Robust Decision Trees Against Adversarial Examples

Honge Chen[1], Huan Zhang[2], Duane Boning[1] and Cho-Jui Hsieh[2]
[1]MIT [2]UCLA

36th International Conference on Machine Learning (ICML)
June 11, 2019, Long Beach, CA, USA

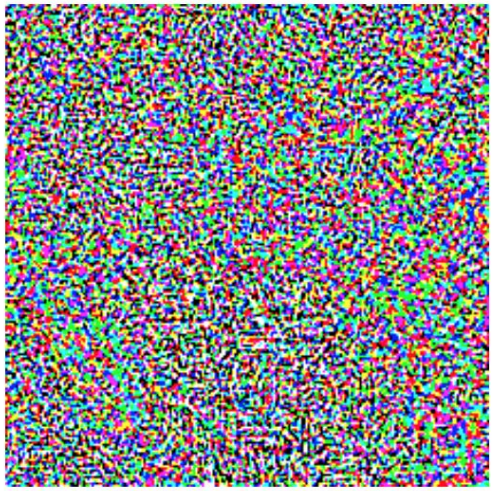
Code (XGBoost compatible!) is available at: https://github.com/chenhongge/RobustTrees

ICML

MIT UCLA

DNNs are vulnerable to **adversarial attacks**

$+ .007 \times$



$=$

Prediction:
Panda (57.7%) ✓

Imperceptible (very small) Adversarial Perturbation

Prediction:
Gibbon (99.3%) ✗

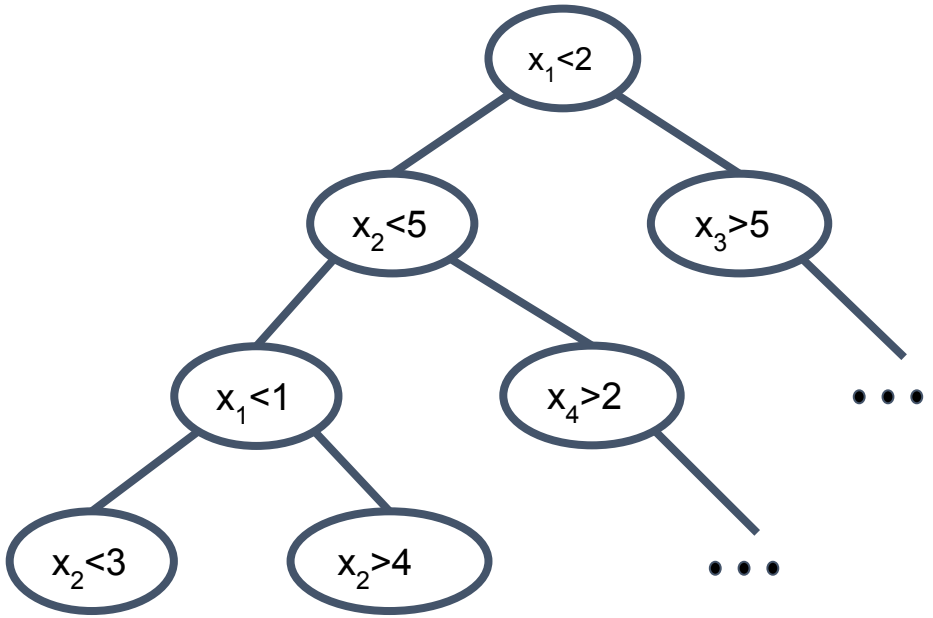
Goodfellow et al, *Explaining and harnessing adversarial examples*, ICLR 2015

Many **defenses** were proposed for **DNNs**:

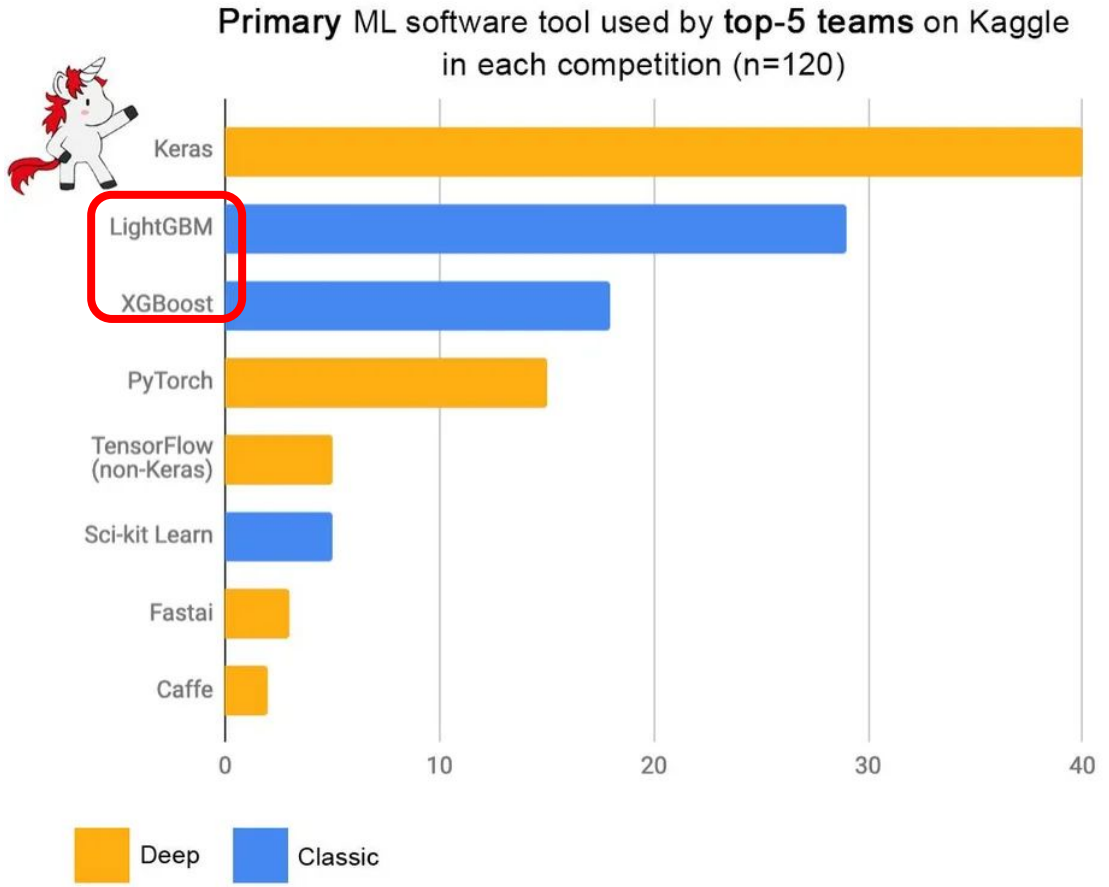
Literature	Method
Madry et al., ICLR 2018	Robust min-max optimization with alternative gradient descent/ascent on weights and inputs
Wong et al., ICML 2018	Certified robust training with linear bounds by ReLU relaxation
Raghunathan et al., ICLR 2018	Certified robust training with relaxation and Semidefinite Programming
Gowal et al., arXiv 2018	Fast certified robust training with interval bound propagation
Xiao et al., ICLR 2019	Certified robust training by enforcing ReLU stability
Zhang et al., arXiv 2019	Stable and efficient certified robust training using tight CROWN bound and interval bound propagation

However, the robustness of **tree-based models** is largely unexplored...

Decision Trees

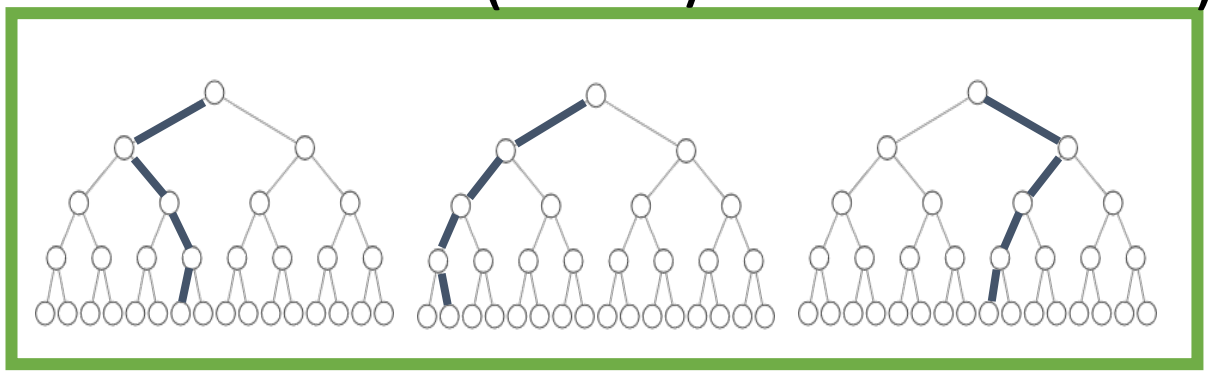


"Among the 29 challenge winning solutions published at Kaggle's blog during 2015, 17 solutions used XGBoost." Chen et al. KDD '16



Source: https://twitter.com/fchollet/status/1113476428249464833 (April 2019)

Tree Ensembles (GBDT/RandomForest)



Adversarial examples also exists in tree-based models.

