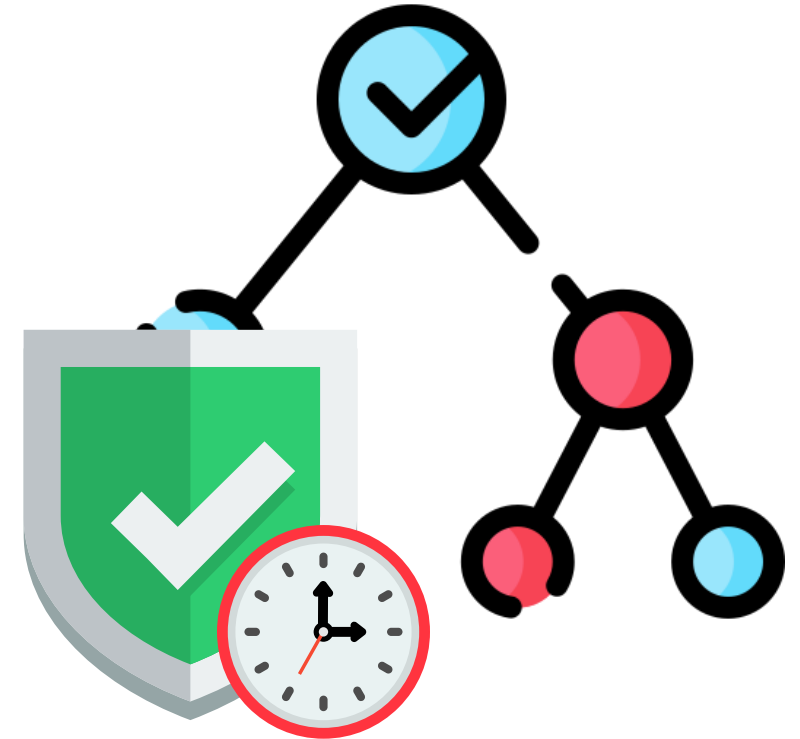




Università
Ca' Foscari
Venezia

Efficient and Principled Verification of Tree-Based Classifiers

Lorenzo Cazzaro (Università Ca' Foscari Venezia)
SAIL - Imperial College London, 29/01/2024



\$ whoami

I am a Ph.D. student in Computer Science under the supervision of prof. Stefano Calzavara at Università Ca' Foscari Venezia.

Main research interest: **Security of AI (and viceversa):**

- Design and verification of (security and fairness) properties of Machine Learning (ML) models.
- Adversarial attacks against ML.
- Using AI to improve the security of web applications.



Artificial Intelligence and Machine Learning

We can build a much brighter future where humans are relieved of menial work using AI capabilities.

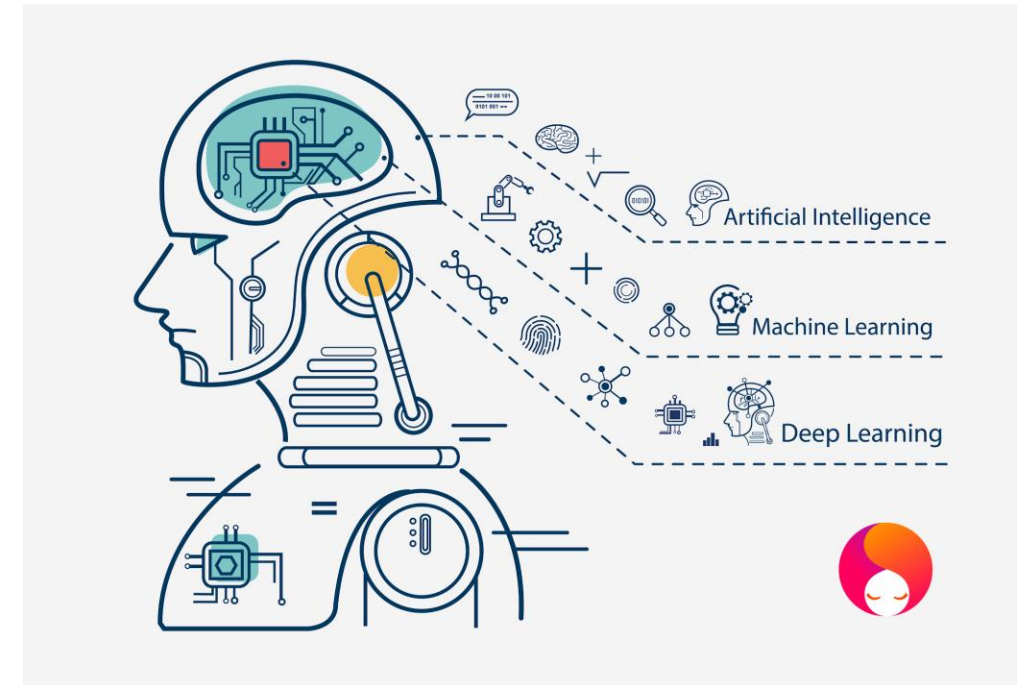
Andrew Ng., 2018

Applications:

- Image recognition (Google Lens)
- Natural language translation (DeepL)
- Recommender systems
- Malware detection
- ...

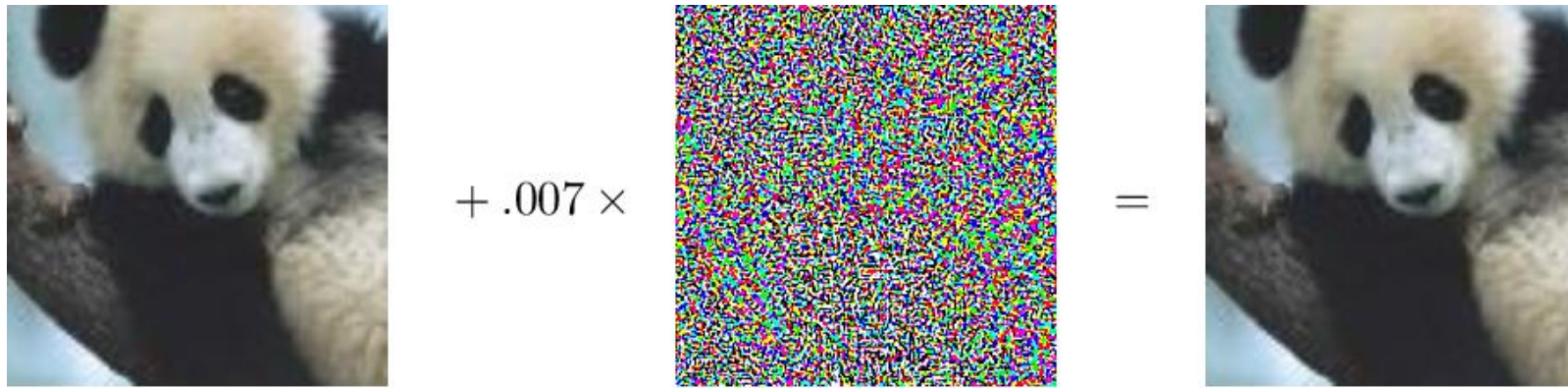
Given its pervasivity, AI (then, ML) must be **trustworthy!**

We are going to focus on ML in this talk.



Is Machine Learning Secure (Robust)?

Adversarial Examples are a serious threat to ML robustness...



“panda”
57.7% confidence

“nematode”
8.2% confidence

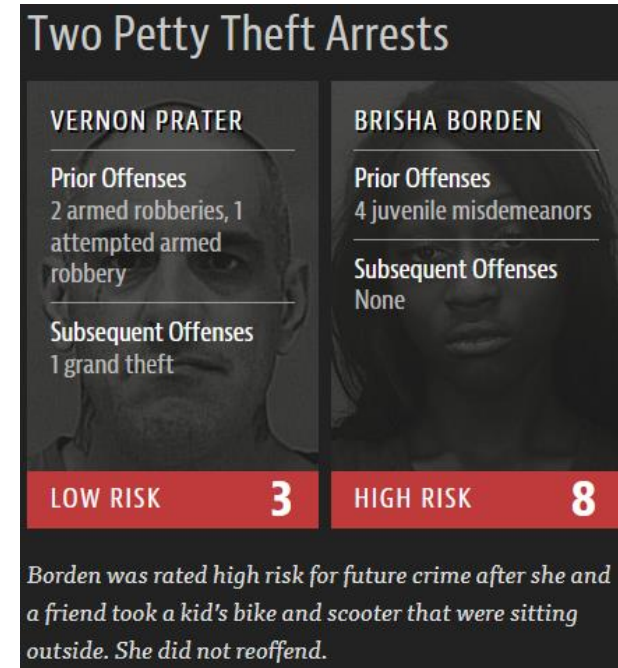
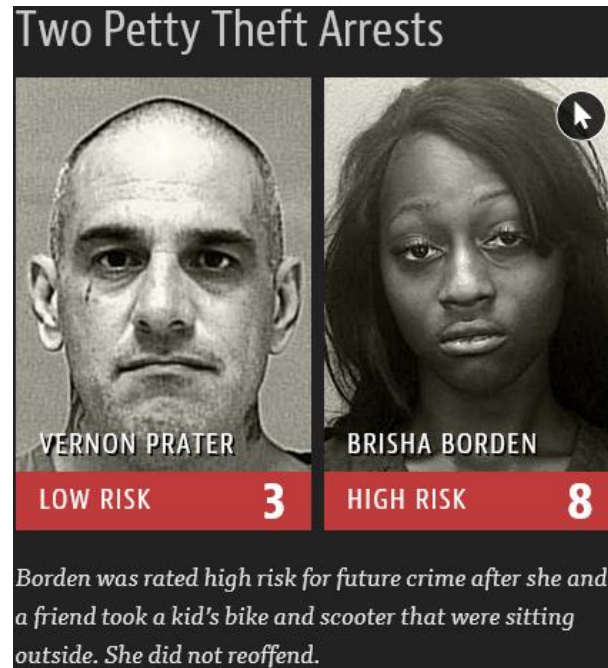
“gibbon”
99.3 % confidence

Credits: Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In ICLR (2015)

They can be generated also in other domains than computer vision, e.g., malware detection.

