longrightarrow X = x_o + U \longrightarrow p = \mathbb{P}[L(X) > 0] < p_c ?$$

The Last Particle applied to ML



Repeat
T times

Randomly draw N samples
 $X_i = x_o + t(G_i)$

Compute scores
 $L(X_1), \dots, L(X_N)$

Find minimum
 $i^* = \arg \min L(X_i)$

Define threshold
 $S \leftarrow L(X_{i^*})$

Replace with one fresh particle

$X_{i^*} \leftarrow x_o + t(G)$ such that $L(X_{i^*}) > S$

How to sample a fresh particle?

$$X \leftarrow x_o + t(G) \text{ such that } L(X_{i^*}) > S$$

1. Randomly pick a survivor

$$G_b \sim \mathcal{U}(g_1, \dots, g_{N-1})$$

2. Repeat T times

a) Apply Gaussian kernel

$$G_a = \frac{(G_b + sN)}{\sqrt{1+s^2}} \text{ with } N \sim \mathcal{N}(0, I)$$

b) Accept upon score increase

$$G_b \leftarrow \begin{cases} G_a & \text{if } L(x_o + t(G_a)) > S \\ G_b & \text{otherwise} \end{cases}$$

s kernel strength

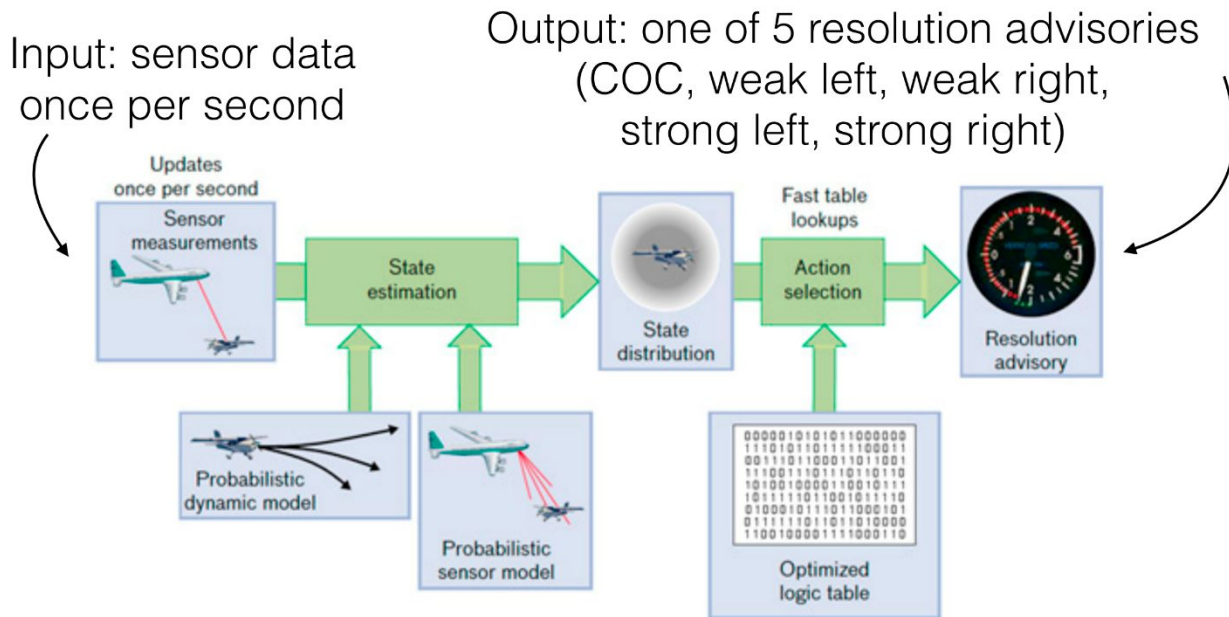
- Large to explore space
- Small to reduce rejection

T iterations

- Large enough to provide Independence
- Small to reduce complexity

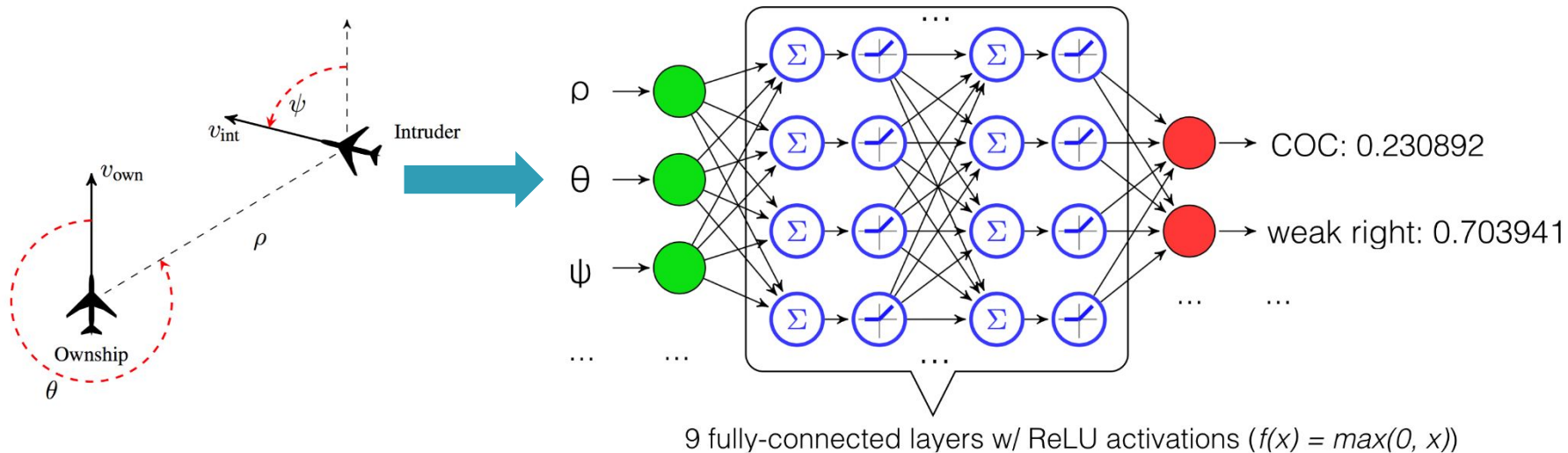
Système ACAS Xu

- Utilise une méthode de modélisation probabiliste (POMDP) → table de decision (>2GB)



Approximation de la table ACAS Xu avec des réseaux de neurones

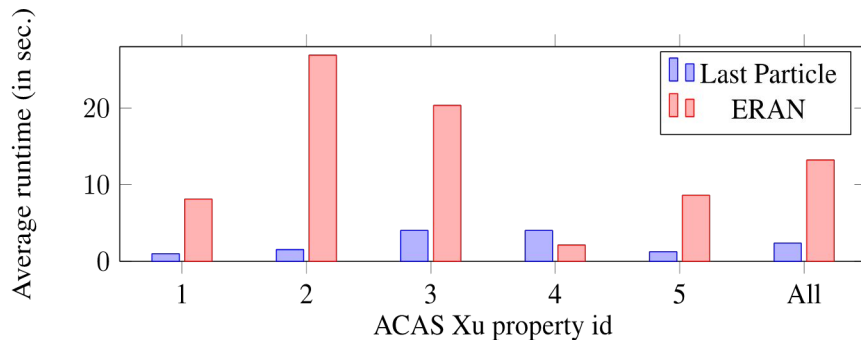
- Permet une compression efficace de la table (2GB \rightarrow qqes centaines de MB)



- **Question:** à quelle point cette approximation est-elle **fiable**?
- Solution partielle: on définit 5 propriétés que doivent **vérifier** le réseau.

Résultats expérimentaux

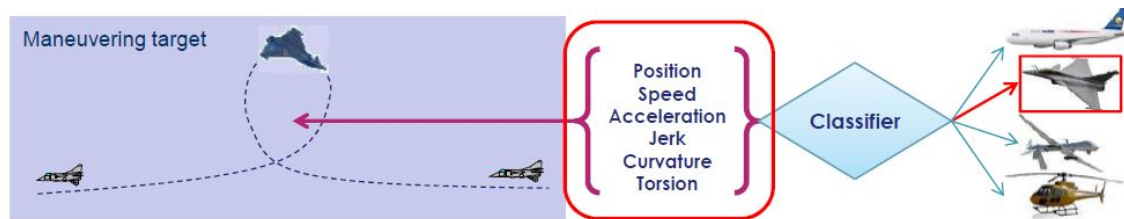
- Comparaison avec le système de vérification formelle ERAN
- On considère 5 réseaux de neurones et 5 propriétés à vérifier



		ERAN			
		Certified	Uncertified	Infeasible	TimeOut
Last Particle	Certified	107	9	1	1
	Uncertified	0	103	4	0