Original

Ankle Boot

Adversarial
(0.074 ℓ_∞ distortion)

Shirt

Original

2

Adversarial
(0.069 ℓ_∞ distortion)

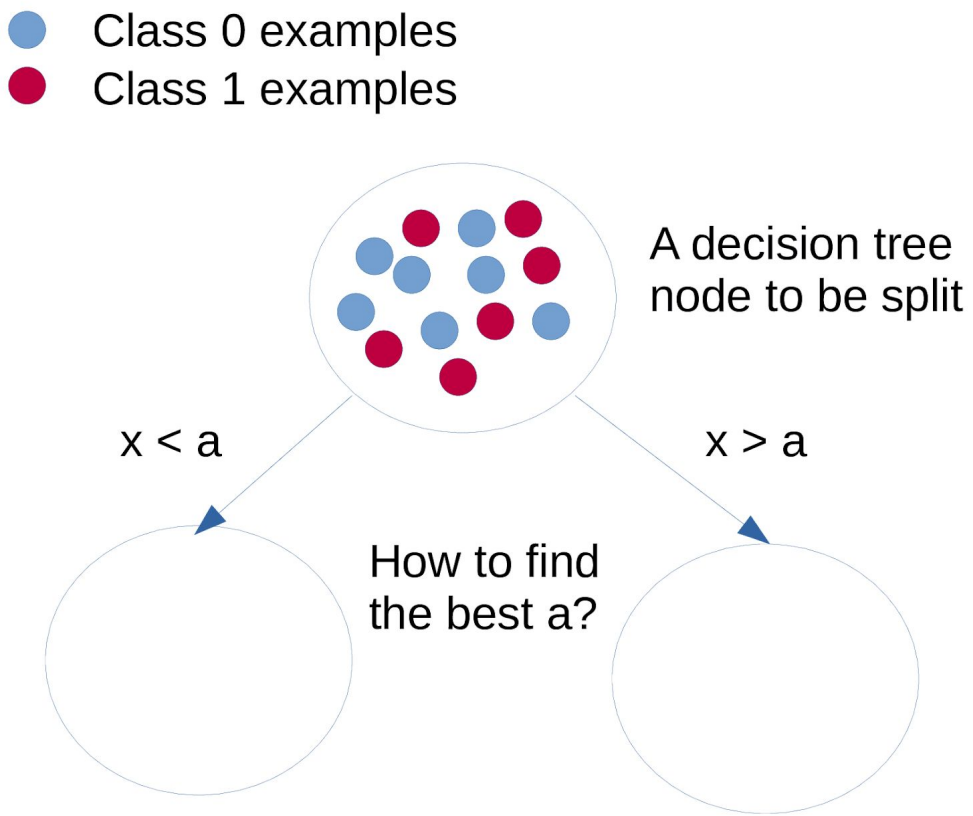
8

Original and adversarial examples of natural GBDT models with 200 trees. Here we use a general search-based black-box attack from Cheng et al. ICLR 2019

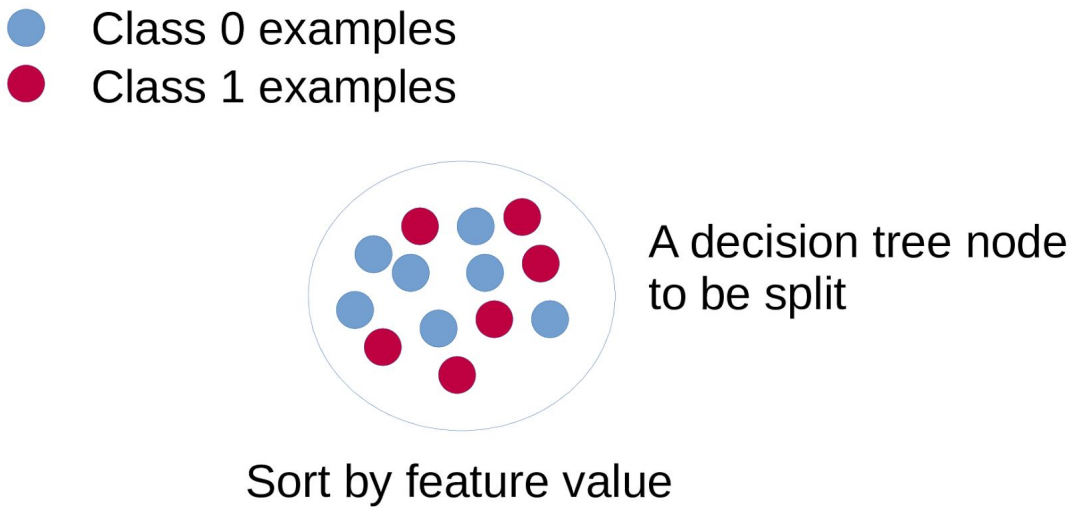
Why adversarial examples also exists in tree-based models?

Ordinary (natural) decision tree training finds the best split to minimize error,
without considering robustness!

How to find the best split in an ordinary decision tree?



How to find the best split in an ordinary decision tree?