Is Machine Learning Fair?

Example: Machine Learning (ML) used to predict recidivism in USA*



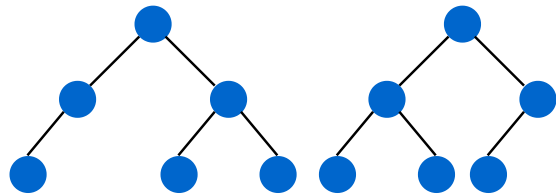
Non-recidivist black people were twice as likely to be labelled high risk than non-recidivist white people.

*<https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>

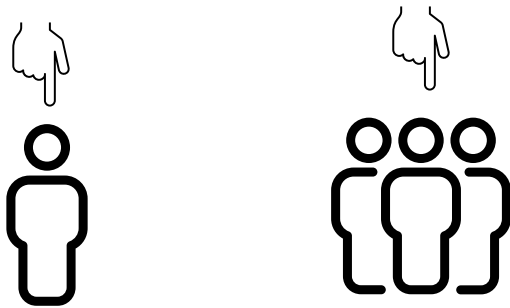
The Need For (Expressive) Properties

We need to describe the trustworthy behaviour of a ML model by defining some **properties**.

Local Properties

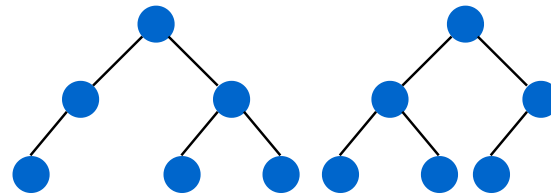


It's robust/fair on



Test set

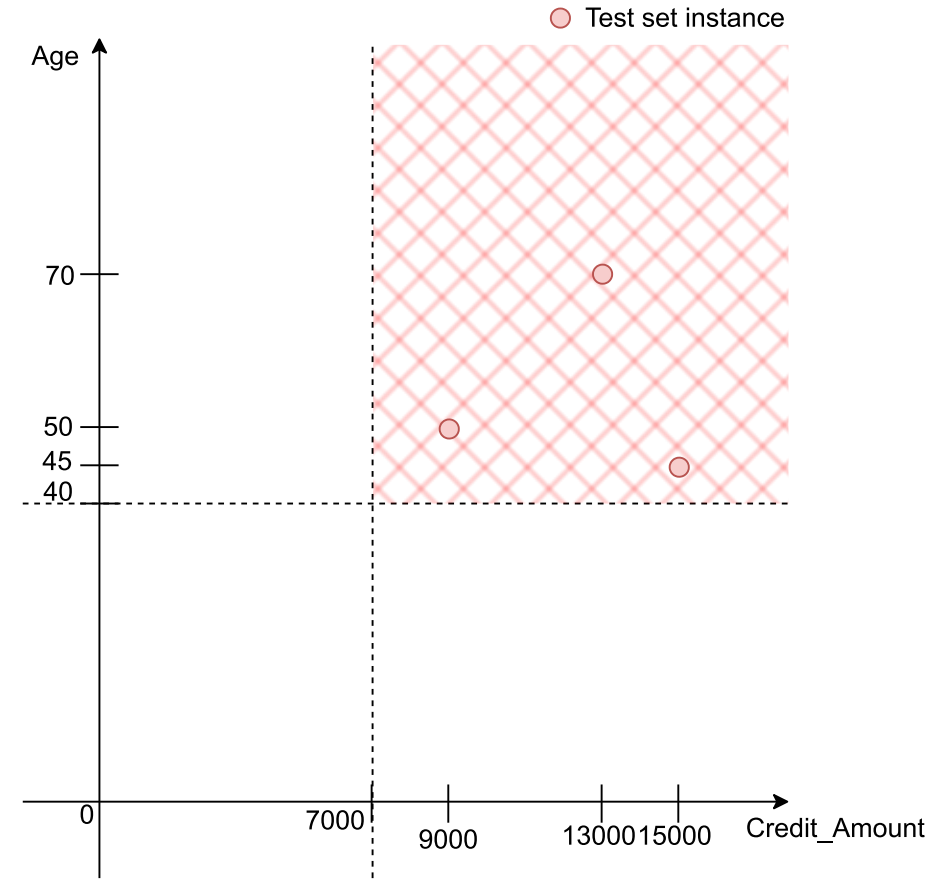
Global / Data-Independent Properties



It is robust/fair on people described by

Age ≥ 40
and
Credit_Amount ≥ 7000

Potentially continuous and unbounded subset of instances!

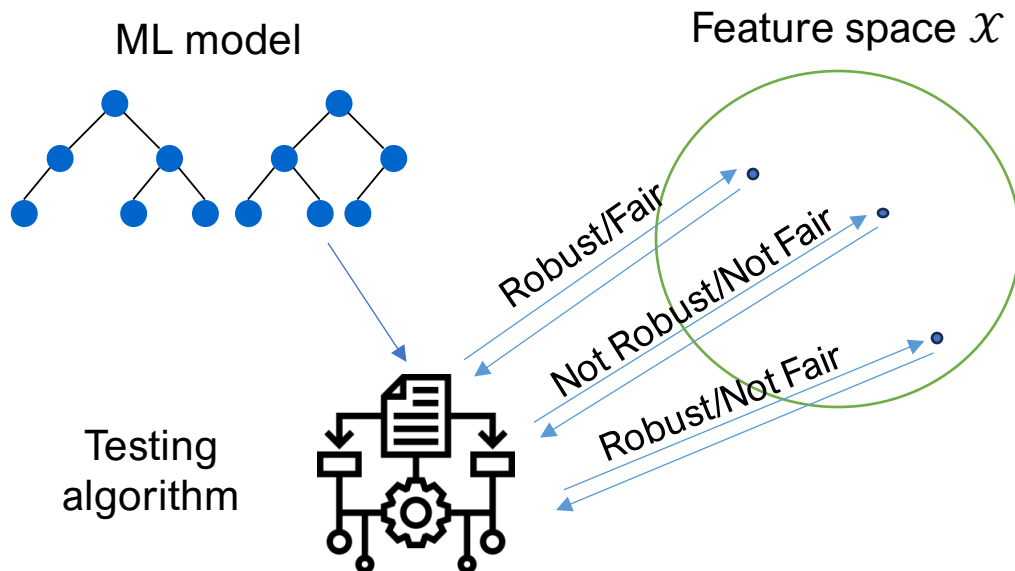


Is the considered property sufficient to describe the desired behaviour of the ML model?

The Need For (Efficient) Formal Verification.

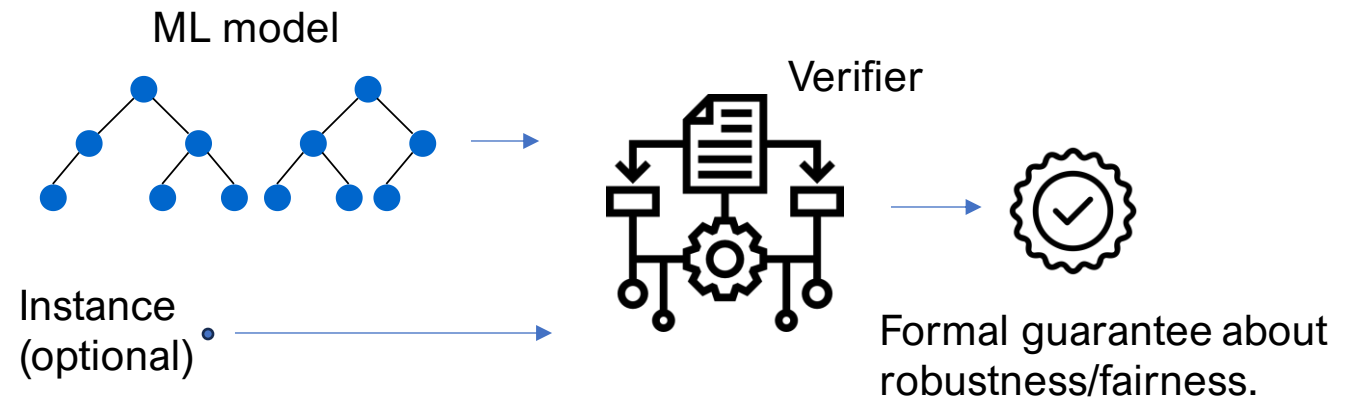
How can we prove the security and/or fairness of ML models?

Empirical Approach / Testing



- **Pros: efficient.**
- **Question: is it sufficient to prove the property of interest?**

Formal Approach



- **Pros: formal guarantees in output + it can cover more properties than the empirical approach.**
- **Questions: Soundness? Completeness? Scalability?**
- **Complete verification may require solving NP-hard problems, so efficiency may be sacrificed!**

Talk Outline

We will see:

1. How to reduce the time complexity of (security) verification of ML models, particularly decision tree ensembles, by rethinking training algorithms.
2. An introduction to the shortcomings of the (local) robustness property for ML models, its data-independent generalization, called *resilience*, and a sound algorithmic way to prove it.
3. An introduction to how fairness testing approaches fail to verify the lack of causal discrimination and how to verify this property by giving explainable formal guarantees in output.

Thanks to: Stefano Calzavara, Claudio Lucchese, Federico Marcuzzi, Salvatore Orlando, Nicola Prezza, Giulio Ermanno Pibiri.