$p_c = 10^{-50}, \alpha = 0.05$, on a laptop

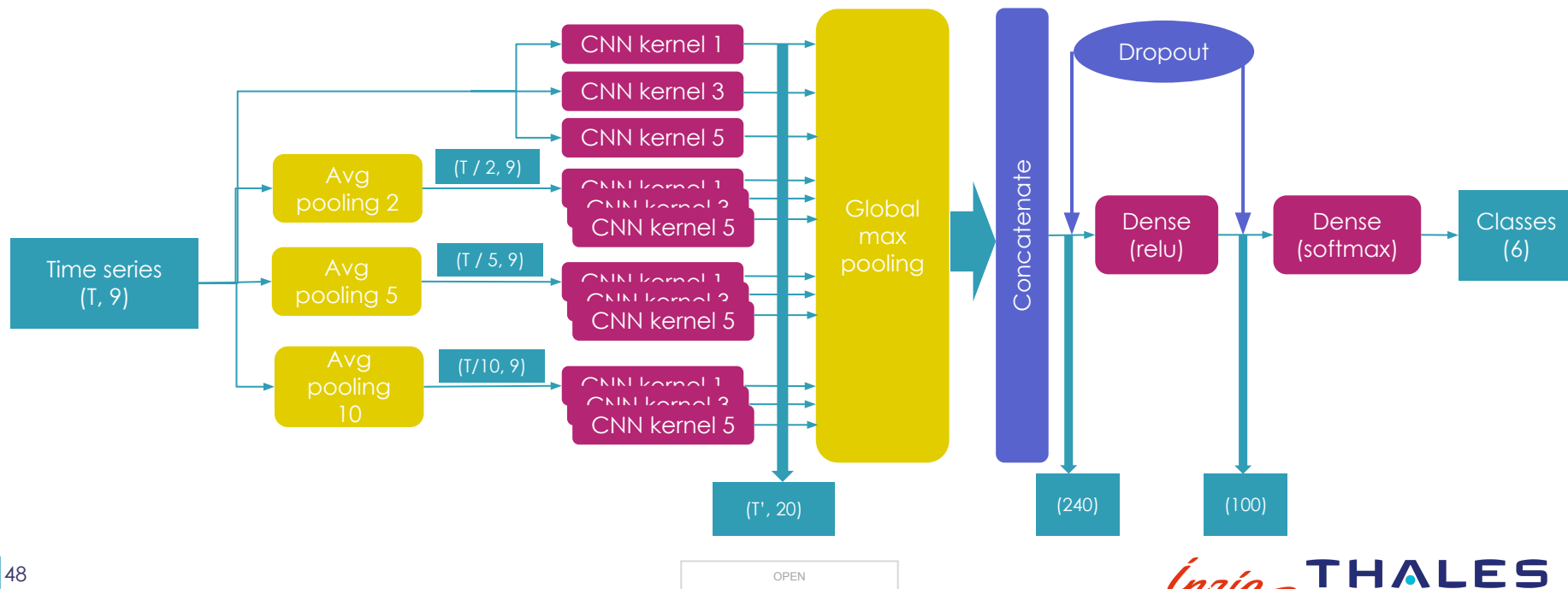
■ Analyser un dataset de trajectoires issues de données ADSB



- En analysant des features statistiques globales à partir des séries temporelles des trajectoires
 - A l'aide d'algorithmes de ML classiques sur données tabulaires (Random Forest, Gradient Boosting, MLP)
- En analysant des séries temporelles brutes, de longueurs variables
 - A l'aide de réseaux de neurones avec convolution et/ou LSTM

Architecture de modèle sur séries temporelles

- Architecture : Avg. Pooling + Conv1d + Globalmax



Résultats sur le dataset de features statistiques globales ($d = 72$)

ε	ERAN		Last Particle		
	Vérifié (%)	temps (sec. \pm std)	Vérifié (%)	temps (sec. \pm std)	faux positifs (%)
0.0001	100	5.0 \pm 5.0	100	5.26 \pm 0.1	0
0.0005	100	5.01 \pm 5.07	100	5.26 \pm 0.10	0
0.001	99	5.03 \pm 5.06	100	5.26 \pm 0.08	0.1
0.005	98	4.91 \pm 5.1	99	5.28 \pm 0.11	1
0.01	95	4.97 \pm 5.2	98	5.21 \pm 0.10	2
0.05	20	6.88 \pm 7.8	61	4.82 \pm 0.3	61
0.1	0.05	6.95 \pm 8.33	43	4.0 \pm 0.5	43
Moy.	74	5.53 \pm 6.17	89	5.01 \pm 0.5	15.4

Résultats sur le dataset de features statistiques avec/sans entraînement robuste

- Expériences sur une architecture de MLP simple avec différents type d'entraînement.

Modèle	ERAN	Last Particle	
	Vérifié (%)	Vérifié (%)	faux positifs (%)
Sans adv. training	73.14	86.29	13.14
adv. training ($\varepsilon = 0.025$, norme: l_2)	73.14	87.0	13.85
adv training ($\varepsilon = 0.025$, norme: l_∞)	73.57	96.0	22.42
random training ($\varepsilon = 0.01$, norme: l_2)	72.71	91.42	18.71
random training ($\varepsilon = 0.01$, norme: l_∞)	72.71	91.14	18.42

Résultats sur le dataset de features statistiques avec des méthodes d'arbres

- Expériences sur différents algorithmes d'ensembling basés sur des arbres de décisions aléatoires

Nb. estimateurs	RF		GB	
	Vérifié (%)	temps (sec. \pm std)	Vérifié (%)	temps (sec. \pm std)
100	87.6	7.38 \pm 0.004	73.6	11.9 \pm 0.08
500	89.6	15.43 \pm 0.065	76.2	32.98 \pm 0.35
1000	90.6	28.55 \pm 0.10	78.5	25.42 \pm 0.09

Résultats sur le dataset de séries temporelles ($d = T \times 9$)

➤ Expérience sur un modèle de convolution 1D

ε	ERAN		Last Particle		
	Vérifié (%)	temps (sec. \pm std)	Vérifié (%)	temps (sec. \pm std)	faux positifs (%)
0.01	100	1553.5 \pm 2396	100	332 \pm 22	0
0.05	100	10686 \pm 10368	100	319 \pm 26	0
Moy.	100	6120.25	100	325.5	0

Stochastic simulation + hypothesis test = network statistical robustness verification

- **Theoretical guarantees**
- **Efficient in terms of number of calls to the network function**
- **Scales up to ImageNet**
 - only done in Baluta et al., ISCE 2021 but for large p_c
- **Black box**
 - adapts easily to different architectures and other classifiers (SVMs, Random Forests, ...)
- **Yet, it does not use gradient information**
 - More sophisticated methods (e.g. Langevin Monte Carlo) can take advantage of back-propagation
 - This is a future direction of research



5. Gradient-Informed SMC for Robustness Estimation



Family parametrized by scalar b

$$\pi_b(dx) = \frac{h(x, b)}{Z_b} \pi(dx)$$

s.t.

$$\pi_0(dx) = \pi(dx)$$

$$\pi_\infty(dx) = \frac{\mathbb{I}(L(x) > 0)}{Z_\infty} \pi(dx)$$

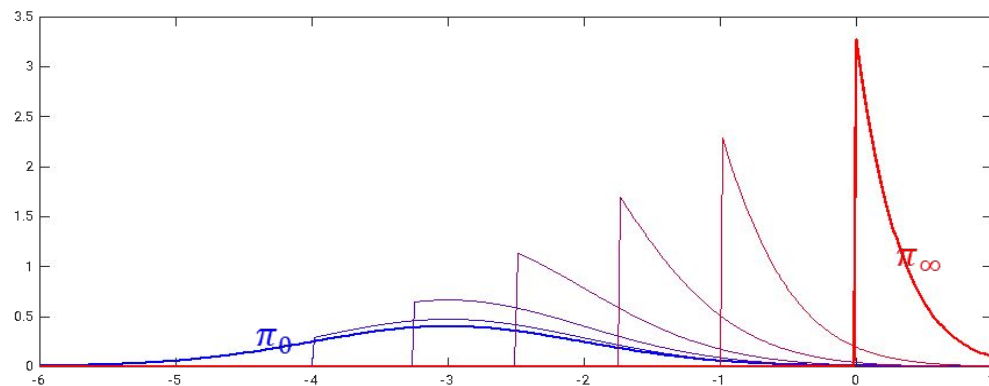
That way

$$Z_0 = 1 \qquad Z_\infty = p$$


Alternative family of distributions

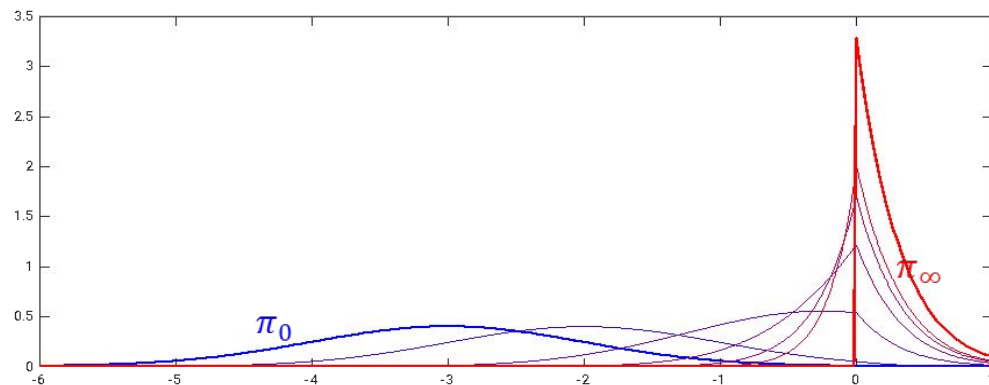
A $h(x, b) = \mathbb{I}(L(x) \geq -1/b)$