Repeat for each feature, finds the best feature and best split value

In the original (natural) decision tree training

$$j^*, \eta^* = \arg\max_{j,\ \eta} S(j,\ \eta,\ \mathcal{I})$$

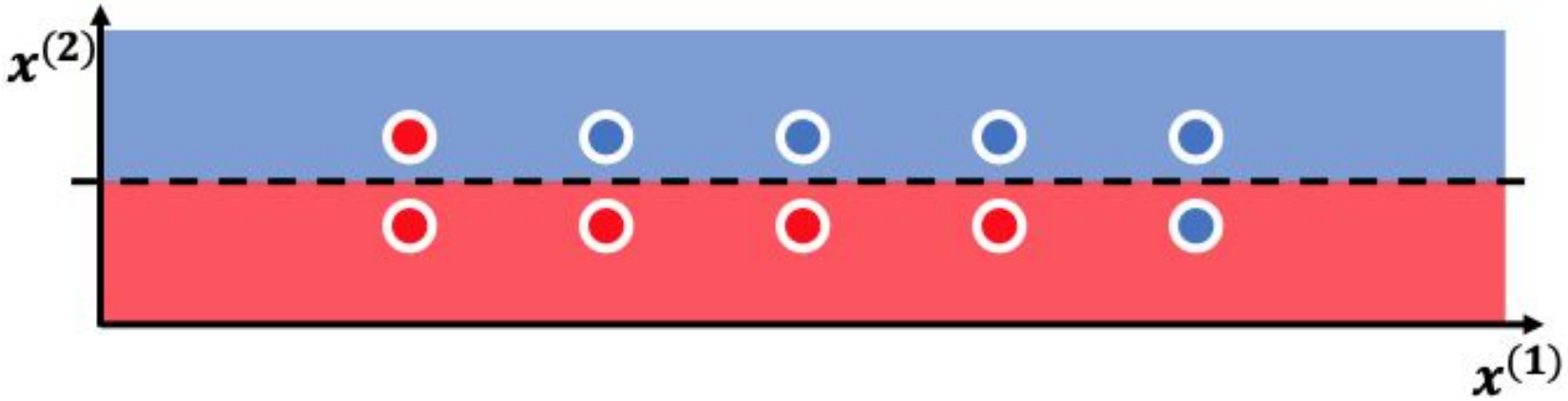
Which feature to split

Split threshold

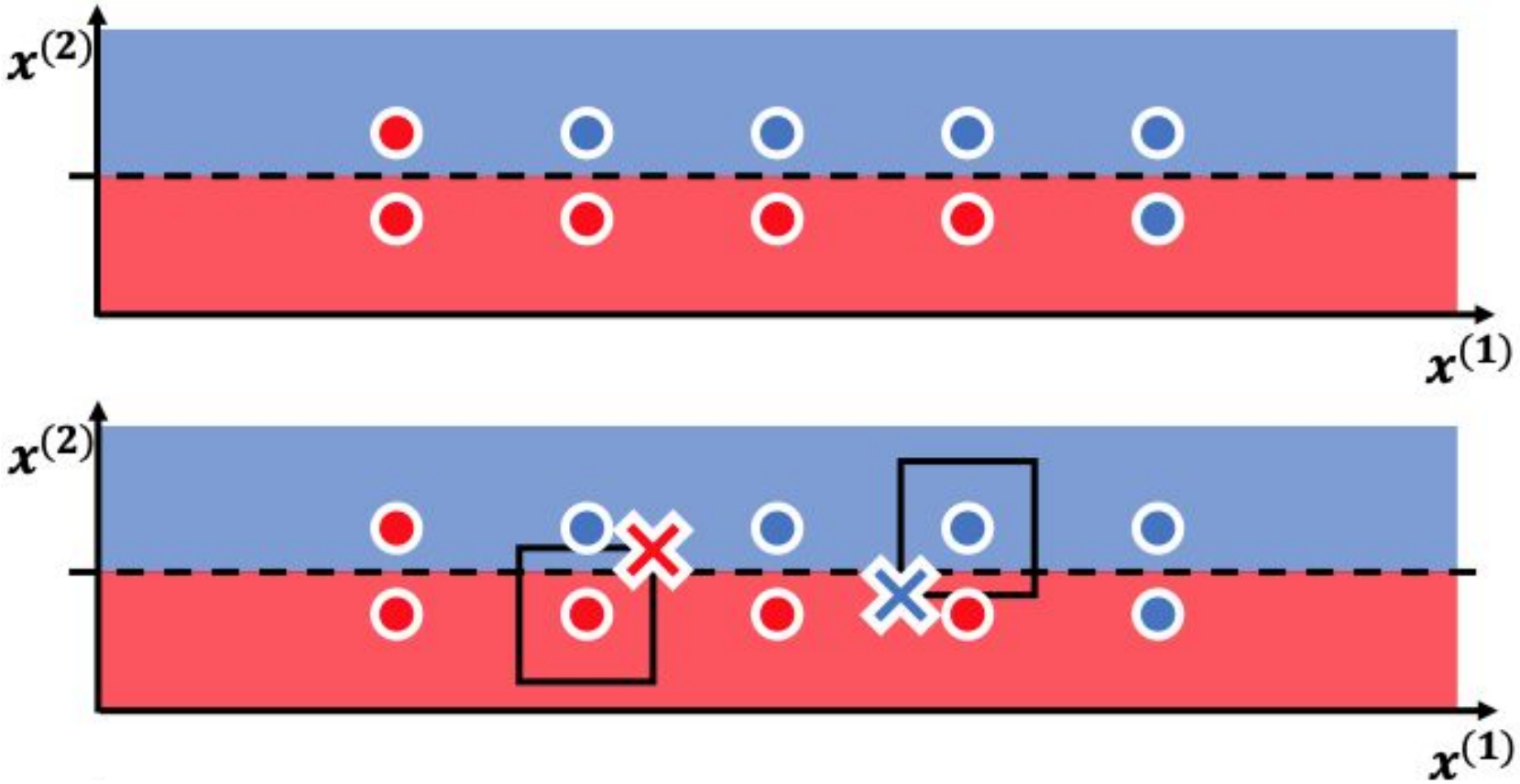
A score function

Points on the current node

Best accuracy ≠ Best robustness

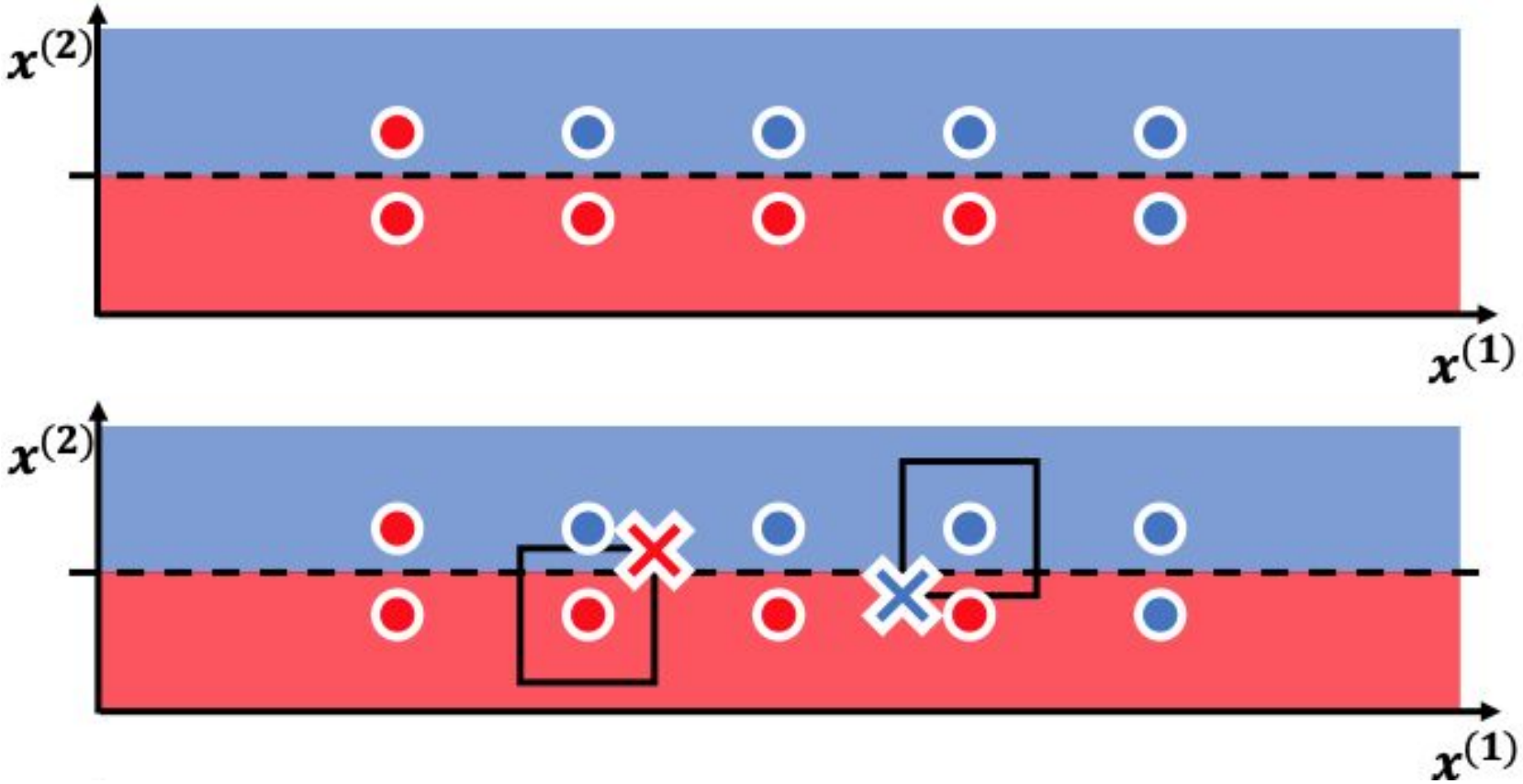


10 data points with two labels, a split on **feature 2 (horizontal)** gives an accuracy of **80%.**



All points are close to the decision boundary and they can be perturbed to any sides of the boundary.

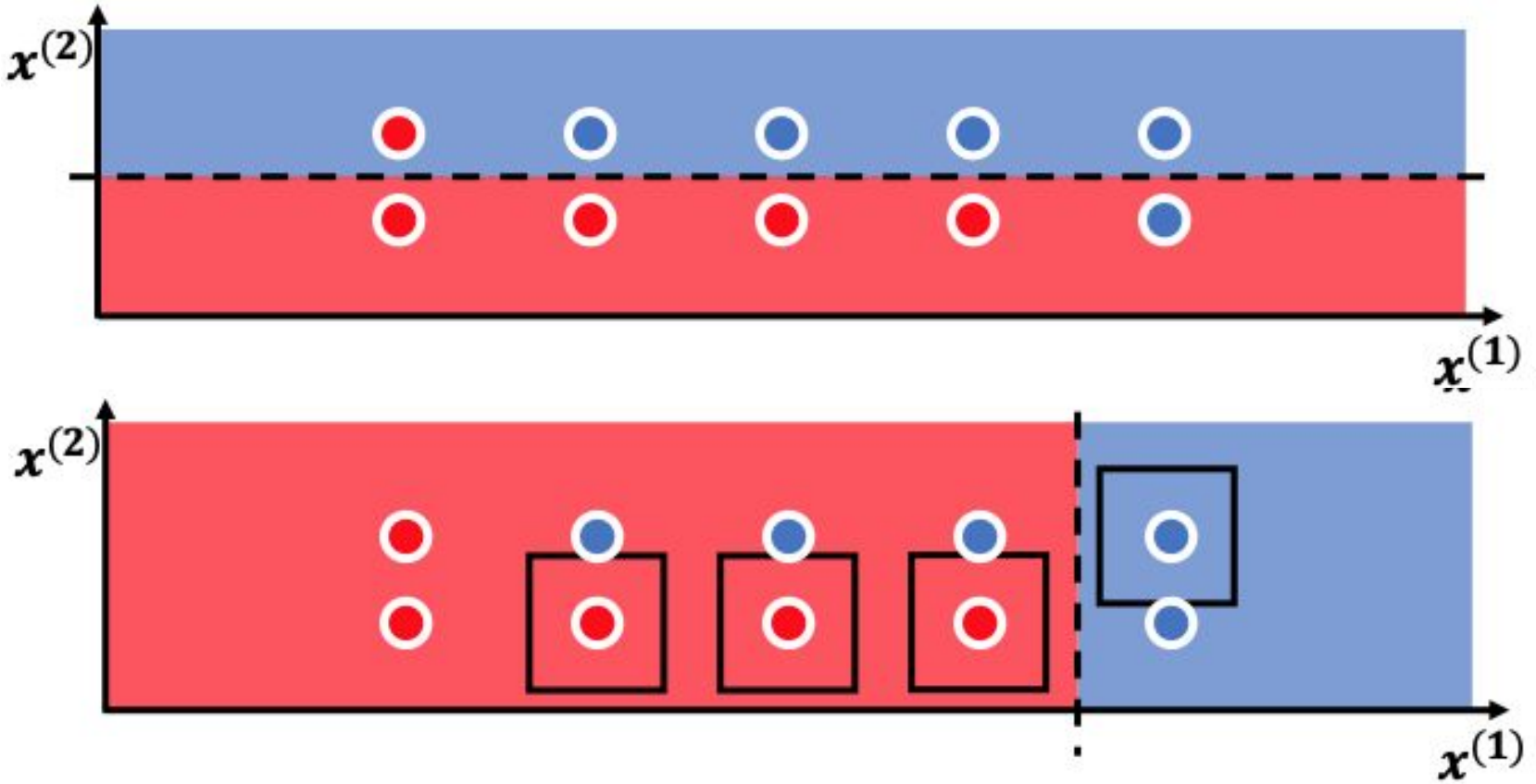
The worst case accuracy under perturbation is 0!