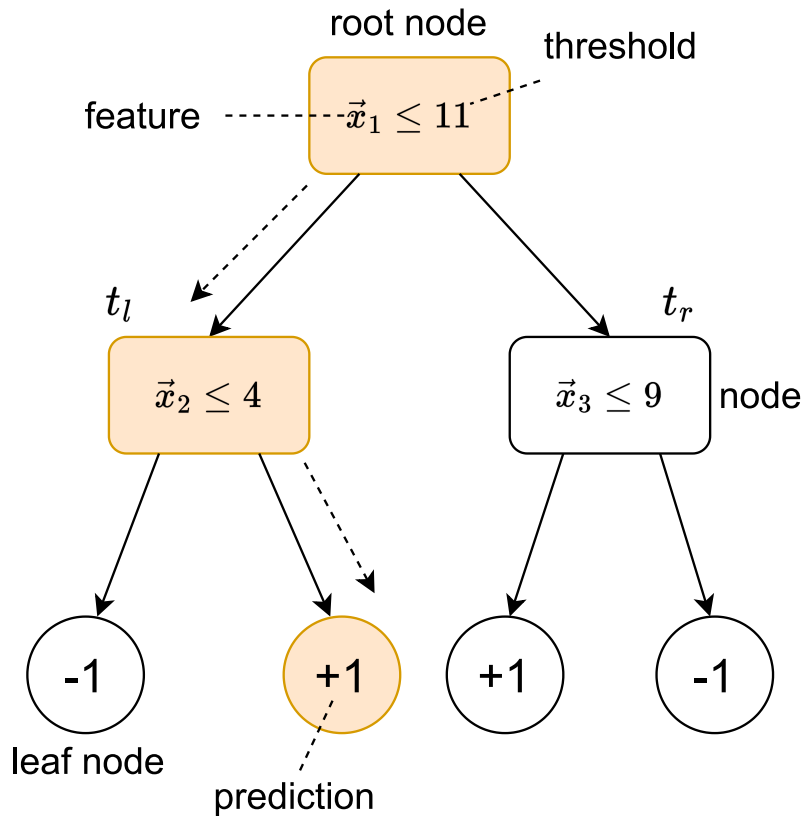
Short Background

Tree-Based Classifiers

Decision Tree Classifier t

$$\vec{x} = \langle 10.5, 4.5, 17 \rangle$$

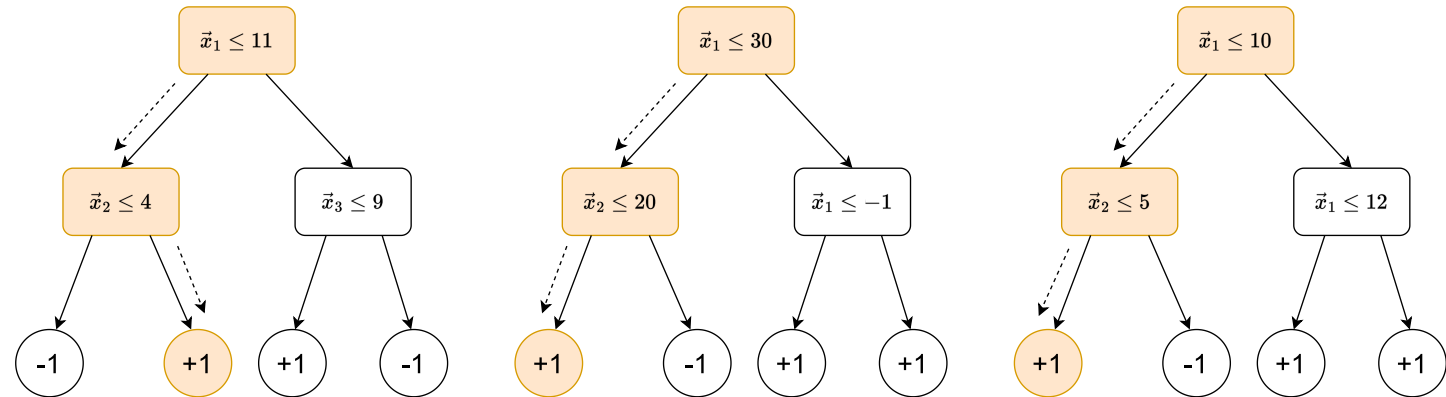
$$y = +1$$



Decision Tree Ensemble $T = \{t_1, t_2, \dots, t_n\}$

$$\vec{x} = \langle 10.5, 4.5, 17 \rangle$$

$$y = +1$$



Ensemble prediction \rightarrow aggregation of the predictions of the single trees.

We consider **majority voting** as aggregation scheme (used by Random Forests).

Making Robustness Verification Efficient with Verifiable Learning

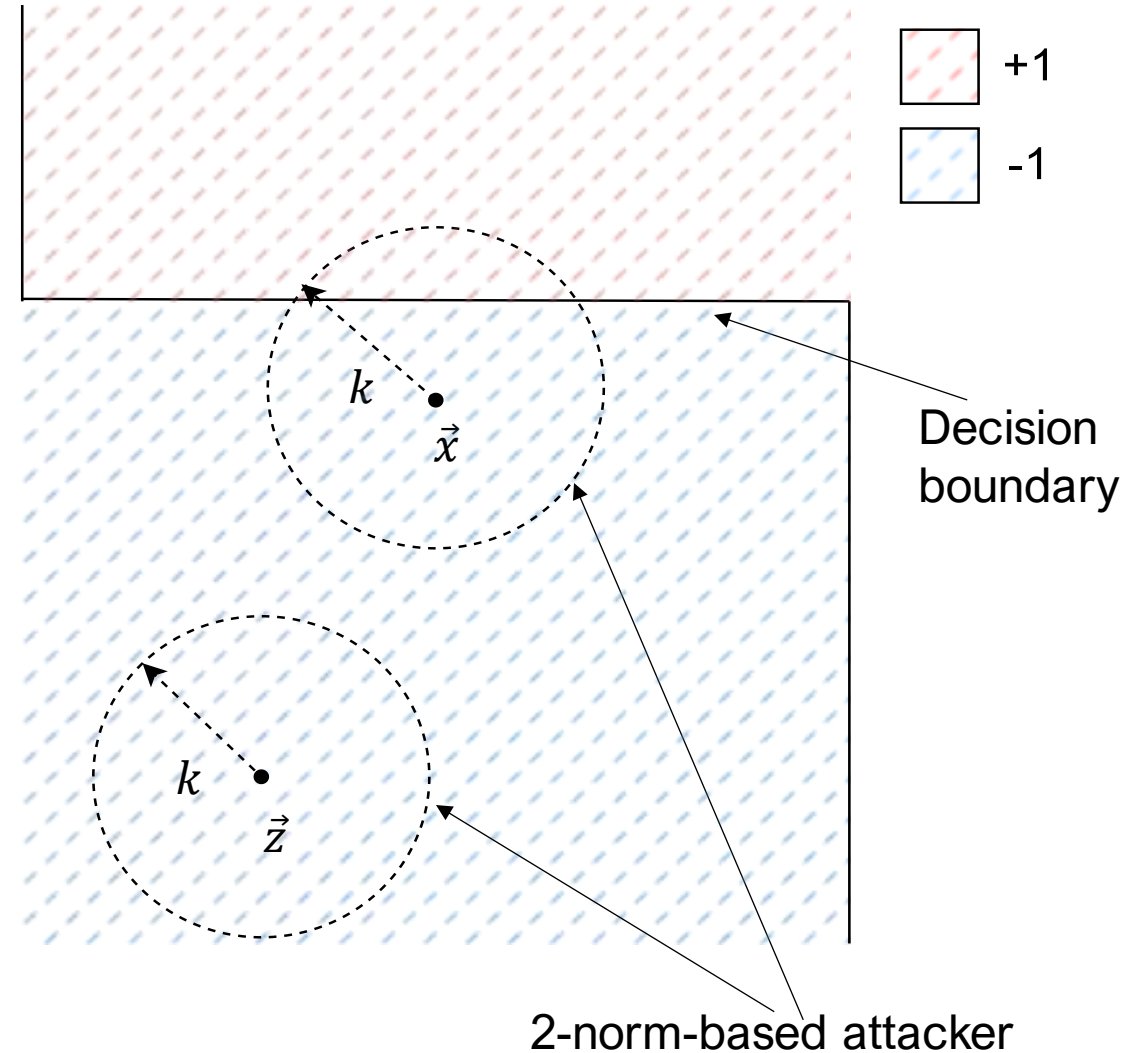
Calzavara S., Cazzaro L., Pibiri G.E., Prezza N. – *Verifiable Learning for Robust Tree Ensembles*, in ACM Conference on Computer and Communications Security 2023 (CCS 2023).

Robustness Verification

Machine Learning (ML) models are vulnerable to **evasion attacks** at test time!

Robustness is estimated as the accuracy under the p -norm-based attacker with maximum perturbation k .

Robustness verification is a well-studied problem both for neural networks and other models like tree ensembles.



Robustness Verification is hard!

Complete robustness verification is **hard** for tree ensembles*!

Existing robustness verification algorithms do not always terminate → give lower and upper bounds for the actual value of robustness, not the precise value.

Complexity of robustness verification for p -norm-based attackers.

Model	Complexity
Decision tree	Linear
Tree ensemble	NP-complete

These analyses are **worst case**. Can we find a **restricted class** of tree ensembles enabling efficient security verification against any norm-based attackers?

*Yihan Wang, Huan Zhang, Hongge Chen, Duane S. Boning, and Cho-Jui Hsieh. 2020. On Lp-norm Robustness of Ensemble Decision Stumps and Trees. In ICML

Contribution: Verifiable Learning

We propose ***Verifiable Learning***: rethink training algorithms in order to make the (robustness) verification of the trained model more efficient (also formally).

We instantiate Verifiable Learning for **decision tree ensembles**.