Importance Splitting
Multi-Level Splitting



B $h(x, b) = \exp(-b | -L(x) |_+)$

Tempered Sequential Monte Carlo



$$\hat{p} = \widehat{Z}_{\infty} = \prod_{m=1}^M \left(\frac{\widehat{Z_{b_m}}}{Z_{b_{m-1}}} \right) \quad \text{with } b_0 = 0 < b_1 < \dots < b_M = \infty$$

A

$$\frac{Z_{b_m}}{Z_{b_{m-1}}} = \mathbb{P} \left[L(X) > -\frac{1}{b_m} \mid L(X) > -\frac{1}{b_{m-1}} \right]$$

B

$$\frac{Z_{b_m}}{Z_{b_{m-1}}} = \mathbb{E}_{\pi_{b_{m-1}}} \left[e^{-(b_m - b_{m-1})| -L(X)|_+} \right]$$

■ Manage a sample of N particles




■ At each iteration

- These particles must be i.i.d. following distribution $\pi_{b_{m-1}}$
- They are used to estimate $Z_{b_m} / Z_{b_{m-1}}$

■ Mechanism

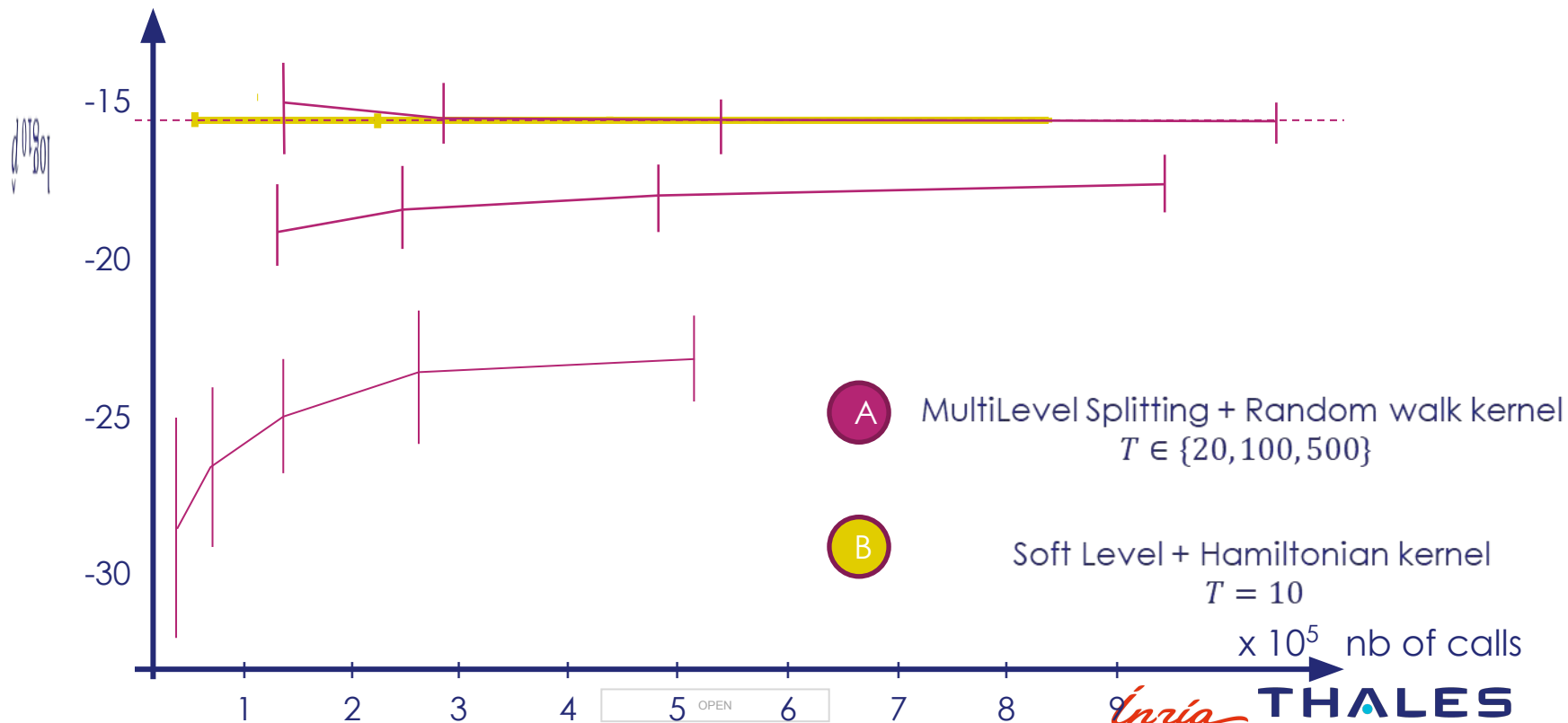
- Selection:
 - Kill K_{m-1} particles + Draw uniformly K_{m-1} particles among survivors
- Mutation:
 - Independently mutate particles with a kernel invariant to π_{b_m}

- Choice of tempering parameters: $b_0 = 0 < b_1 < \dots < b_M = \infty$
 - Too rapidly \rightarrow increase of the variance of $\left(\frac{\widehat{Z_{b_m}}}{Z_{b_{m-1}}}\right)$
 - Too slowly \rightarrow increase the number of iterations to reach the rare event
- Adaptive rule [Beskos, 2016]
 - Maintain Efficient Sample Size constant
 - Find b_{m+1} s.t. $\text{ESS}(b_{m+1}, \{X_m^{1:N}\}) = \alpha N$

- Random walk kernel + Metropolis step
 - Random walk  
- Hamiltonian Monte Carlo
 - L iterations of the Verlet scheme with potential $U = -\log \pi_b + cst$ 
 - Computation of $\nabla U(x)$
 - Easy thanks to backpropagation / auto-diff
 - Complexity $\approx 2 \times$ complexity of computing $U(x)$

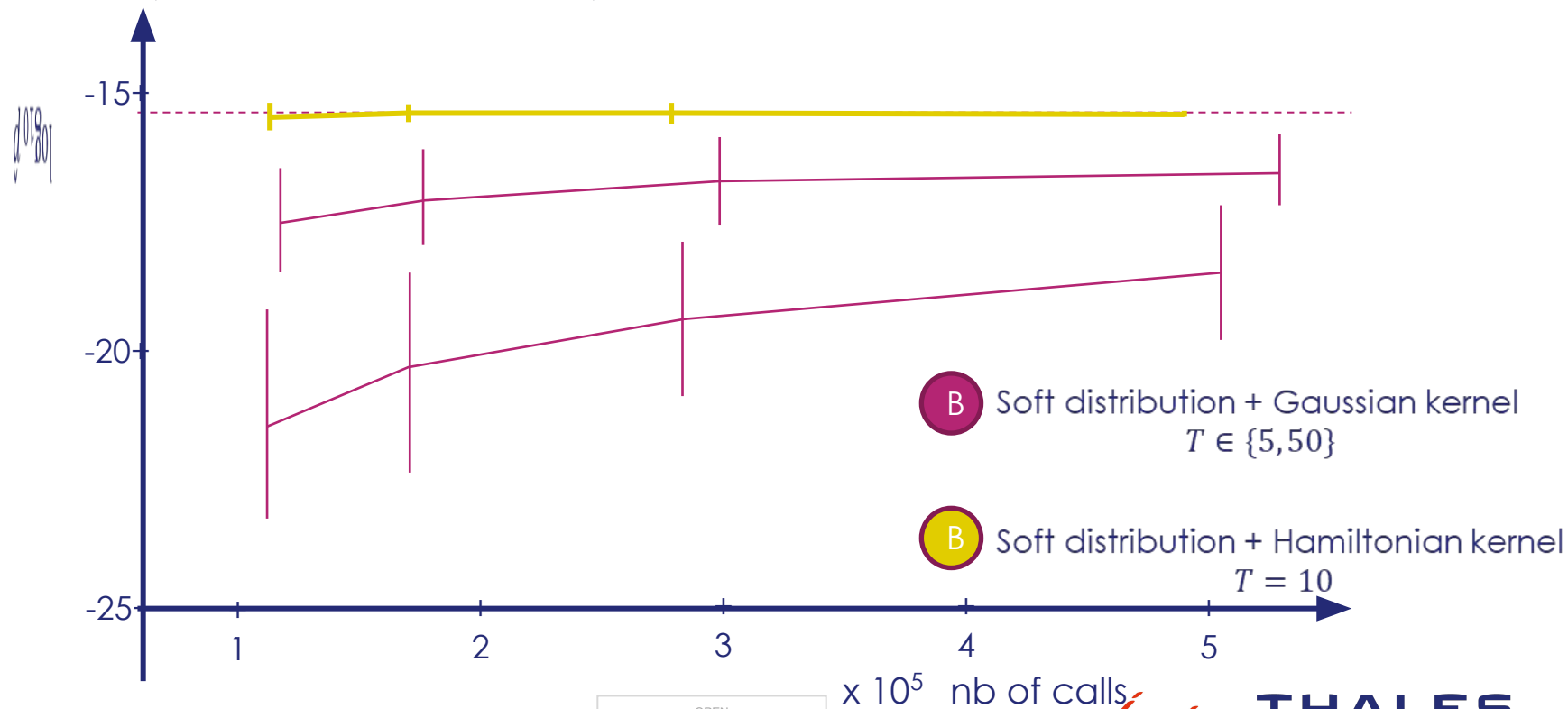
Experimental results

- MNIST, $d = 784$, uncertainties uniformly distributed over ℓ_∞ ball
- $N \in \{32, 64, 128, 256, 512, 1024\}$, 100 runs