How to make it robust?

All points are close to the decision boundary and they can be perturbed to any sides of the boundary.

The worst case accuracy under perturbation is 0!



Choose another feature!

A better split would be on the **feature 1 (vertical)**, which **guarantees a 70% accuracy** under perturbations.

In the original (natural) decision tree training

$$j^*, \eta^* = \arg\max_{j,\ \eta} S(j,\ \eta,\ \mathcal{I})$$

Which feature to split

Split threshold

A score function

Points on the current node

Proposed robust decision tree training framework

$$j^*, \eta^* = \arg\max_{j,\,\eta} RS(j, \eta, \mathcal{I})$$

Robust Score function
(a maximin optimization function)

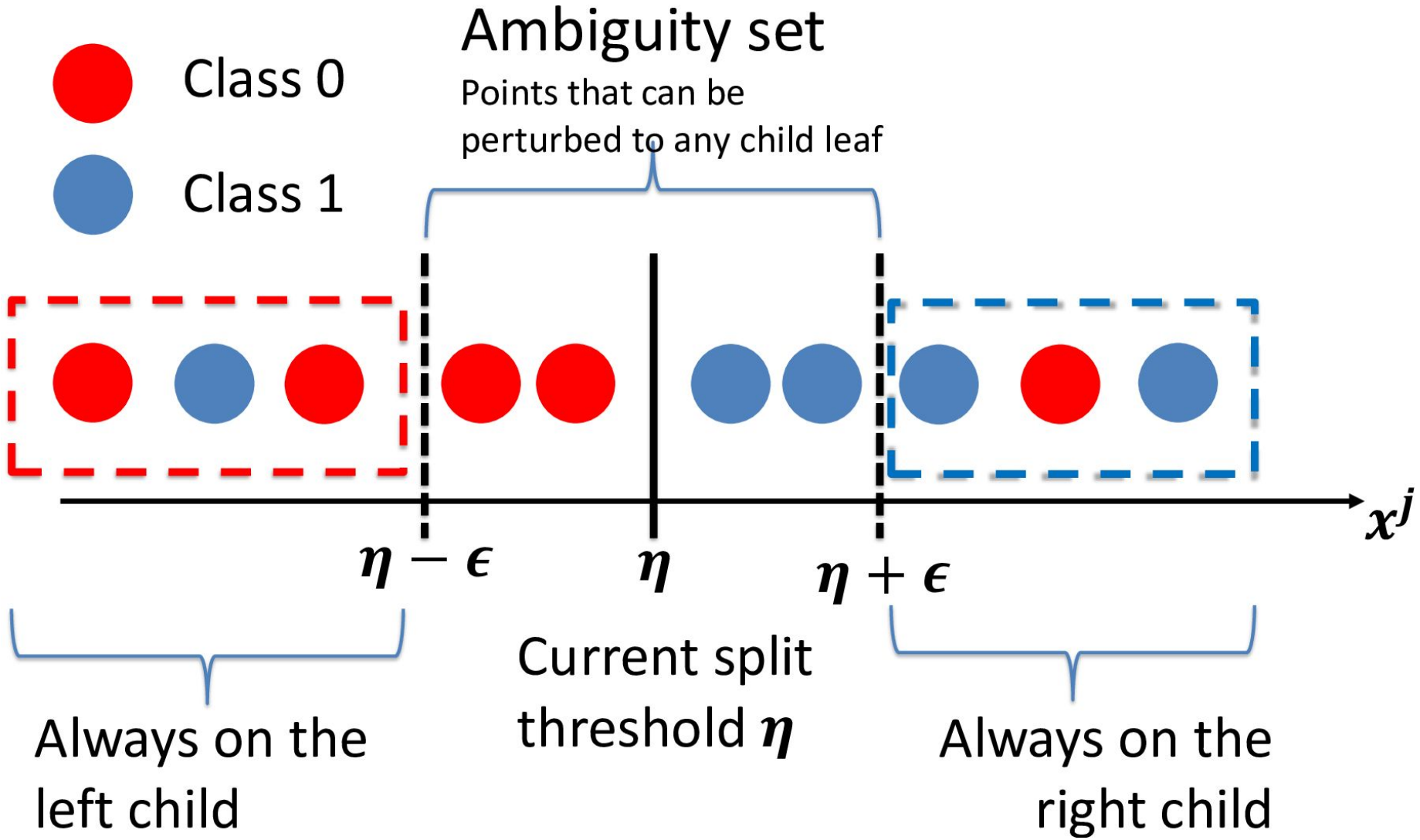
$$RS(j, \eta, \mathcal{I}) := \min_{\mathcal{I}'=\{(\mathbf{x}'_i,\, y_i)\}} S(j, \eta, \mathcal{I}')$$

$$\text{s.t. } \mathbf{x}'_i \in B_\epsilon^\infty(\mathbf{x}_i), \text{ for all } \mathbf{x}'_i \in \mathcal{I}'.$$

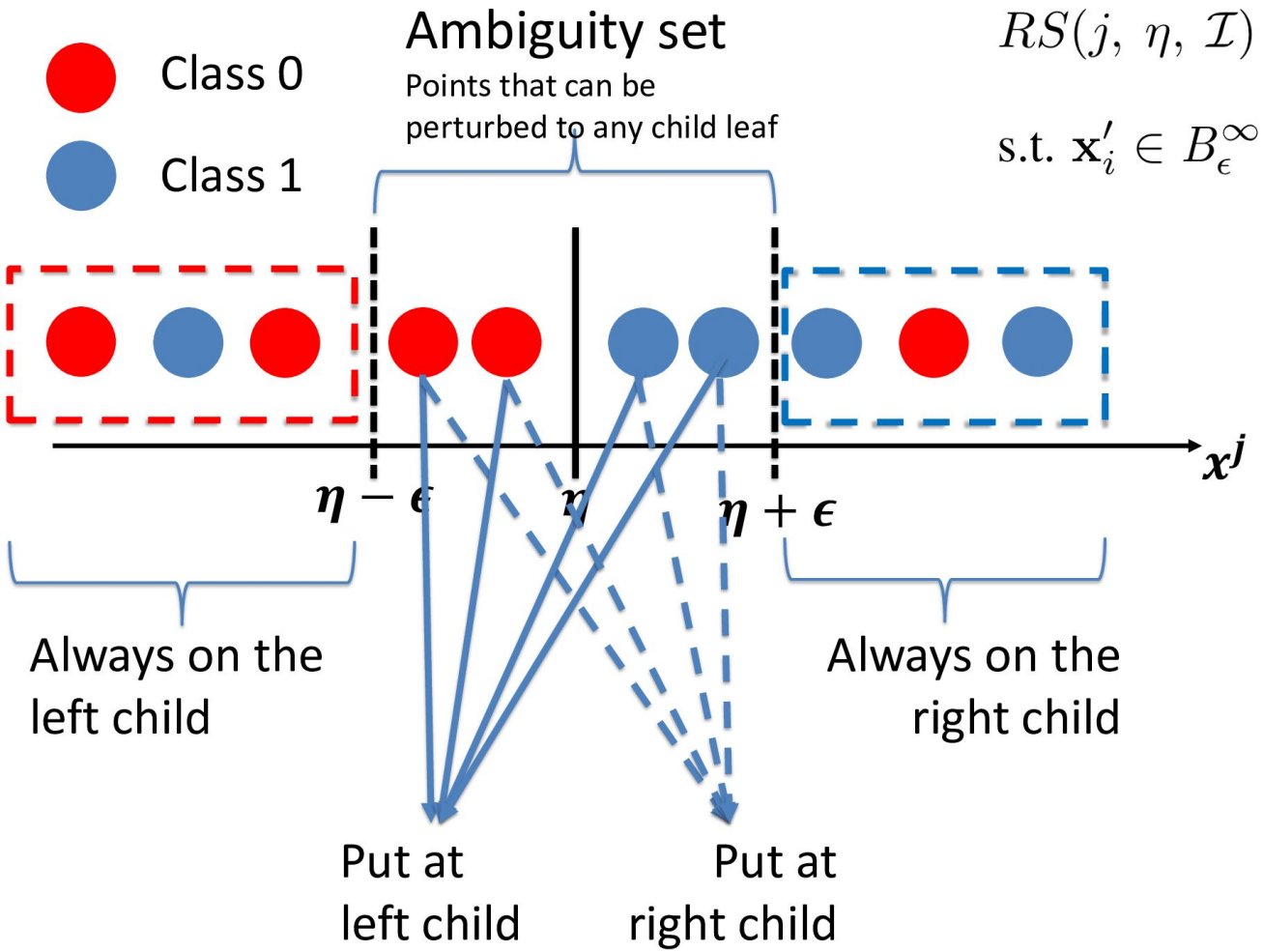
Worst case score

example x perturbed in an ℓ_∞ ball

It's actually a **1D** problem.