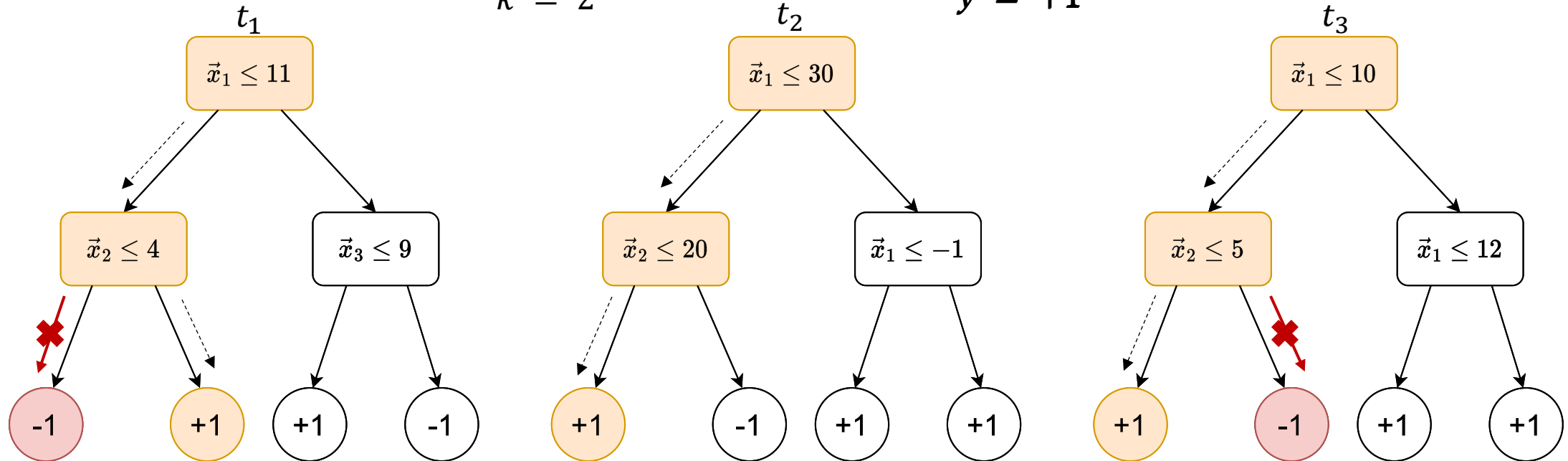
Our contribution consists of 5 parts:

1. We identify what makes the verification problem NP-complete.
2. We restrict the shape of the model in order to avoid the source of the high complexity.
3. We design a (formally proven) efficient verification algorithm for the class of restricted models.
4. We design an (efficient) training algorithm for the class of restricted models.
5. We experimentally verify the effectiveness of our proposal.

Robustness verification of tree ensembles

1-norm-based attacker
 $k = 2$

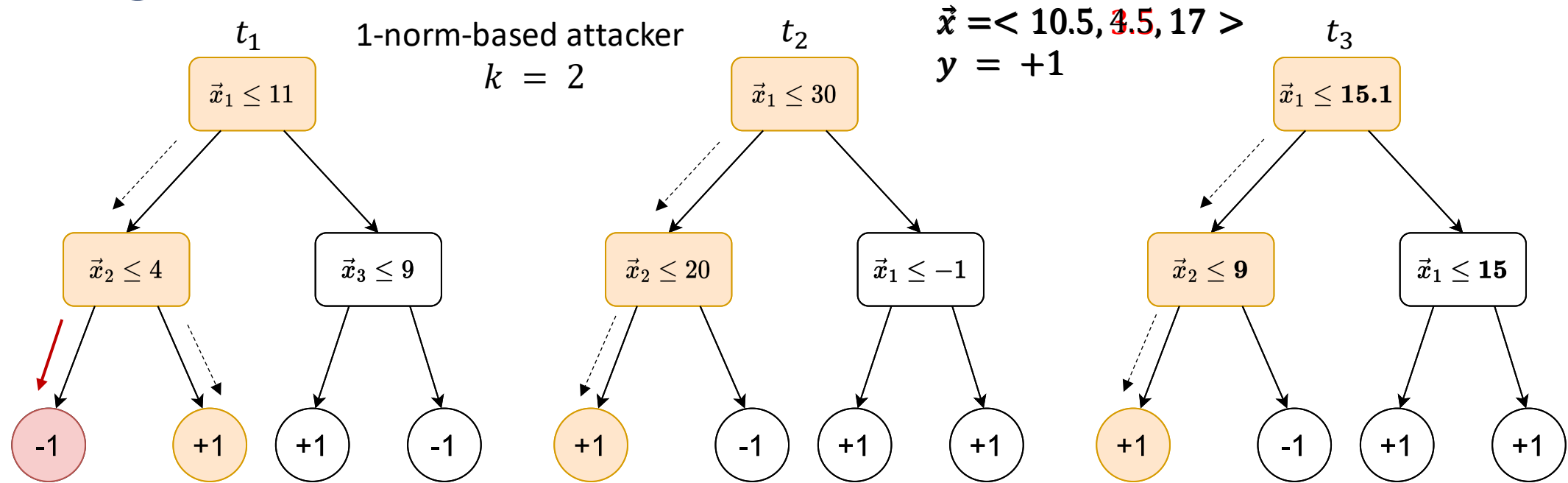
$\vec{x} = \langle 10.5, 4.5, 17 \rangle$
 $y = +1$



Problem: even though it is efficient to verify the robustness of a decision tree, it is not possible to compose the results to make the verification efficient for ensembles.

Step to the solution: if the structure of trees makes **only compatible attacks feasible**, we can **compose the attacks** on the single trees in an efficient way.

Large-spread ensembles



Large-spread condition: any two thresholds for the same feature occurring in two different trees are at a distance of at least $2k$, where k is the maximum adversarial perturbation.

Intuition: if thresholds are sufficiently far away, attacks on different trees **cannot interfere** with each other and can be composed.

Key formal result: if the ensemble is large-spread, then attacks operating against different trees of the ensemble are orthogonal \rightarrow they can be added to produce an attack working against all such trees!

Efficient robustness verification

Our verifier CARVE* (suppose that the large-spread ensemble contains m trees):

1. Analyze the m individual trees of the ensemble, using the existing linear time algorithm.
2. If less than $\frac{m}{2} + 1$ trees can be attacked, then no attack on the ensemble is possible (since the aggregation scheme is majority voting).
3. Otherwise, find the $\frac{m}{2} + 1$ trees with the attacks of minimum perturbation: an attack on the ensemble is possible if and only if the sum of these attacks does not exceed the maximum adversarial perturbation k .

Theorem: robustness can be verified in polynomial time for large-spread tree ensembles for any norm-based attackers.

*CARVE - CompositionAI Robustness Verifier for tree Ensembles

Training large-spread ensembles with LSE

The training algorithm LSE* is based on mutation and pruning:

1. Train a traditional forest T of $\gg m$ trees and initialize the large-spread ensemble E with a random tree from T .
2. Iterate for $m - 1$ rounds:
 - A. Pick the tree t in T that minimizes the overlaps with E .
 - B. Fix the overlaps of t with E by perturbing the thresholds of t and E that overlap (mutation).
 - C. Extend E with t (if all the overlaps have been fixed).
3. Return E (m trees out of $\gg m$ the trees in T if LSE succeeds in building the entire large-spread ensemble \rightarrow pruning).

*LSE - Large-Spread Ensemble

Experimental Evaluation

We implemented our verifier CARVE in C++ and our training algorithm LSE in Python (both publicly available on Github!).

Research questions:

1. Can we train a large-spread ensemble with the proposed algorithm?
2. What are the accuracy and the robustness of large-spread ensembles?
3. What is the benefit of the large-spread condition in terms of verification time and memory consumption over a state-of-the-art complete verifier (SILVA*)?

*Francesco Ranzato and Marco Zanella. 2020. Abstract Interpretation of Decision Tree Ensemble Classifiers. In AACL.

Performance of Large-Spread Ensembles

Verified using SILVA

Verified using CARVE

We are using an ∞ -norm-based-attacker

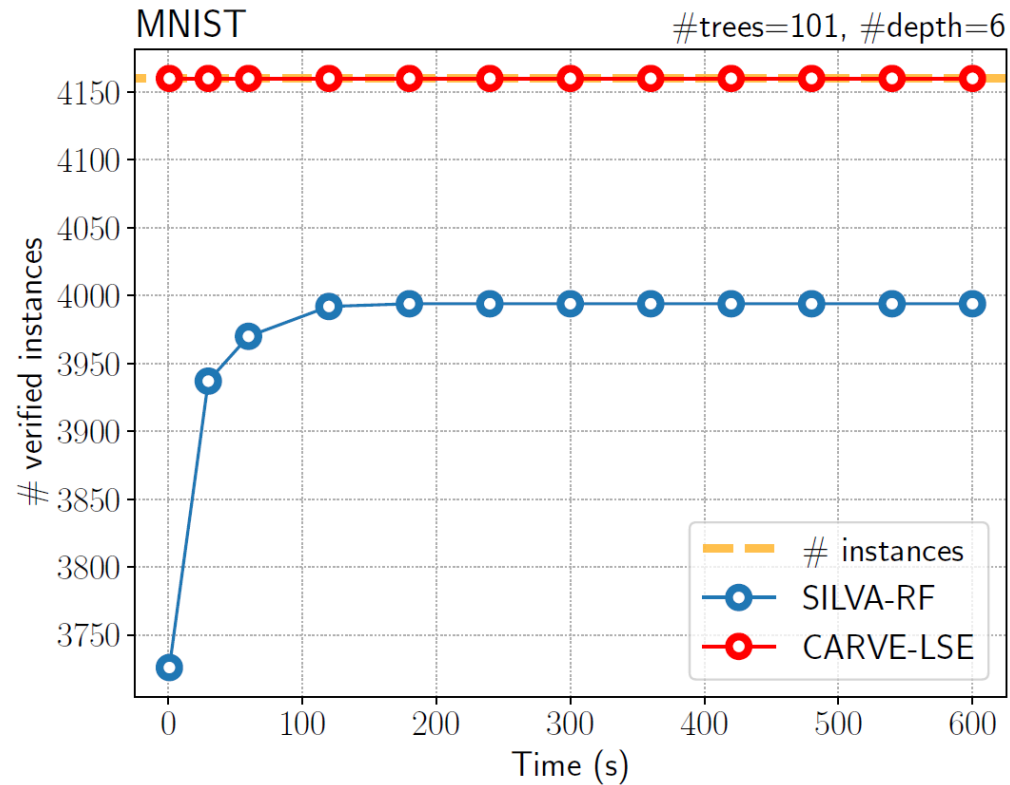
Dataset	k	Trees	Depth	Accuracy		Robustness	
				Traditional	Large-Spread	Traditional	Large-Spread
MNIST	0.0050	25	4	0.97	0.97	0.90	0.96
		101	6	0.99	0.99	0.94	0.97
	0.0100	25	4	0.97	0.97	0.72	0.90
		101	6	0.99	0.99	0.77 ± 0.02	0.97
	0.0150	25	4	0.97	0.97	0.64	0.83
		101	6	0.99	0.99	0.67 ± 0.05	0.94
Webspam	0.0002	25	4	0.90	0.90	0.83	0.87
		101	6	0.94	0.91	0.88	0.90
	0.0004	25	4	0.90	0.89	0.80	0.86
		101	6	0.94	0.89	0.85	0.86
	0.0006	25	4	0.90	0.89	0.78	0.85
		101	6	0.94	0.85	0.81	0.82