Experimental results

- MNIST, $d = 784$, uncertainties uniformly distributed over ℓ_∞ ball
- $N \in \{32, 64, 128, 256, 512, 1024\}$, 100 runs

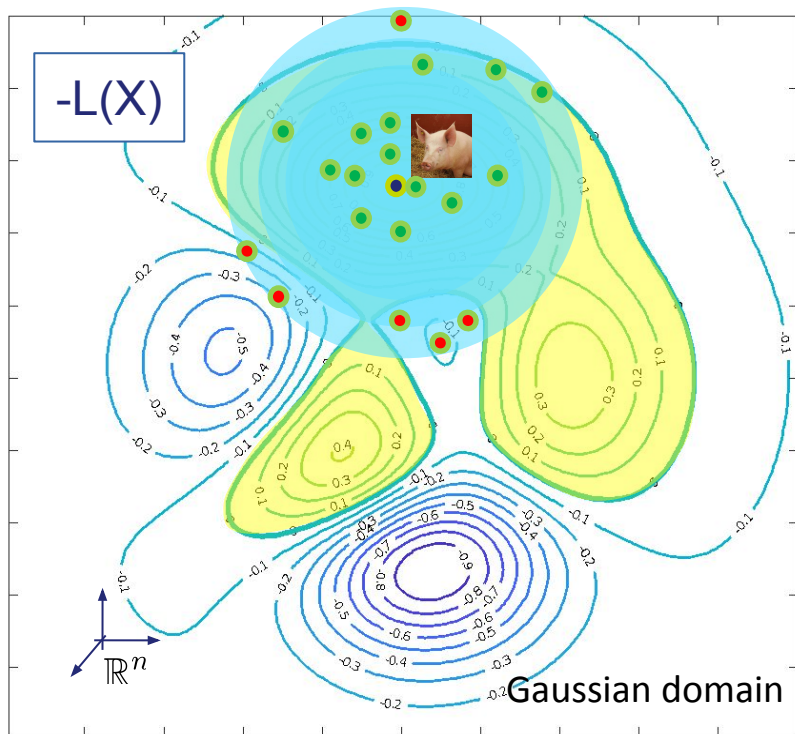


6. Adversarial Attack-Driven Importance Sampling



Statistical Reliability Estimation

$$P_f = \mathbb{P}_{X \sim \mathcal{N}(0,1)} [\operatorname{argmax} f(x_0 + X) \neq \text{pig}]$$



	Images classified as 'pig' by ResNet50
	Original (clean) image
	Misclassified Gaussian samples
	Correctly classified Gaussian samples
$L(X)$	Logit, output of NN
P_f	Probability of failure

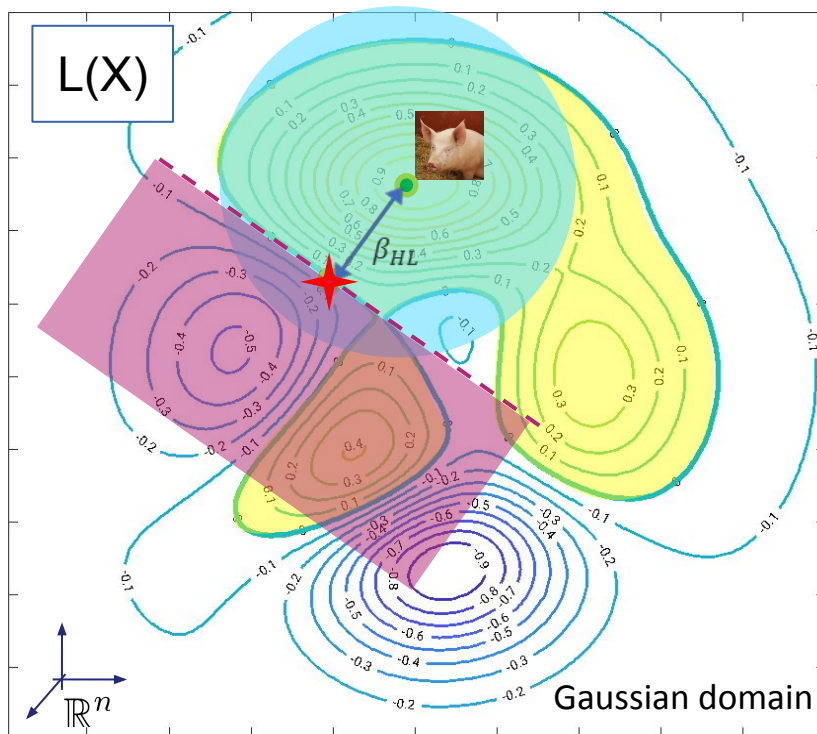
Monte Carlo Estimation :

$$\hat{P}_f = \frac{1}{N} \sum_i^N \mathbf{1}_{f(x_i) \neq \text{pig}}$$

Connection between Adversarial Attacks and Reliability Engineering

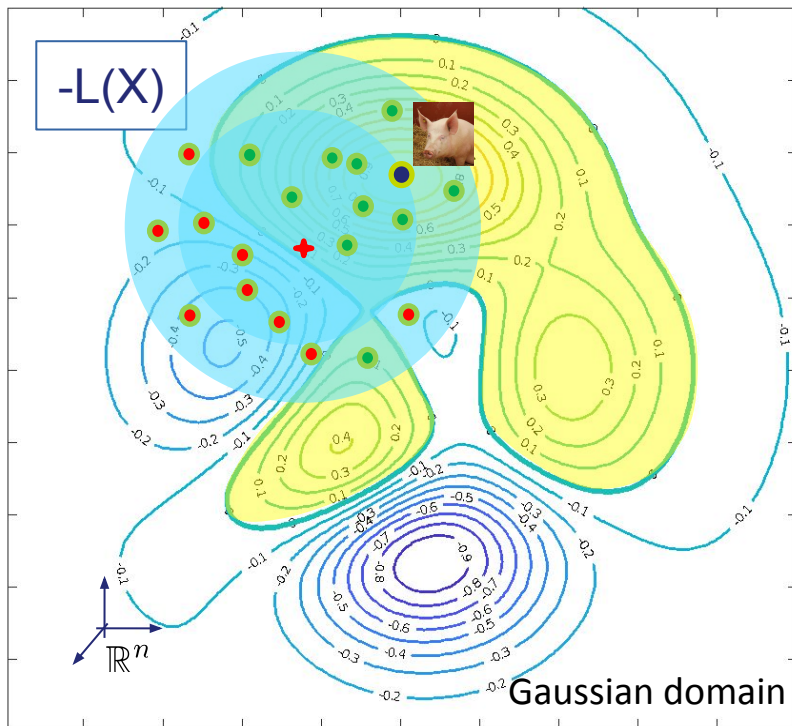
First-Order Reliability Method (FORM)

$$\hat{P}_{FORM} = \Phi(-\beta_{HL})$$



	Images classified as 'pig'
	Approximate Linear Failure Region
	Image originale
	Attaque adverse
$L(X)$	Logit, sortie du réseau
Φ	CDF of the Gaussian distribution
β_{HL}	Distance to closest Adversarial Attack

Importance Sampling (IS)



	Images classified as 'pig' by ResNet50
●	Original (clean) image
+	
●	Misclassified Gaussian samples
●	Correctly classified Gaussian samples
$q(x)$	Biasing distribution
\hat{P}_f^{IS}	Importance Sampling Estimator

IS Estimation:

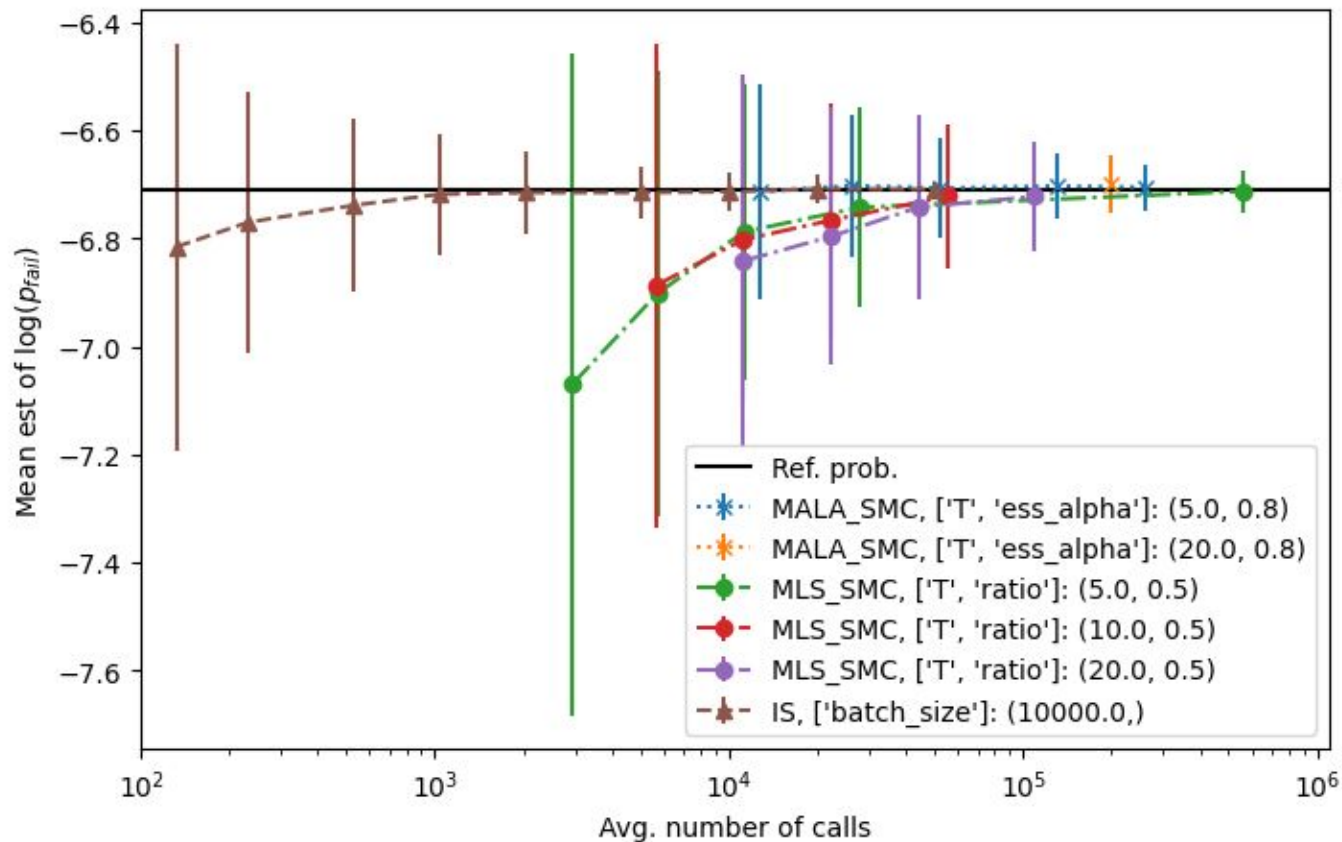
$$\hat{P}_f^{IS} = \frac{1}{N} \sum 1_{f(\tilde{X}_i) \neq \text{pig}} W(\tilde{X}_i)$$

Where $W(x) = \frac{f_X(x)}{q(x)}$ and $\tilde{X}_i \sim q(x)dx$

Experimental Results

MNIST

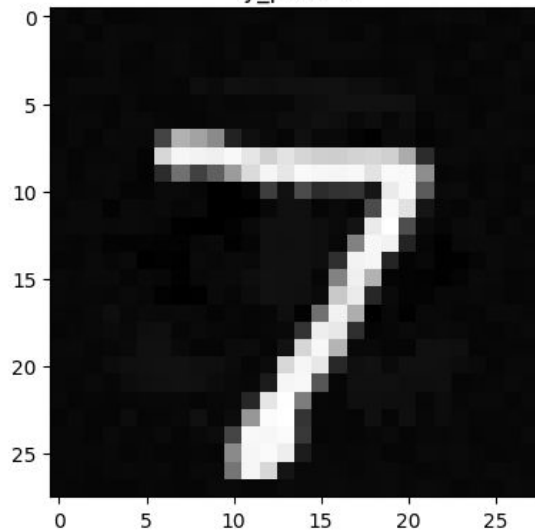
IS	Importance Sampling
ML S	Importance Splitting
MA LA	Gradient-Inf. SMC a.k.a. LMC



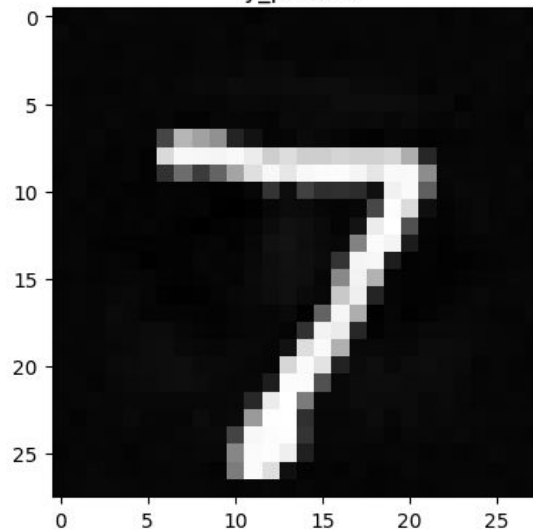
Experimental Results

FORM Results:

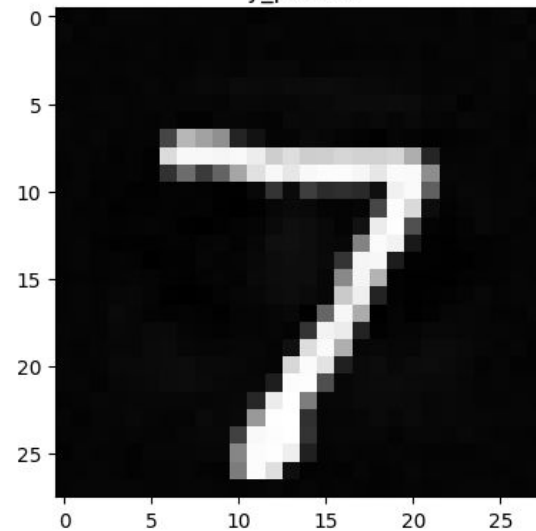
Carlini attack
form: $1.36\text{E-}04$, $\beta = 3.641$
 $y_{\text{pred}}: 3$