We need to optimize the worst case scenario.



$$RS(j,\ \eta,\ \mathcal{I}) := \min_{\mathcal{I}'=\{(\mathbf{x}_i',\ y_i)\}} S(j,\ \eta,\ \mathcal{I}')$$

$$\text{s.t. } \mathbf{x}_i' \in B_\epsilon^\infty(\mathbf{x}_i), \text{ for all } \mathbf{x}_i' \in \mathcal{I}'.$$

However there are exponentially many possibilities...

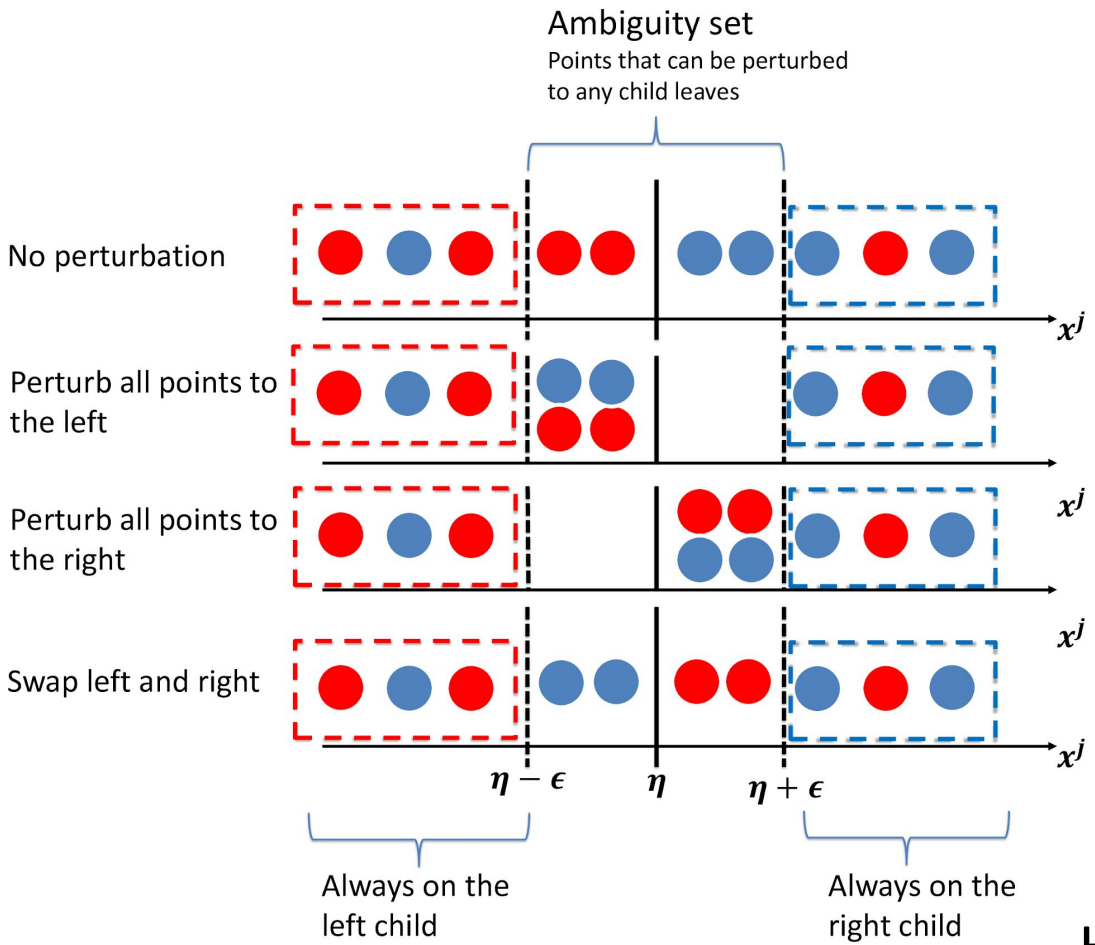
- For **Information Gain or Gini Impurity** scores, there is a **closed form** solution to approximate the optimal perturbation to minimize the score.

- For general scores, we need to solve a **0-1 integer minimization** to put each point in ambiguity set to left/right leaf, which can be very slow.

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma$$

XGBoost's score function

- Instead, we consider **4 representative cases** to approximate the **robust score**
- Does not increase the asymptotic complexity of the original decision tree training algorithm (only a constant factor slower)



$S(j,\ \eta,\ \mathcal{I}'_1)$

$S(j,\ \eta,\ \mathcal{I}'_2)$

$S(j,\ \eta,\ \mathcal{I}'_3)$

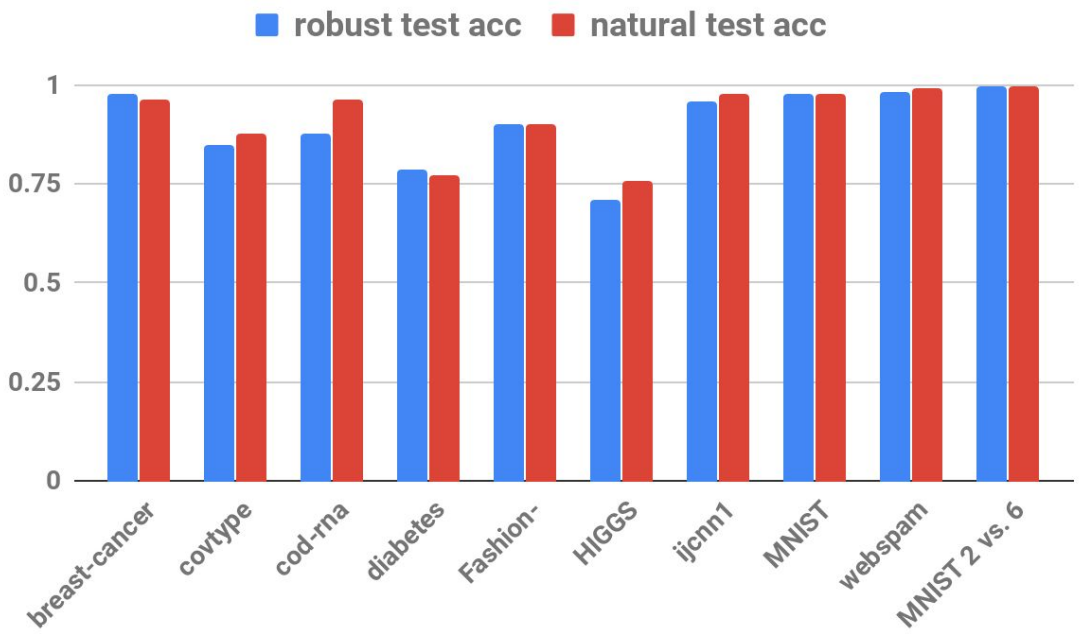
$S(j,\ \eta,\ \mathcal{I}'_4)$

$$RS(j,\ \eta,\ \mathcal{I}) \approx \min_{i\ =\{1,2,3,4\}} S(j,\ \eta, \mathcal{I}'_i)$$

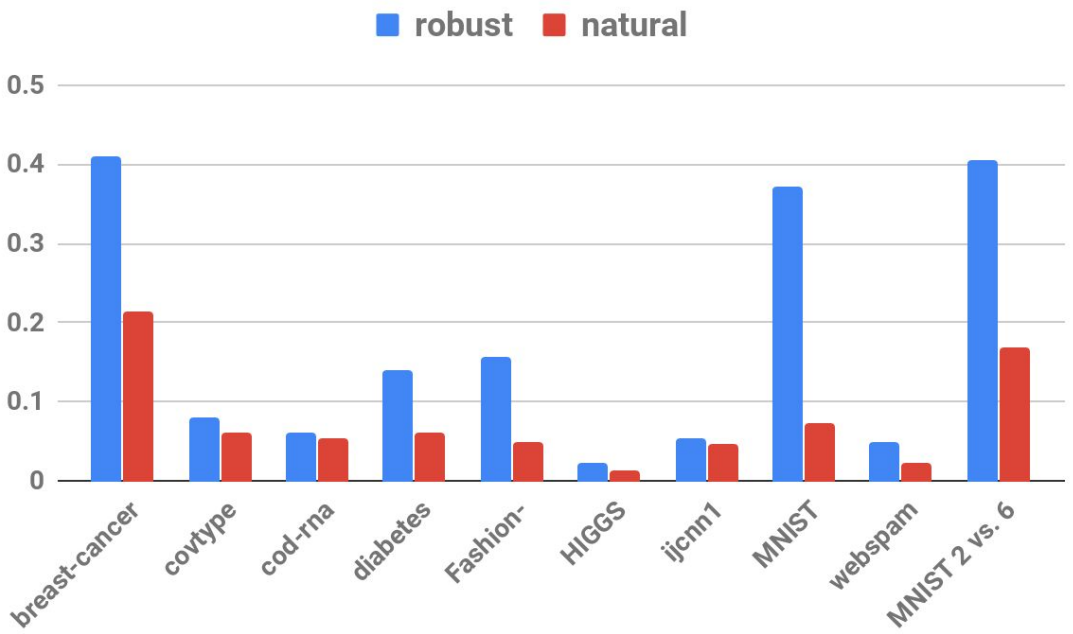
How well this approximation works?