1. Large-Spread Ensembles are more robust.
2. SILVA may be forced to approximate the robustness.

Reasonable accuracy

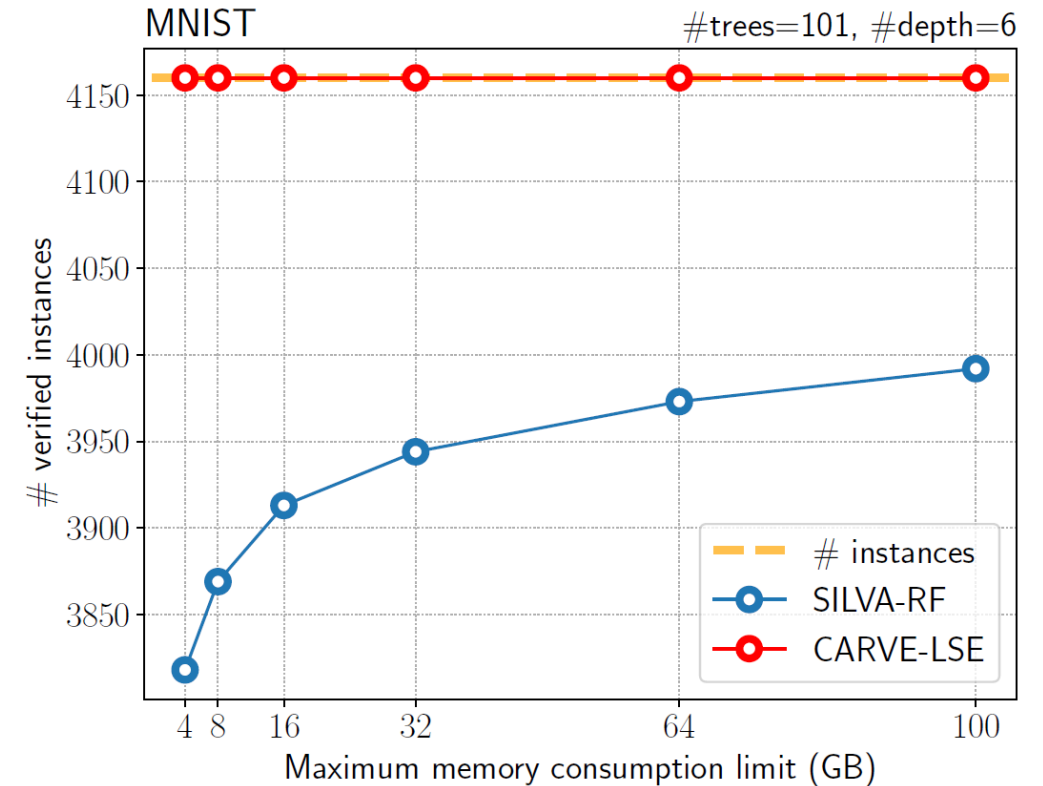
Efficiency of CARVE

Time



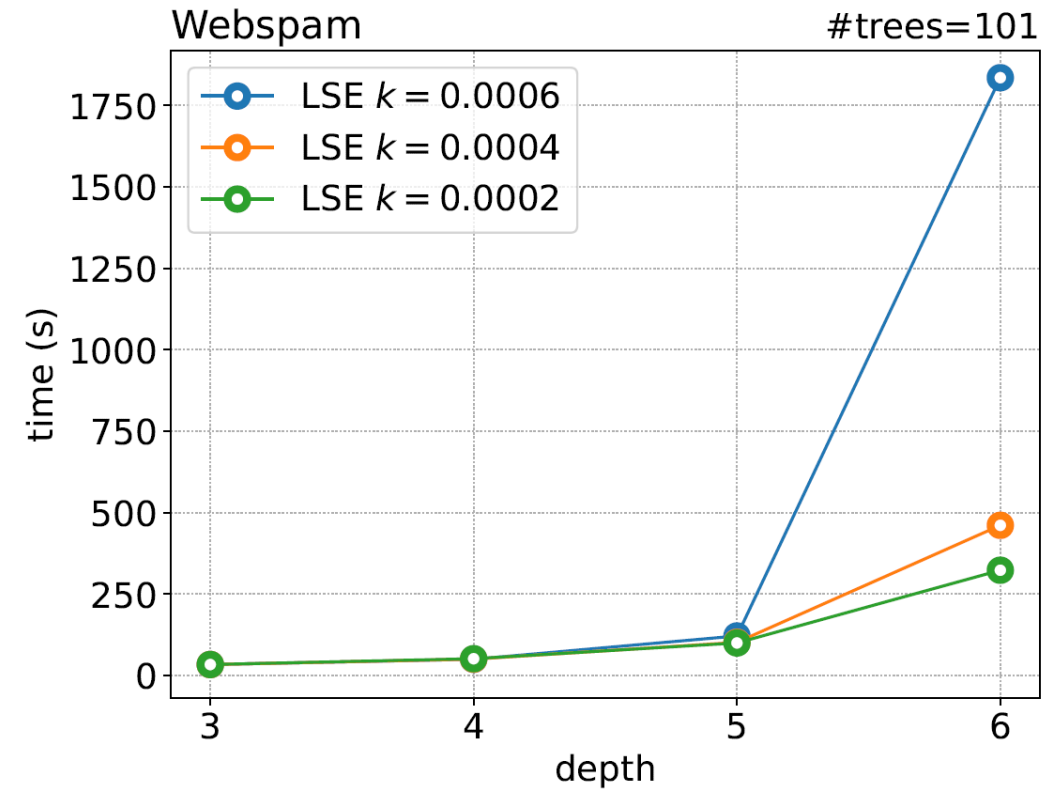
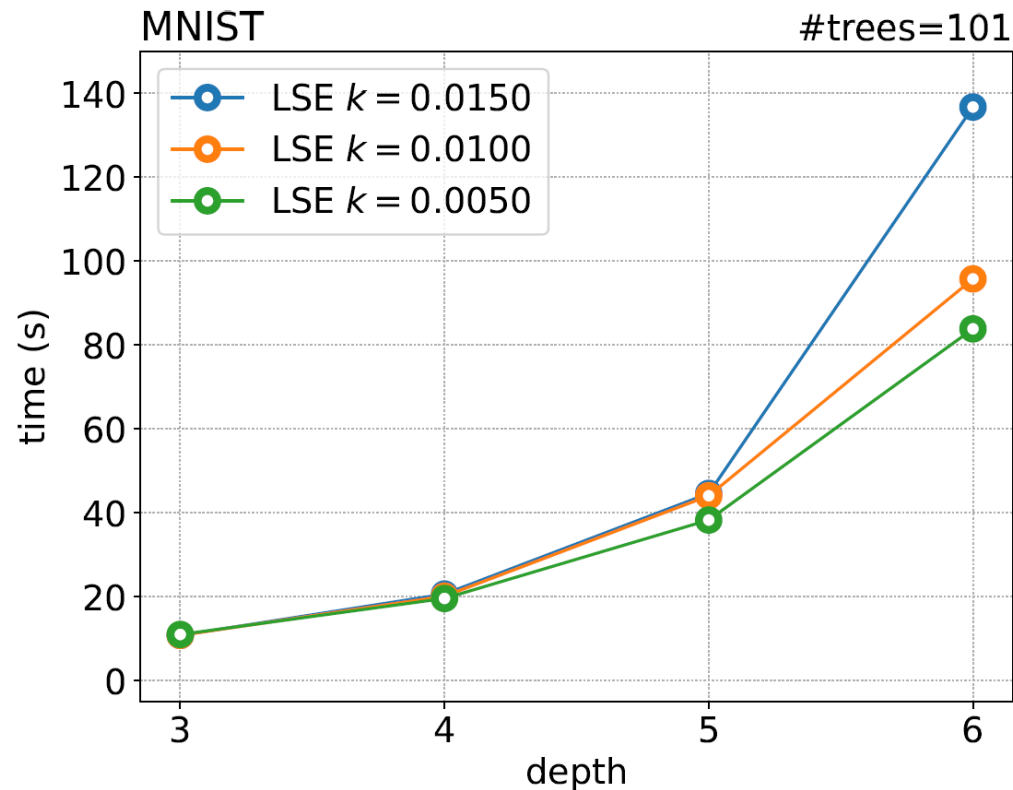
CARVE requires less than one second per instance
VS
SILVA may not verify some instances even in 10 minutes!