FMNA attack
form: $1.27\text{E-}04$, $\beta = 3.658$
 $y_{\text{pred}}: 3$

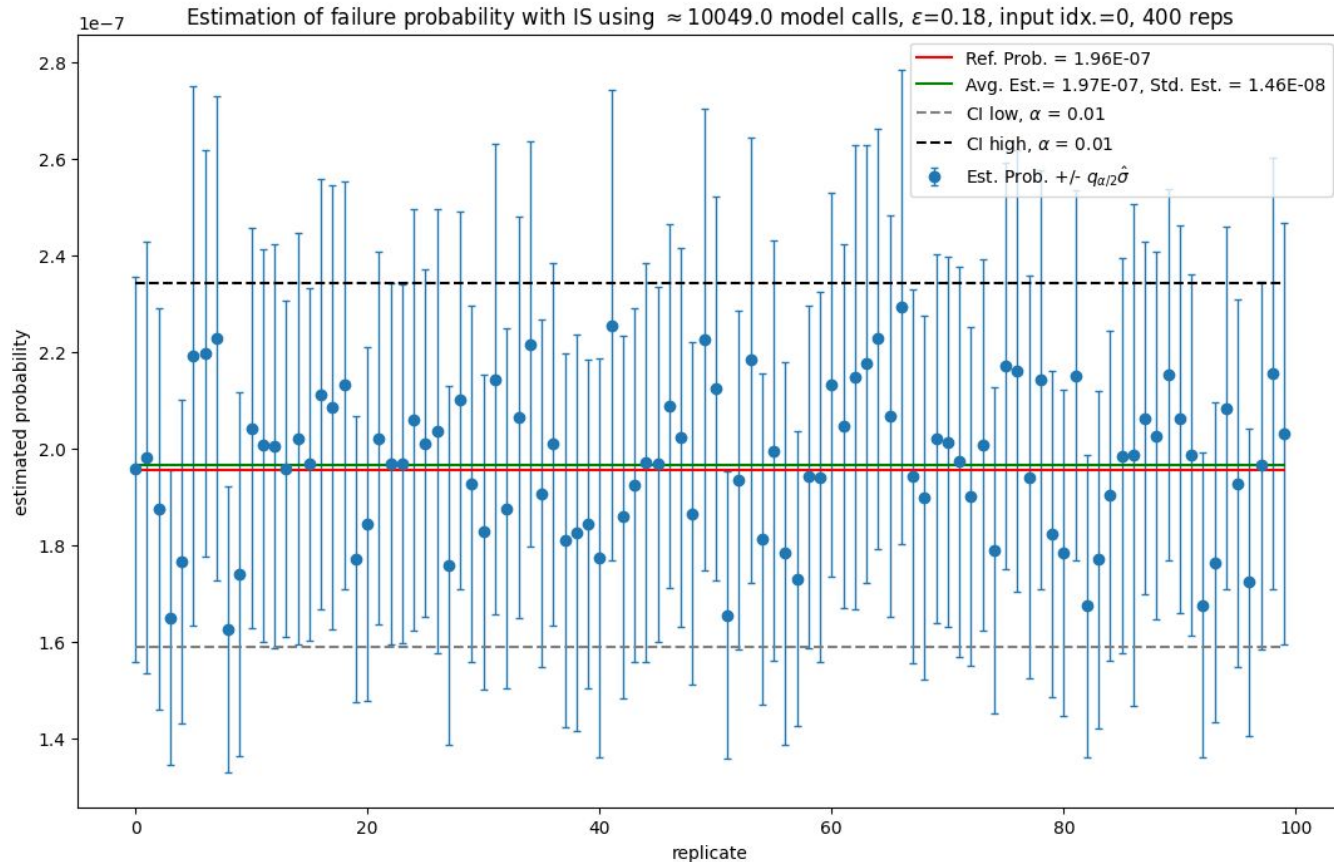


HLRF attack
form: $1.37\text{E-}04$, $\beta = 3.638$
 $y_{\text{pred}}: 3$



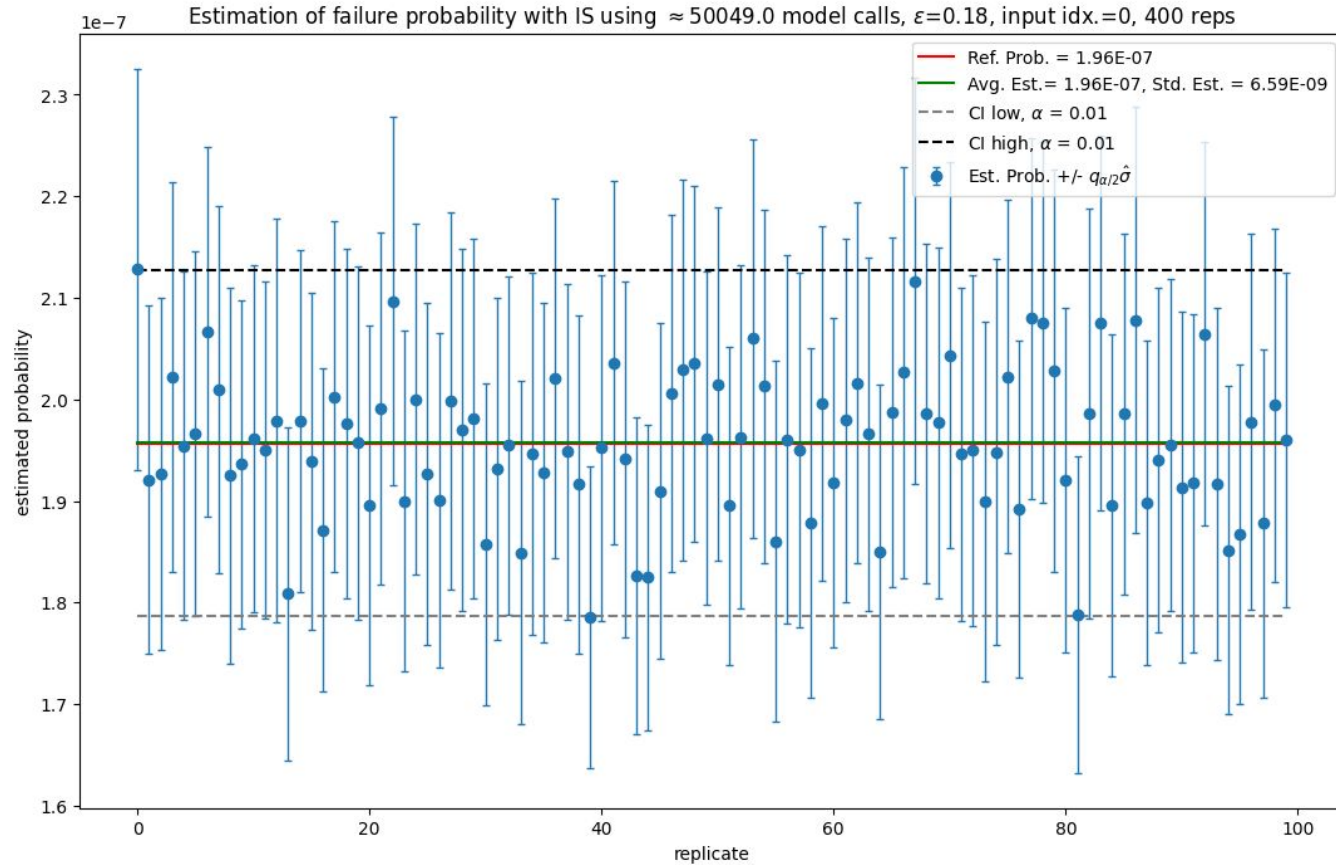
Experimental Results

IS



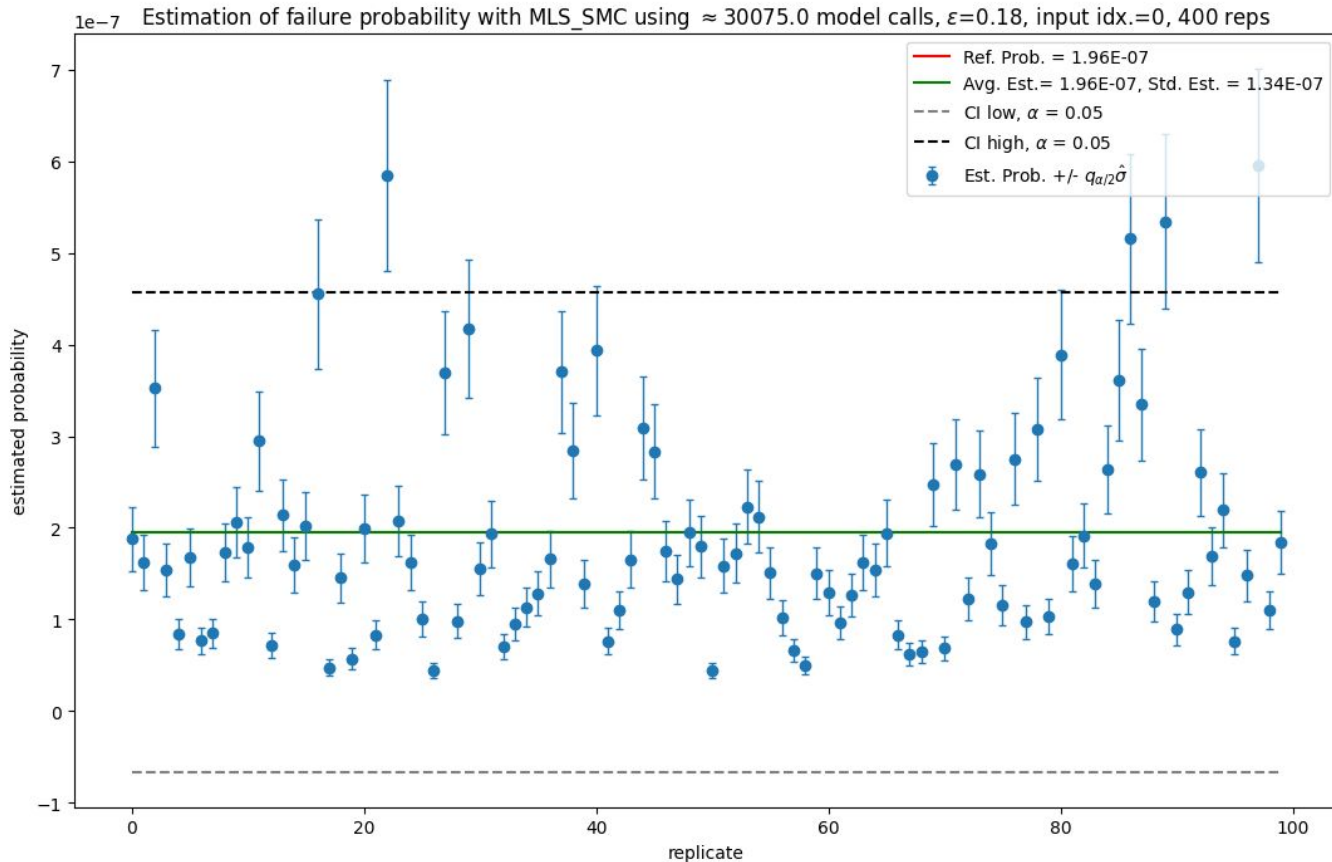
Experimental Results

IS



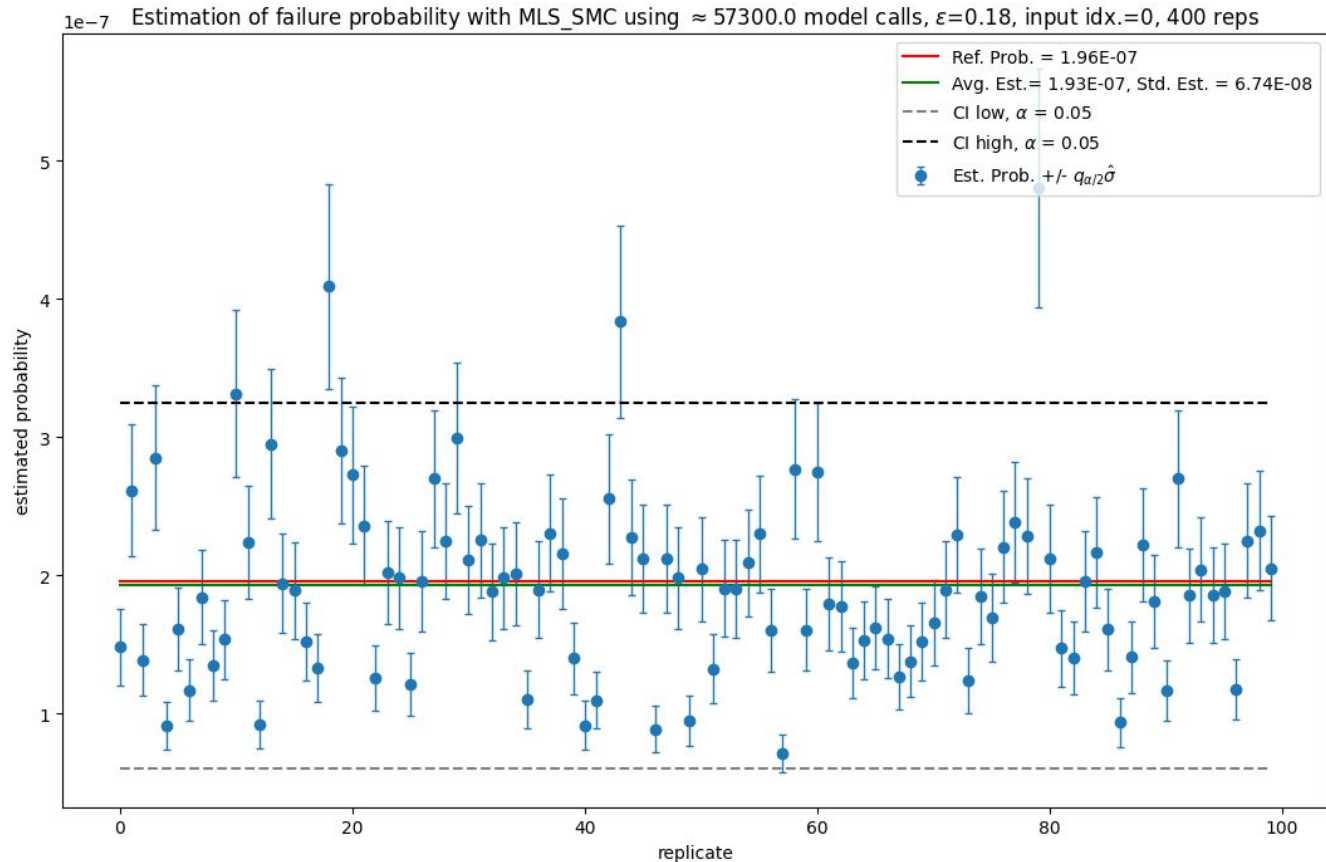
Experimental Results

MLS



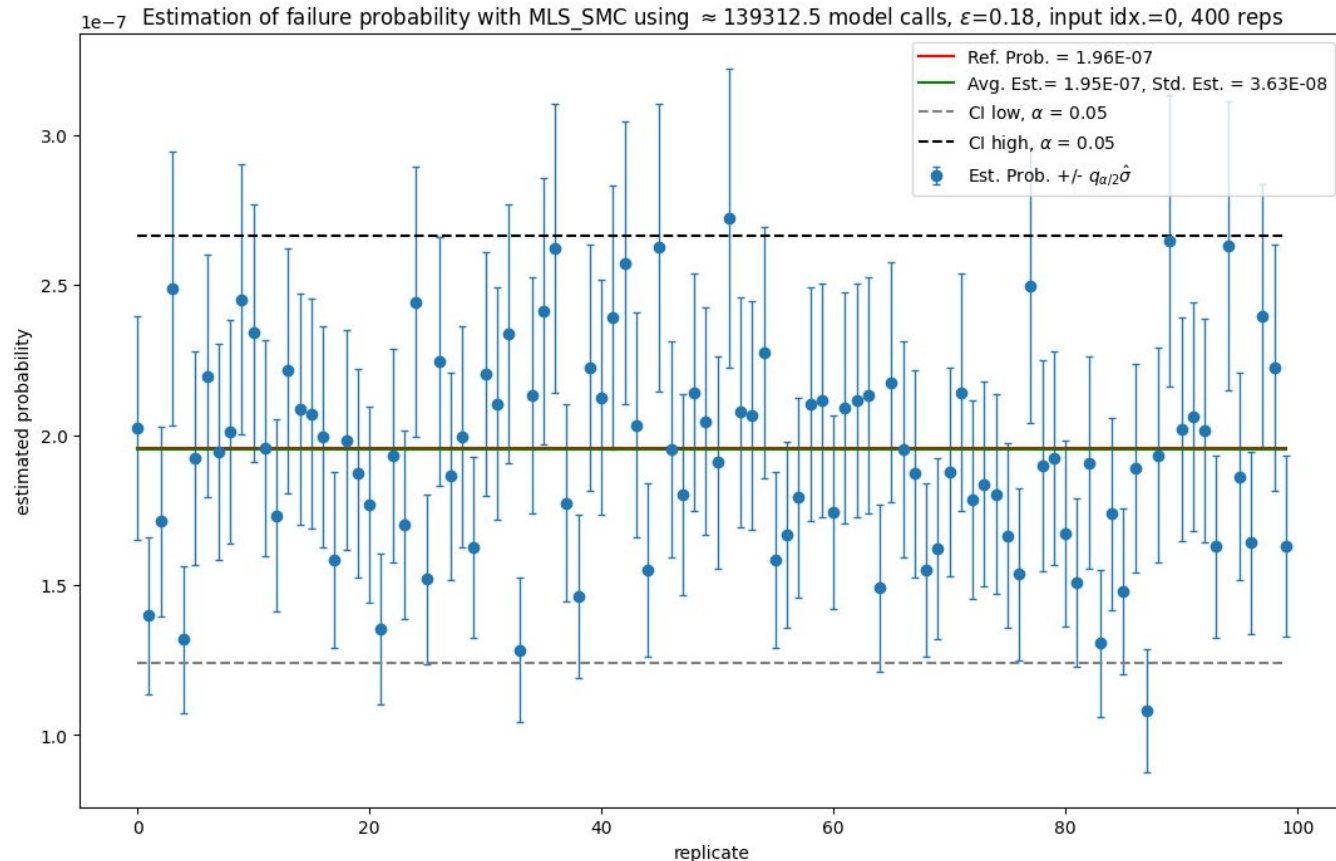
Experimental Results

MLS



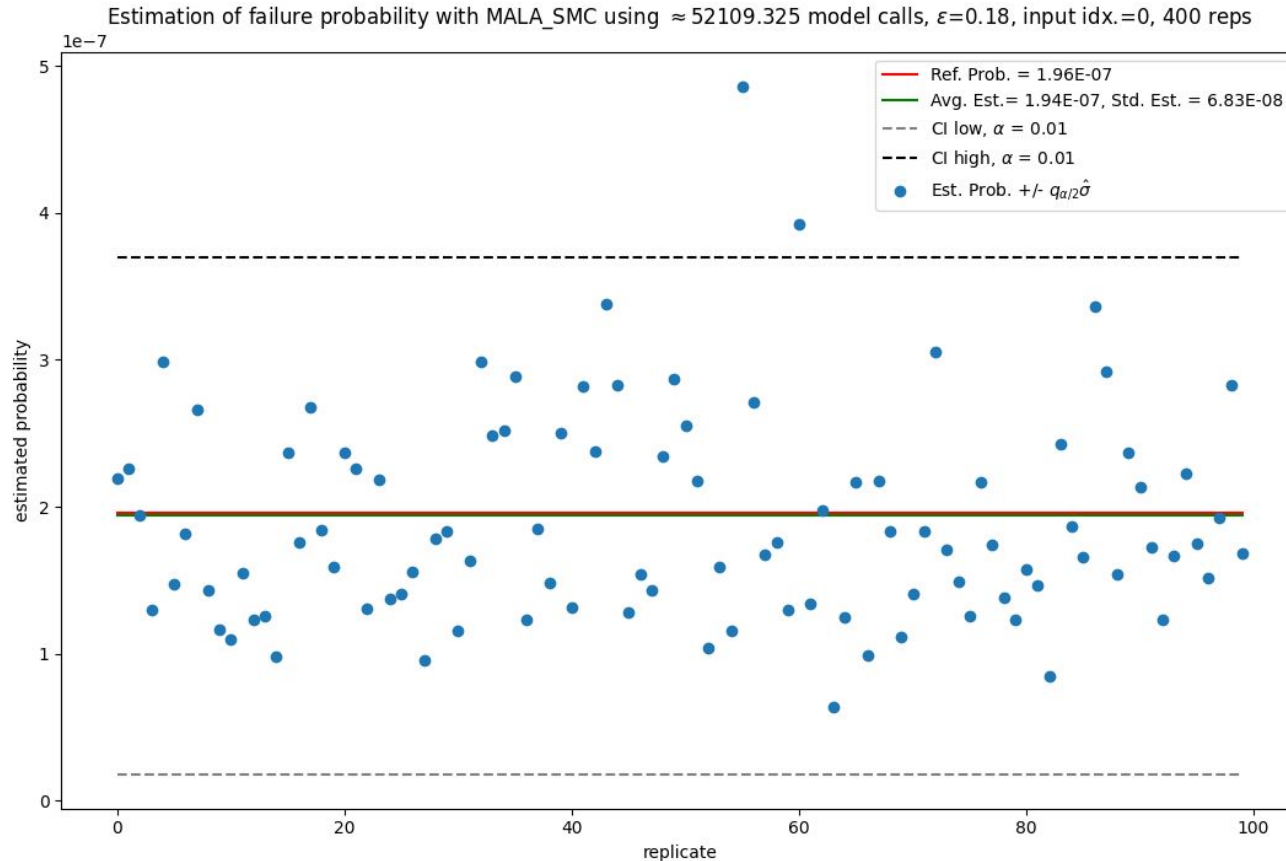
Experimental Results

MLS



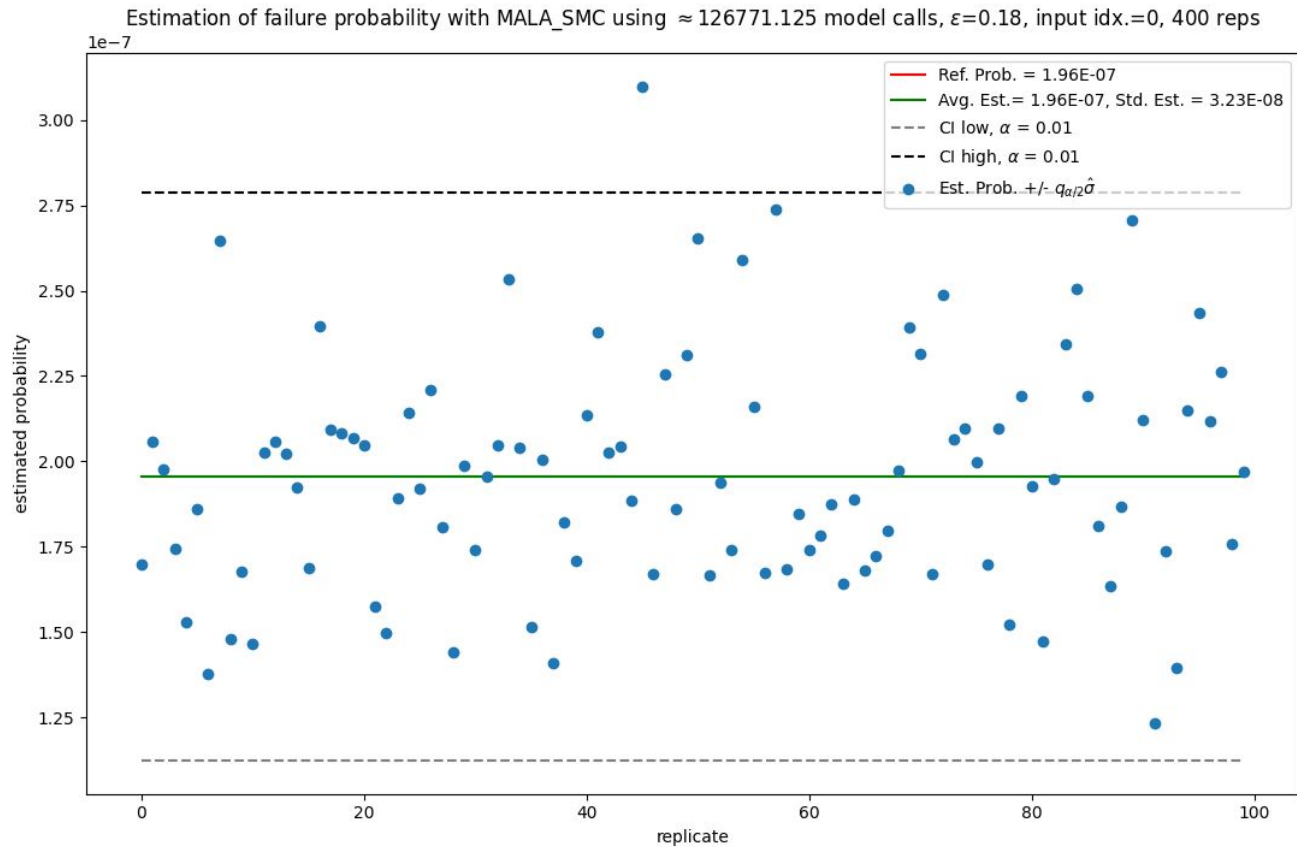
Experimental Results

LMC



Experimental Results

LMC

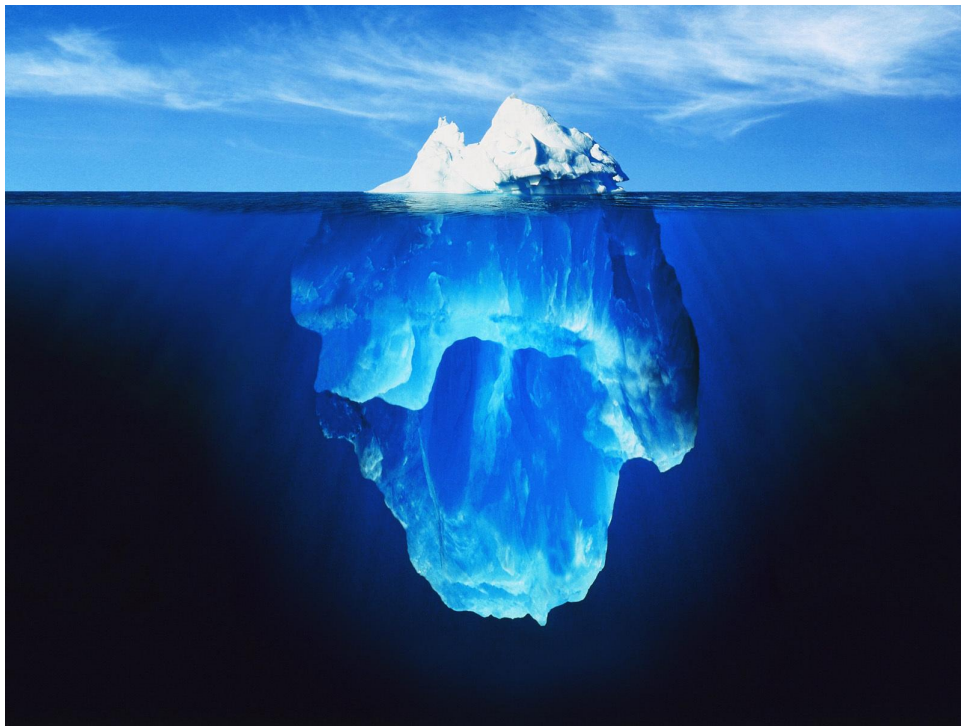




7. Conclusion



La partie émergée de l'Ice Berg (Sécurité de l'IA)



Attaque lors du
test/déploiement