Experiments

Test accuracy

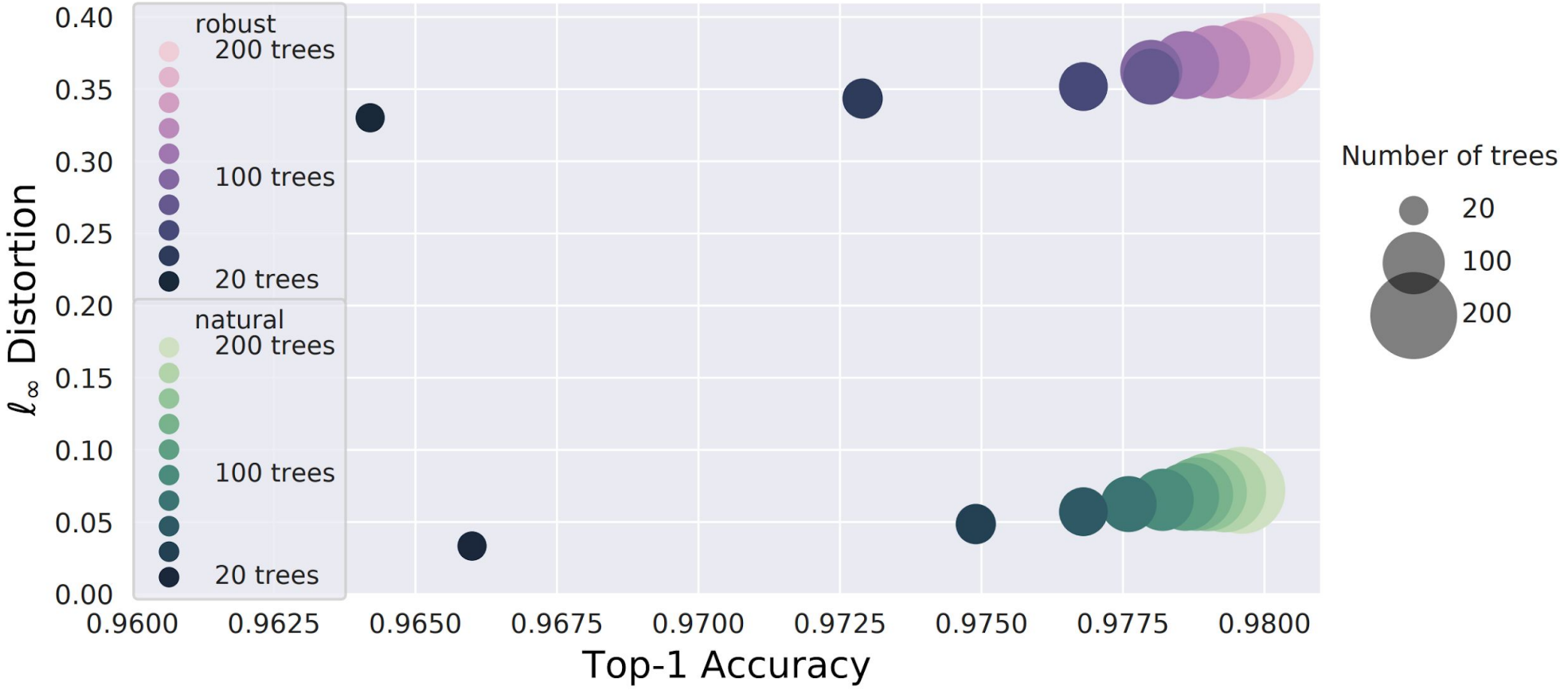


avg. ℓ_∞ norm of the adv. examples found by Cheng et al.'s attack



- Empirical results of robust and natural GBDT tree ensemble models on 10 datasets
- Using a general attack for non-smooth non-differentiable function (Cheng et al. ICLR 2019)
- **Remarkable robustness improvement on all datasets, without harming accuracy**

"Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach". Minhao Cheng, Thong Le, Pin-Yu Chen, **Huan Zhang**, Jinfeng Yi, Cho-Jui Hsieh. *ICLR 2019*



- MNIST models **with different number of trees** in GBDT
- Regardless the number of trees in the model, the **robustness improvement is consistently observed.**

	Original	natural model's adversarial example (ℓ_∞ distortion)	robust model's adversarial example (ℓ_∞ distortion)
MNIST		 (**0.074**)	 (**0.394**)
Fashion-MNIST		 (**0.069**)	 (**0.344**)

Robust Decision Trees Against Adversarial Examples. **Hongge Chen, Huan Zhang**, Duane Boning, Cho-Jui Hsieh. https://github.com/chenhongge/RobustTrees

- Does there exist a stronger attack?
- Can robustness be **formally verified**?

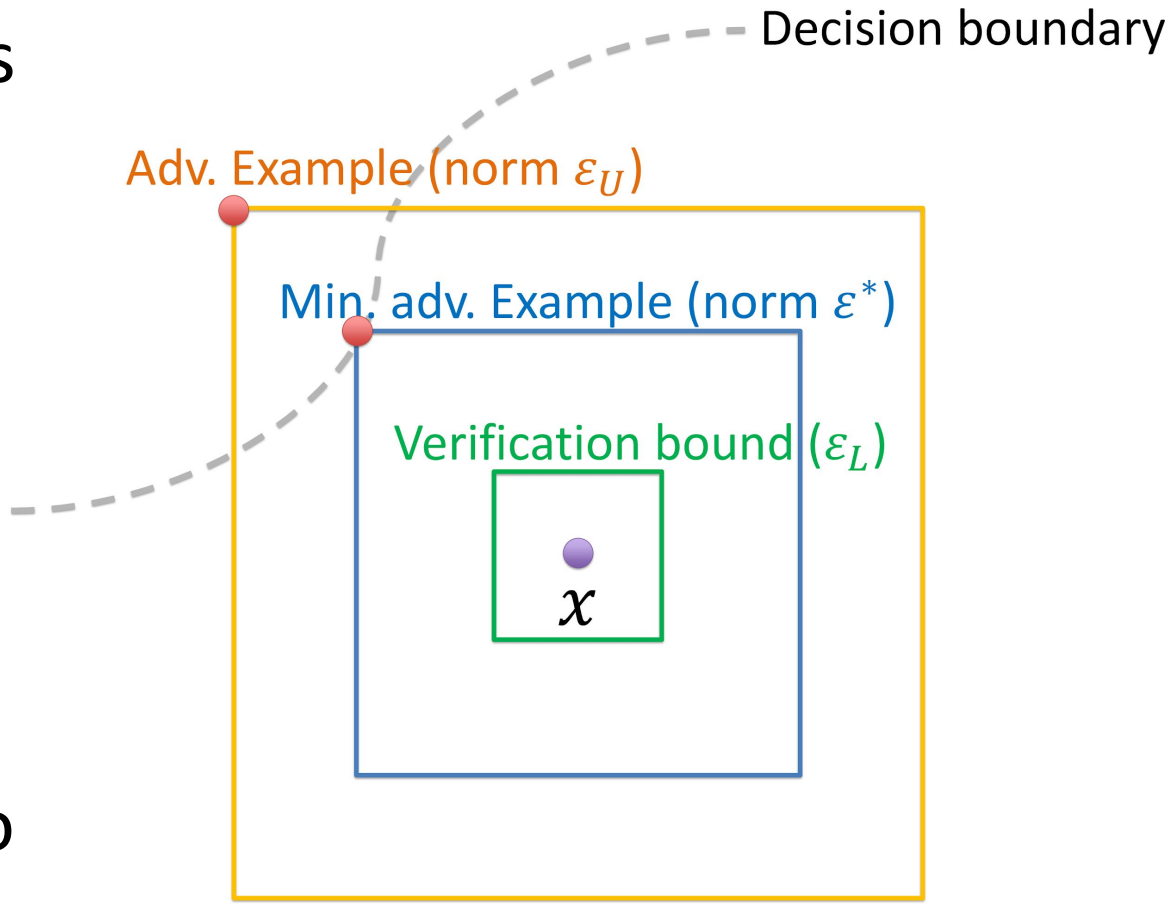
The robustness verification problem:

For a point x, a constant ε, and a classifier $f(\cdot)$, does there exists an x' such that $|x - x'|_\infty \leq \varepsilon$ and $f(x) \neq f(x')$?