Memory



CARVE requires less than 4GB RAM per instance
VS
SILVA may not verify some instances even with 100GB RAM!

Efficiency of LSE



Moderate size of the ensemble or small adversarial perturbation → LSE is efficient!

Large size of the ensemble or big adversarial perturbation → the time required by LSE may increase.

Experiments With Different Norms

Dataset	k	Trees	Depth	Robustness		
				$A_{\infty,k}$	$A_{2,k}$	$A_{1,k}$
Fashion-MNIST	0.0050	25	4	0.90	0.90	0.90
		101	6	0.93	0.93	0.94
	0.0100	25	4	0.87	0.88	0.89
		101	6	0.91	0.91	0.93
	0.0150	25	4	0.88	0.89	0.89
		101	6	0.89	0.89	0.91
MNIST	0.0050	25	4	0.96	0.96	0.97
		101	6	0.97	0.98	0.98
	0.0100	25	4	0.90	0.93	0.95
		101	6	0.97	0.98	0.98
	0.0150	25	4	0.83	0.88	0.93
		101	6	0.94	0.95	0.97
REWEMA	0.0050	25	4	0.87	0.87	0.87
		101	6	0.89	0.89	0.89
	0.0100	25	4	0.87	0.87	0.87
		101	6	0.88	0.88	0.88
	0.0150	25	4	0.85	0.87	0.87
		101	6	0.88	0.88	0.88
Webspam	0.0002	25	4	0.87	0.88	0.88
		101	6	0.90	0.90	0.90
	0.0004	25	4	0.86	0.86	0.86
		101	6	0.86	0.86	0.87
	0.0006	25	4	0.85	0.86	0.86
		101	6	0.82	0.83	0.83

Take-Home Messages

1. Verifiable Learning: rethink traditional learning algorithms to make (robustness) verification of the trained model feasible.
2. The large-spread condition applied to tree-based classifiers enables complete robustness verification in poly time (NP-hard problem in general).
3. Our pruning algorithm fixes the thresholds of a traditional decision tree ensemble to enforce the large-spread condition (with a «reasonable» efficiency).
4. Large-spread ensembles sacrifice a limited amount of the predictive power but their robustness is normally higher and much more efficient to verify.

Verifiable Boosted Tree Ensembles

In our follow-up work, we consider SOTA decision tree ensembles trained through boosting schemes, e.g., GBDTs:

- The leaves of each tree now contain real-values, so it is harder to identify the optimal evasion strategy. Indeed, attacks cannot be no more totally ordered.
- Complexity results:
 - For a ∞ -norm attacker, robustness verification can be performed in **polynomial time** w.r.t. the model size.
 - For a 0-norm attacker, the robustness verification problem can be reduced to a 0-1 knapsack problem. Then, robustness verification can be performed in **polynomial time** w.r.t. the model size.
 - For a p -norm attacker, with $p \in \mathbb{N}_0$, the problem still remains **NP-hard!**

More details can be found in the paper that will be released on Arxiv in few weeks!

Data-Independent Verification of Robustness of Tree-Based Classifiers

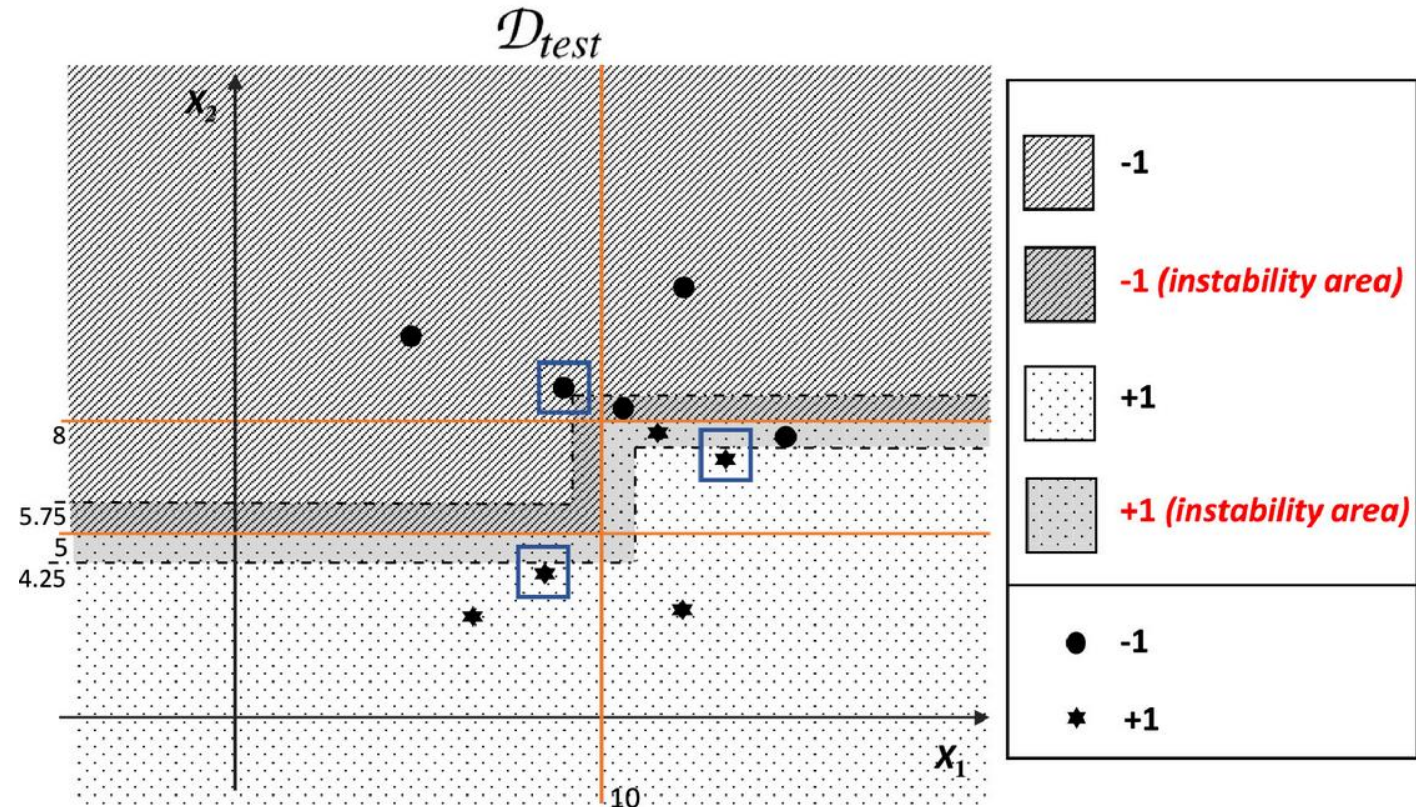
Calzavara S., Cazzaro L., Lucchese C., Marcuzzi F., Orlando S. - *Beyond Robustness: Resilience Verification of Tree-Based Classifiers*, in *Computers & Security* (2022)

Robustness

The attacker $A(\vec{x}): X \rightarrow P(X)$ maps each input to the adversarial manipulations of the instance \vec{x} .

The classifier f is **robust** on the instance \vec{x} with label y if:

1. $f(\vec{x}) = y$.
2. For all $\vec{z} \in A(\vec{x})$ we have $f(\vec{z}) = y$ (**stability** property).