Exemples et défenses « adversariales »

Attaque pendant ou avant
l'entraînement
Menace contre les données d'entraînement:

- Empoisonnement des données (parfois même involontaire)
- Cheval de Troie
- « Radio-activité »

Sources primaires

- [Gradient-Informed Neural Network Statistical Robustness Estimation](#), K. Tit, Teddy Furon, Mathias Rousset, 26th International Conference on Artificial Intelligence and Statistics (AISTATS), Apr 2023, Valencia, Spain.
- [Efficient Statistical Assessment of Neural Network Corruption Robustness](#), K. Tit, T. Furon, M. Rousset, NeurIPS 2021 - 35th Conference on Neural Information Processing Systems, , Dec 2021, Sydney (virtual), Australia.
- [Évaluation statistique efficace de la robustesse de classifieurs](#), K. Tit, T. Furon, M. Rousset, L.-M. Traonouez, CAID 2021 - Conference on Artificial Intelligence for Defense, Nov 2021, Rennes, France. pp.115-125

Sources secondaires

- [A statistical approach to assessing neural network robustness](#), Stefan Webb, Tom Rainforth, Yee Whye Teh, and M Pawan Kumar. In International Conference on Learning Representations, 2019
- [Simulation and estimation of extreme quantiles and extreme probabilities](#), Arnaud Guyader, Nicolas Hengartner, and E. Matzner-Løber. Applied Mathematics & Optimization, 64:171–196, 10 2011. doi: 10.1007/s00245-011-9135-z.
- [On the convergence of adaptive sequential monte carlo methods](#), Alexandros Beskos, Ajay Jasra, Nikolas Kantas, and Alexandre Thiery. The Annals of Applied Probability, 26(2):1111–1146, 2016.

THALES



Questions ?

www.thalesgroup.com

OPEN