For a point x, a constant ε, and a classifier $f(\cdot)$, does there exists an x' such that $|x - x'|_\infty \leq \varepsilon$ and $f(x) \neq f(x')$?

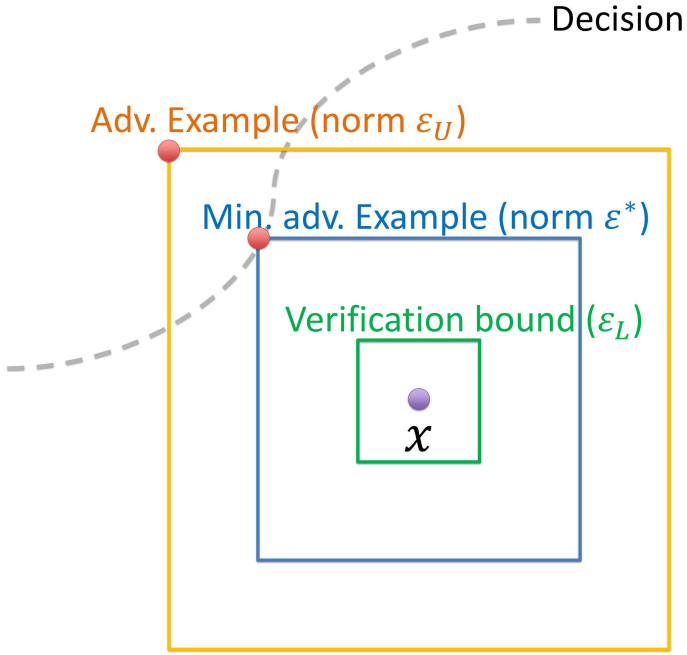
- **minimum adversarial distortion**: ε^* is the smallest ε such that an adversarial example exists (reflects **true robustness**)

- Attack algorithms find an upper bound ε_U of ε^*
- Verification algorithms find a lower bound ε_L of ε^* (can **guarantee** that no adversarial example exists if $\varepsilon < \varepsilon_L$)



- Finding the minimum adversarial distortion ε^* is **NP-complete** for general tree ensembles
- A Mixed Integer Linear Programming (**MILP**) based method was proposed by Kantchelian et al. (ICML 2016) and is not hopelessly slow

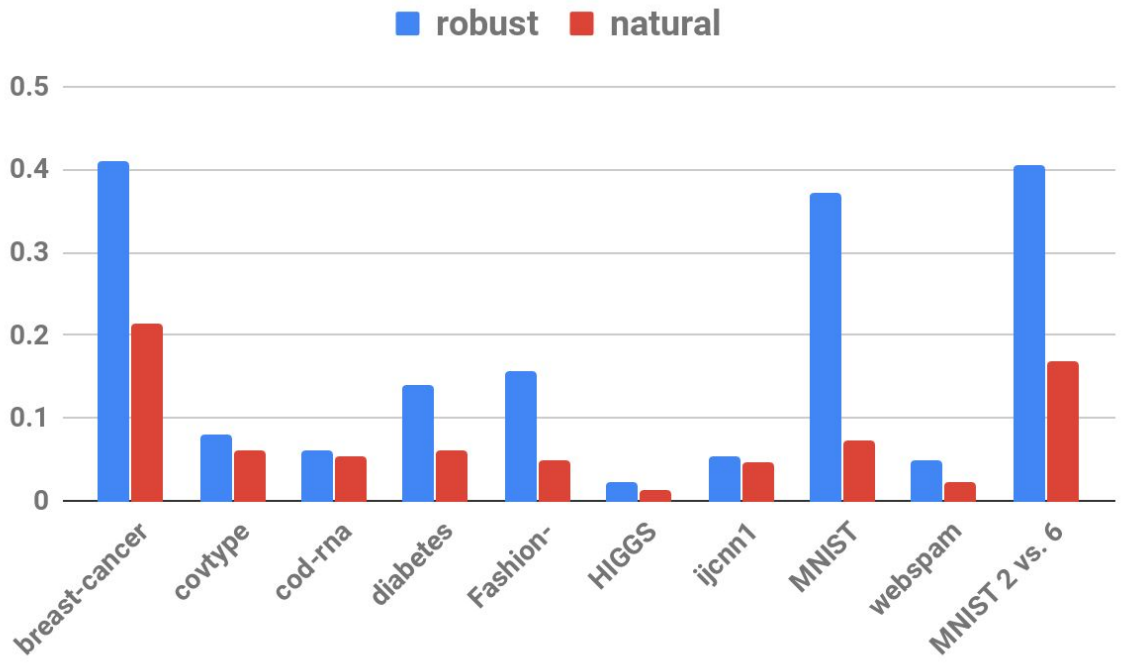
- MILP is the **strongest possible** attack (since it finds minimum ε^*)
- MILP gives robustness guarantee that **xists** with perturbation less than ε^*
- Finding ε^* is impractical for typical large neural networks (NNs are harder to verify)



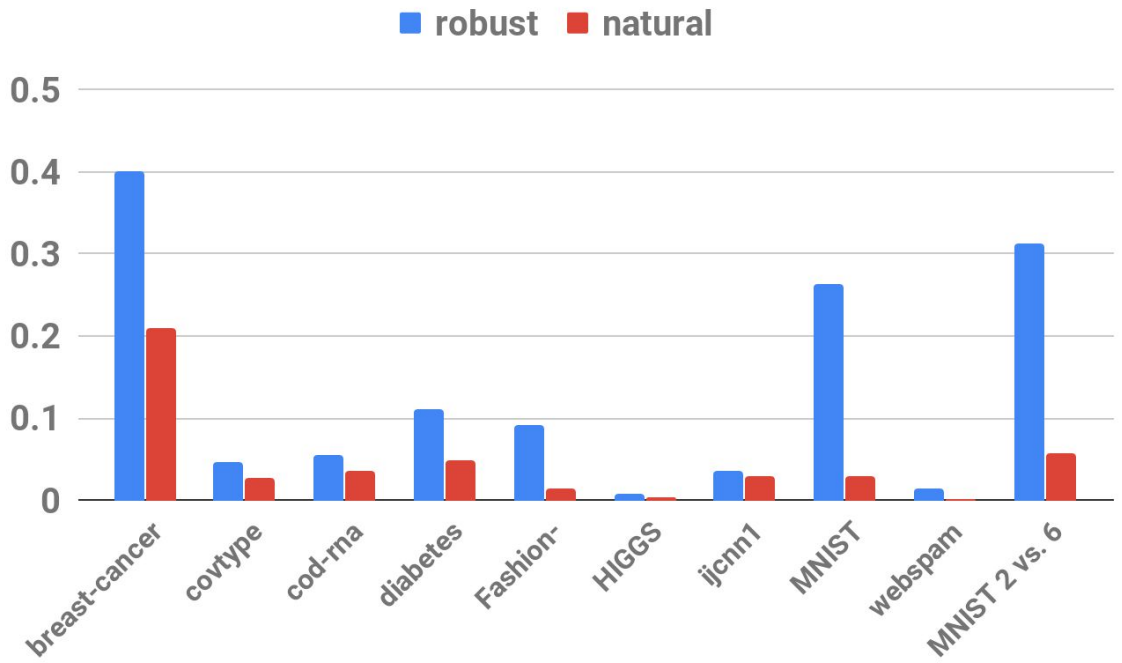
"Evasion and hardening of tree ensemble classifiers". Alex Kantchelian. J. D. Tygar, and Anthony Joseph. *ICML 2016*

Attack vs. MILP based verification

avg. ℓ_∞ norm of the adv. examples found by Cheng et al.'s black-box attack



avg. ℓ_∞ norm of the **minimum** adv. examples found by MILP



- The same trend can be observed!
- **Remarkable verifiable robustness improvement on all datasets**

MILP can still be slow (takes days/weeks to run) if the model or dataset is large!

We recently proposed an efficient and tight robustness verification bound for tree-based models.