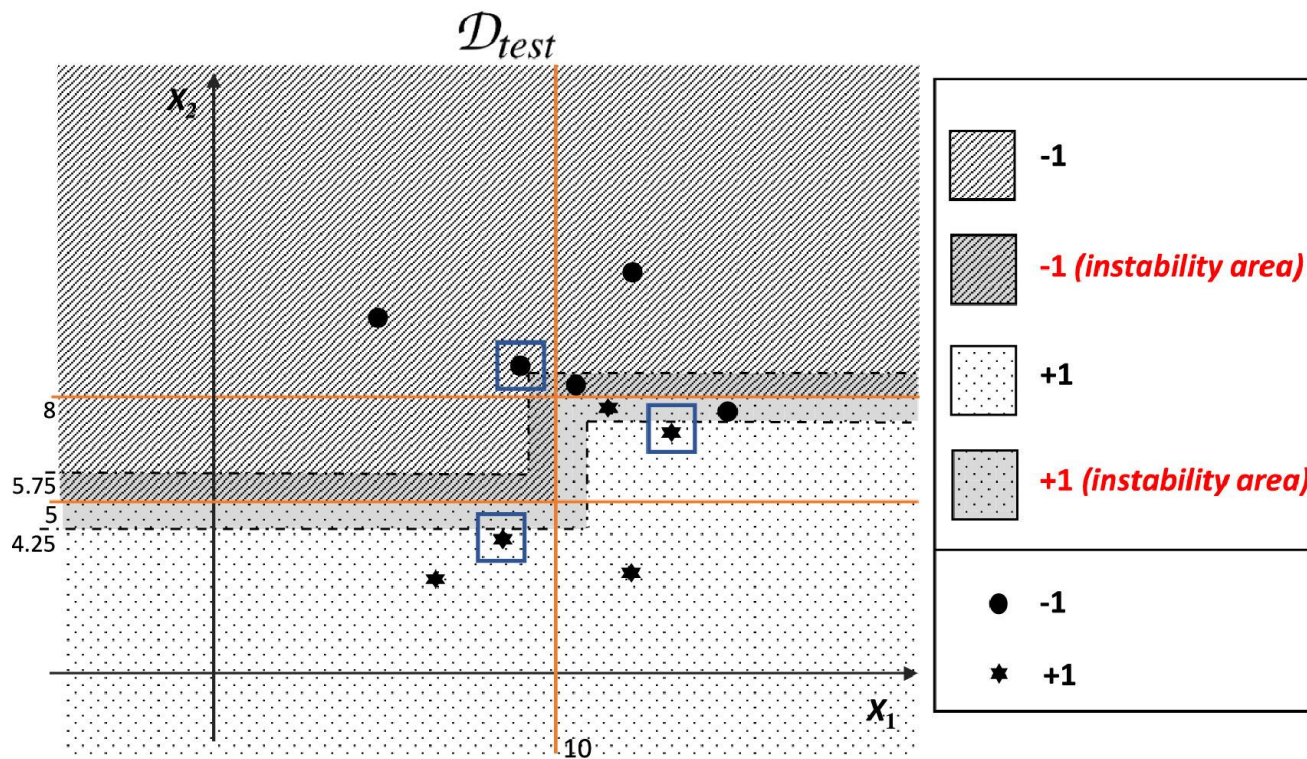
$$\text{Accuracy} = 9/10 = 0.9$$

$$\text{Robustness} = 7/10 = 0.7$$

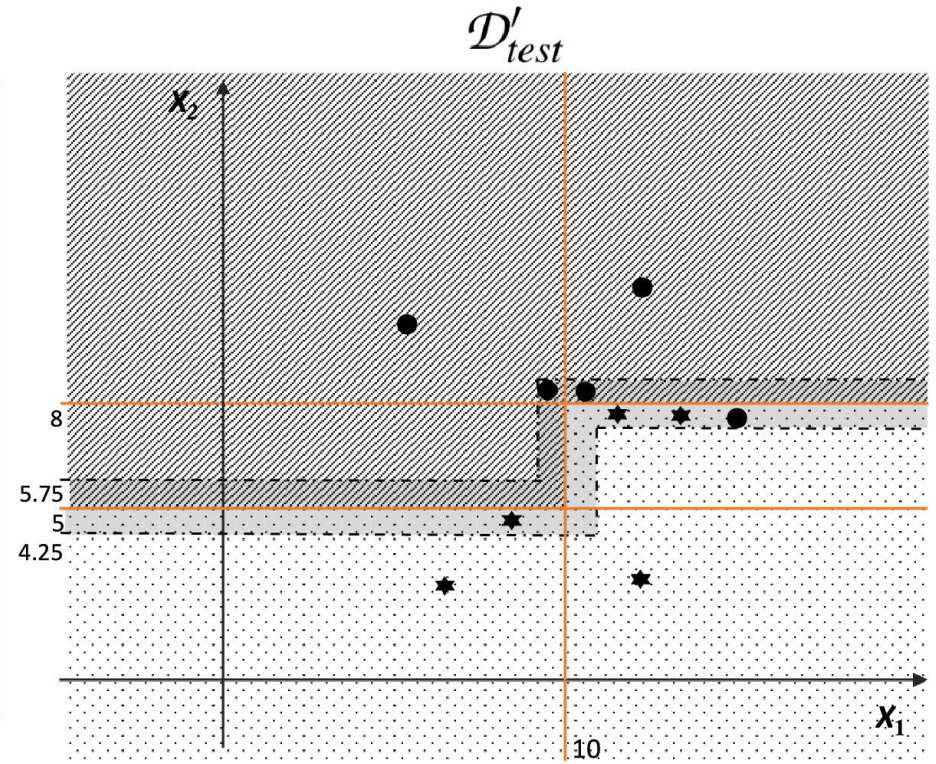
Shortcomings of Robustness

A key problem of robustness is its ***data-dependence***.

Tiny difference between two test sets \rightarrow *quite different values* of robustness!



Accuracy = $9/10 = 0.9$
Robustness = $7/10 = 0.7$

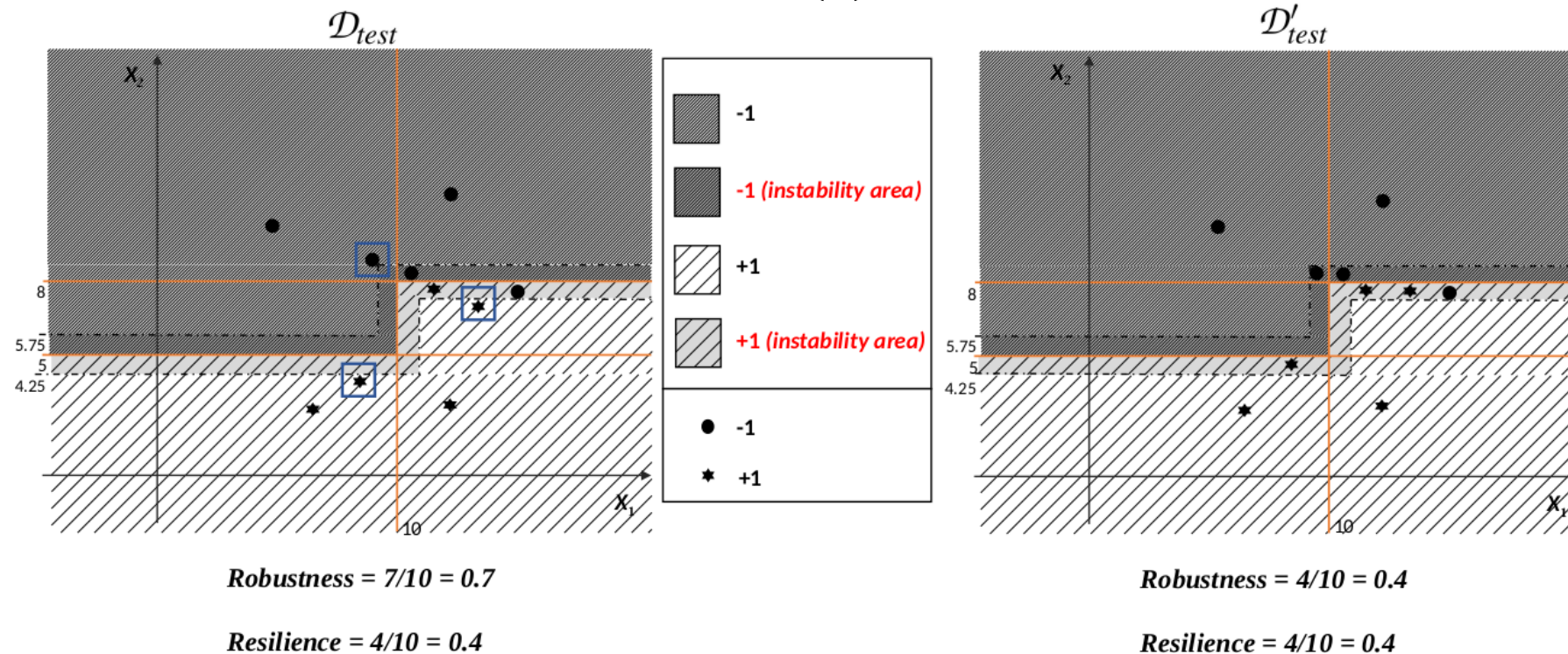


Accuracy = $9/10 = 0.9$
Robustness = $4/10 = 0.4$

Resilience

$N(\vec{x})$ is the set of neighbours of \vec{x} , instances that could have been sampled in place of \vec{x} → it helps to generalize robustness beyond the test-set.

Resilience: a classifier f is **resilient** on the instance \vec{x} if and only if f is robust on \vec{x} and f is stable on all the instances $\vec{z} \in N(\vec{x})$.



Resilience Verification

A classifier f is **resilient** on the input \vec{x} if and only if:

1. f is robust on \vec{x} \rightarrow existing methods allow to address this problem.
2. f is stable on all the instances $\vec{z} \in N(\vec{x})$ \rightarrow more challenging, because $N(\vec{x})$ is generally an infinite set of instances.

Solution to point 2: use a Data-Independent Stability Analysis (DISA), which symbolically identifies a subset $S \subseteq X$ where f is proved to be stable:

- Point 2 of resilience requires that $N(\vec{x}) \subseteq S$
- Since S is also infinite in general, we characterize it using a closed form

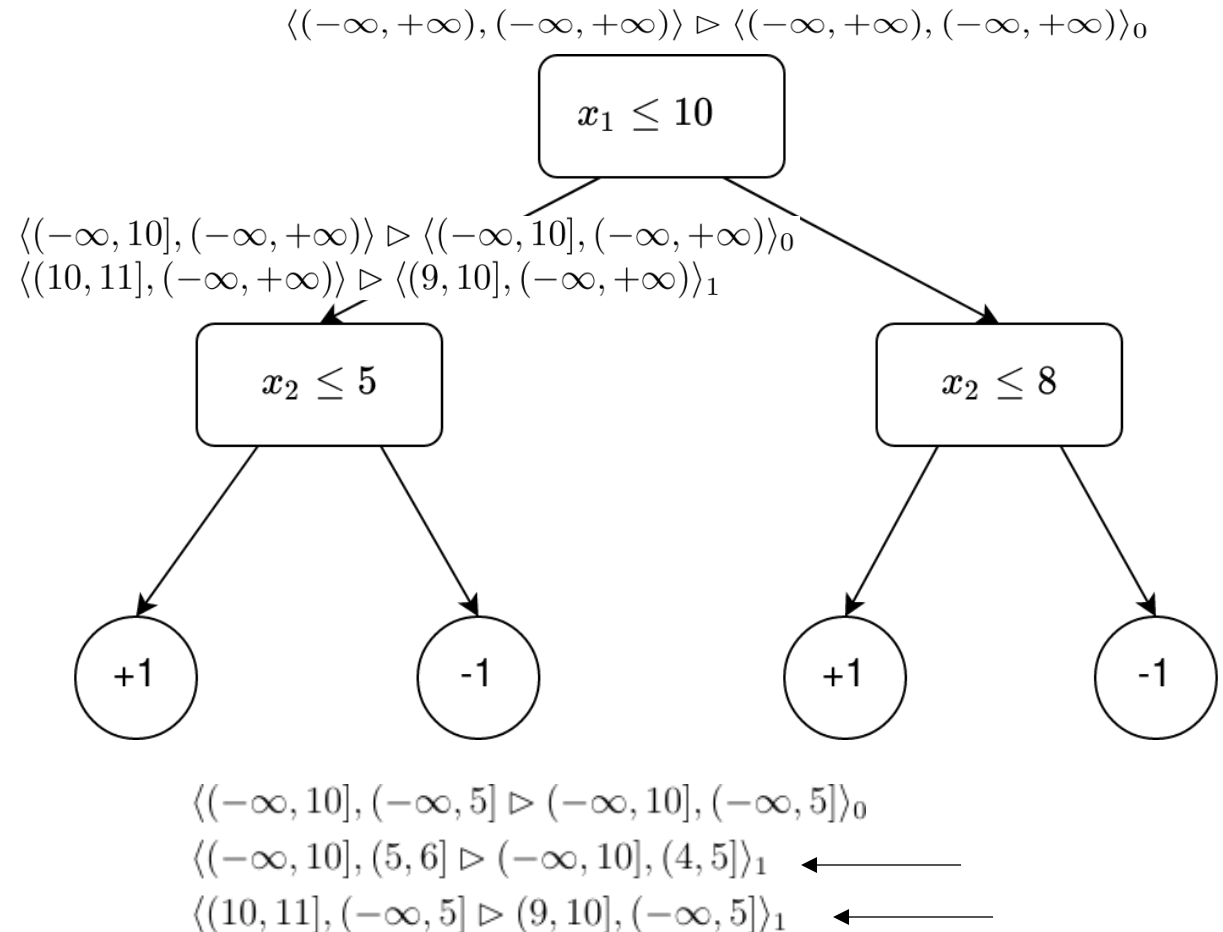
The analysis is **data-independent**: it depends on the classifier, not its inputs!