Robustness Verification of Tree-based Models,

Hongge Chen*, Huan Zhang*, Si Si, Yang Li, Duane Boning, and Cho-Jui Hsieh

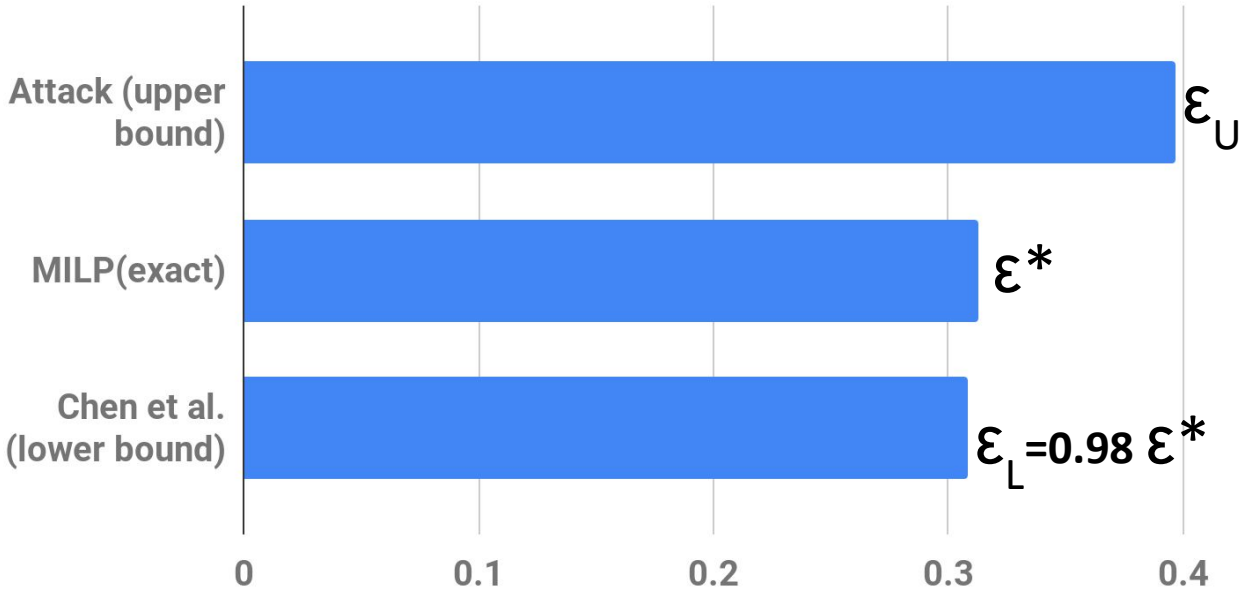
(*equal contribution)

https://arxiv.org/abs/1906.03849

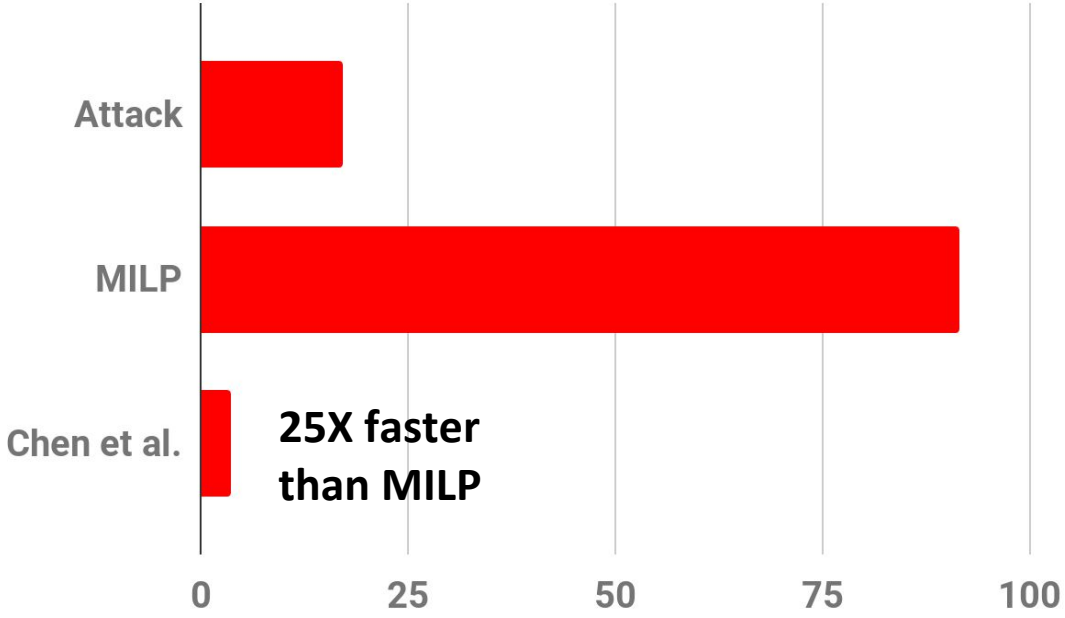
It's at the SPML workshop on Friday!

Average ℓ_∞ distortion and running time on a **1000-tree robust GBDT model** trained with MNIST 2 vs. 6 (a binary classification)

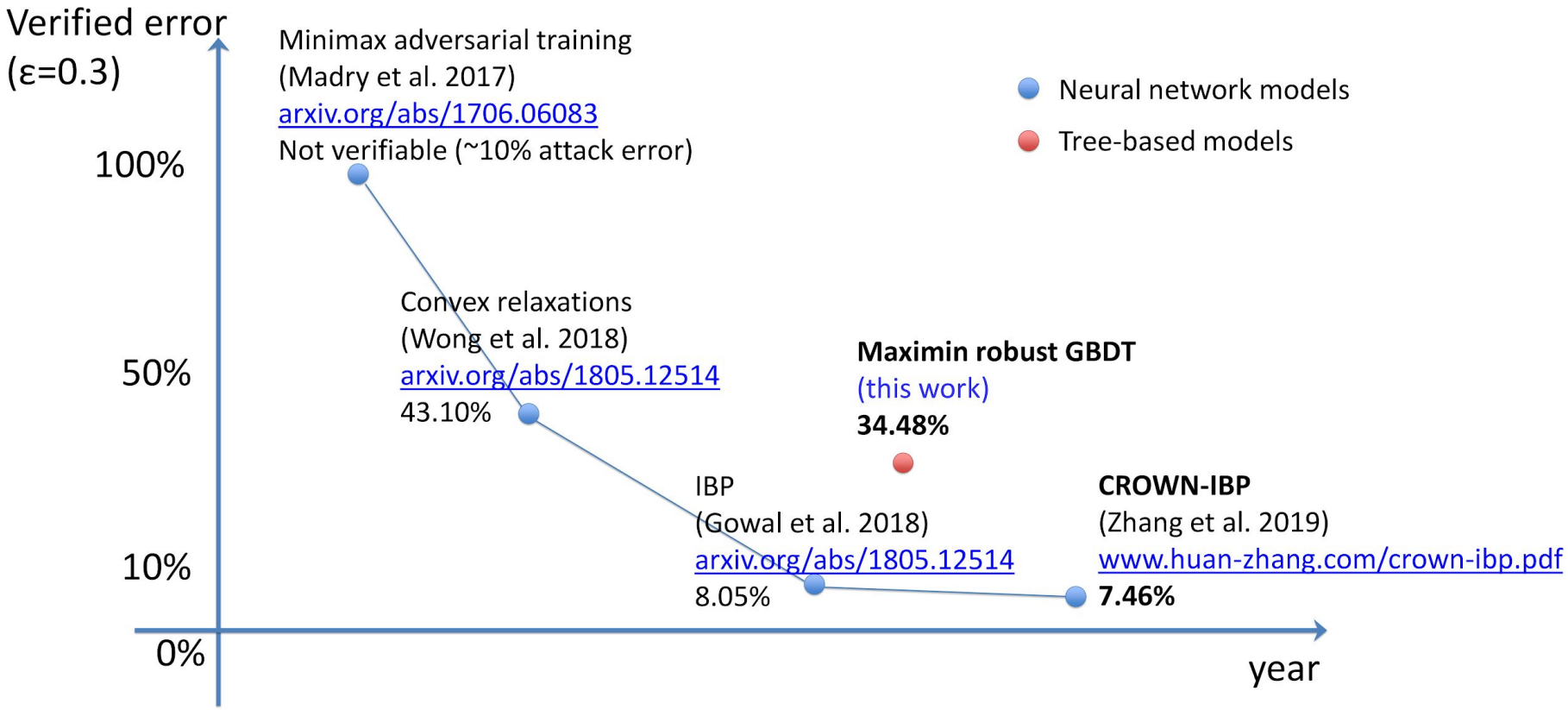
Average ℓ_∞ distortion



verification time per example



Comparing to DNNs: verified error on MNIST dataset with ε=0.3



- Unlike the minimax based adversarial training on deep training, our method uses a similar maximin robust optimization formulation but can be verified.
- Decision tree based models are **more verifiable** (fast and tight bounds exist)
- **Future work: how to further improve verified error of tree based ensembles?**

Conclusions

- Tree-based models are also vulnerable to adversarial examples
- Maximin robust optimization based training is effective on tree-based models
- Tree robustness can be more easily verified than DNNs

Thank You!

Code available at https://github.com/chenhongge/RobustTrees
Code is compatible with XGBoost and we plan to merge it into XGBoost upstream

Paper: https://arxiv.org/pdf/1902.10660.pdf

Checkout our new paper on **fast robustness verification of tree-based models**: https://arxiv.org/abs/1906.03849. It's also at the **SPML workshop** on Friday!