DISA for Tree Classifiers

Scenario: budget $b = 1$, perturbation in $[-1,1]$.

The subset S where f is stable is easy to identify when f is a decision tree:

- Linear-time tree traversal algorithm, which recursively annotates the nodes with a set of **symbolic attacks** (with pre-image, post-image and budget).
- Afterwards: look for leaves with budget 0 and leaves with budget > 0 with different labels and overlapping pre-images \rightarrow their \cap is part of the instability area.
- Incrementally extend the instability area until all leaves have been processed.



DISA for tree ensembles

The stability area S is harder to identify when g is a tree ensemble:

- We can prove **NP-hardness** by using the fact that robustness verification is NP-hard for decision tree ensembles [Kantchelian et al., ICML 2016]

In the paper, we present an iterative algorithm to approximate S :

- Fixed-point algorithm computing ever-increasing subsets of S .
- Early stopping does not sacrifice soundness, but will break completeness.

The analyzer in C++ is available on Github: <https://github.com/FedericoMarcuzzi/resilience-verification>

Effectiveness of Resilience Verification

Methodology:

- assess whether our resilience estimate \hat{R} accurately captures robustness for the “most unlucky” neighborhood of the test set, noted \bar{r} .

Results:

- \hat{R} is a rather precise under-approximation of \bar{r} .
- The difference between the real robustness r and the resilience \hat{R} can be significant.

Dataset	ε	# Trees	Depth	Standard Models					Robust Models				
				a	r	\hat{r}	\bar{r}	\hat{R}	a	r	\hat{r}	\bar{r}	\hat{R}
diabetes	0.01	5	3	0.708	0.662	0.643	0.656	0.636	0.727	0.714	0.701	0.675	0.662
		7	3	0.714	0.649	0.630	0.636	0.623	0.727	0.714	0.708	0.675	0.662
		9	3	0.747	0.656	0.630	0.623	0.617	0.753	0.740	0.727	0.695	0.688
cod-rna	0.01	5	3	0.775	0.686	0.672	0.639	0.621	0.752	0.715	0.707	0.698	0.691
		7	3	0.775	0.686	0.666	0.640	0.612	0.750	0.714	0.713	0.698	0.697
		9	3	0.769	0.677	0.663	0.625	0.605	0.750	0.714	0.713	0.698	0.697

Take-away messages

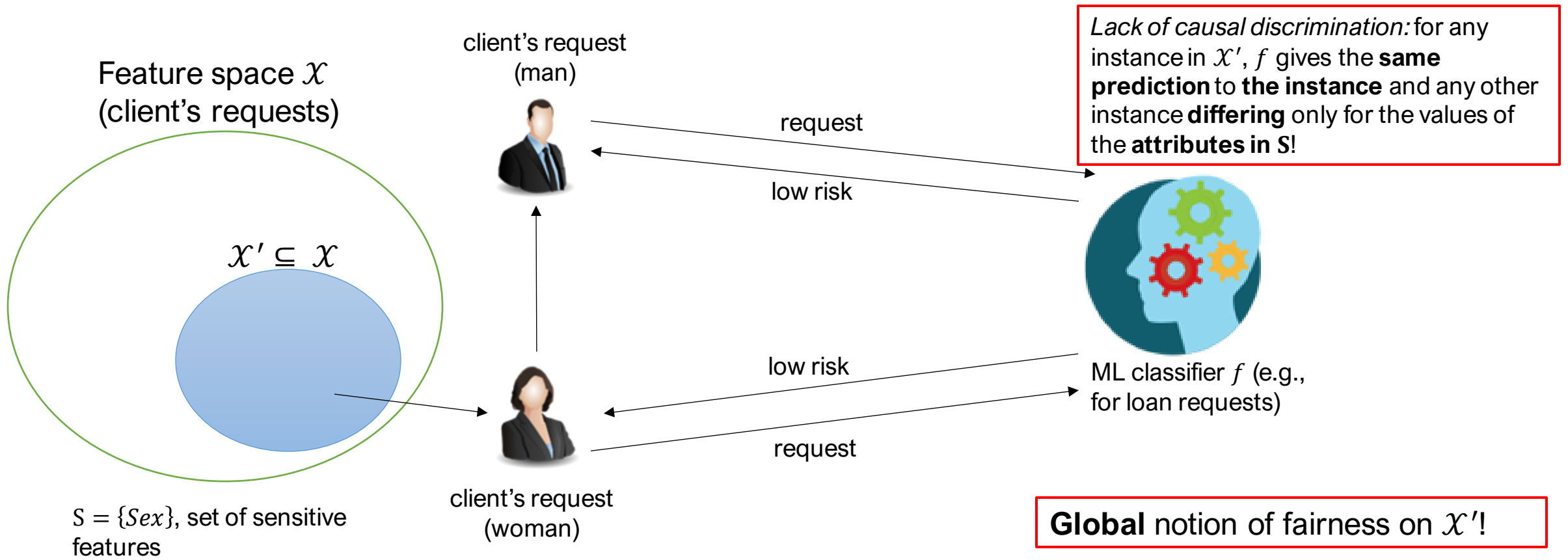
1. **Robustness may give a false sense of security.** More expressive properties (data-independent or global) are needed.
2. **Resilience is useful in practice**, since it gives a lower bound of the robustness computed on the “most unlucky” neighborhood of the test set.
3. Verification tools that analyze the inherent structure of a classifier, without relying on specific instances, are also needed.
4. Resilience can be estimated by extending existing robustness verifiers with a data-independent stability analysis.
5. The DISA may be expensive, but it is sufficient to run it only once! Moreover, it allows to verify a more expressive property!

Data-Independent Fairness Verification of Tree-Based Classifiers

Calzavara S., Cazzaro L., Lucchese C., Marcuzzi F. – *Explainable Global Fairness Verification of Tree-Based Classifiers*, in IEEE Conference on Secure and Trustworthy Machine Learning (SaTML 2023)

Causal Discrimination

We focus on **individual fairness***: give similar predictions to similar individuals.
In particular, we focus on **lack of causal discrimination****.



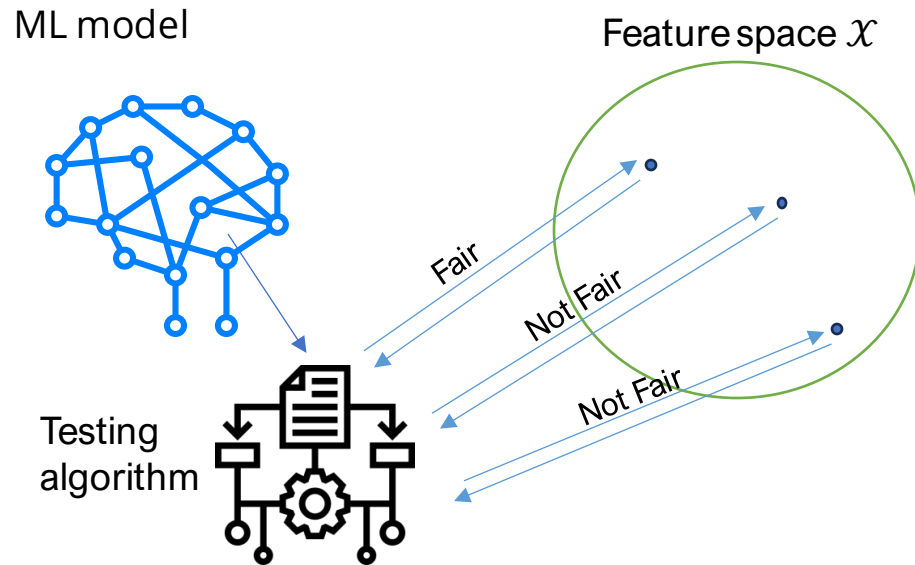
*S. Caton and C. Haas, *Fairness in machine learning: A survey*, 2020

**S. Galhotra, Y. Brun, and A. Meliou, *Fairness testing: testing software for discrimination*, ESEC/FSE 2017

SOTA of Fairness Verification

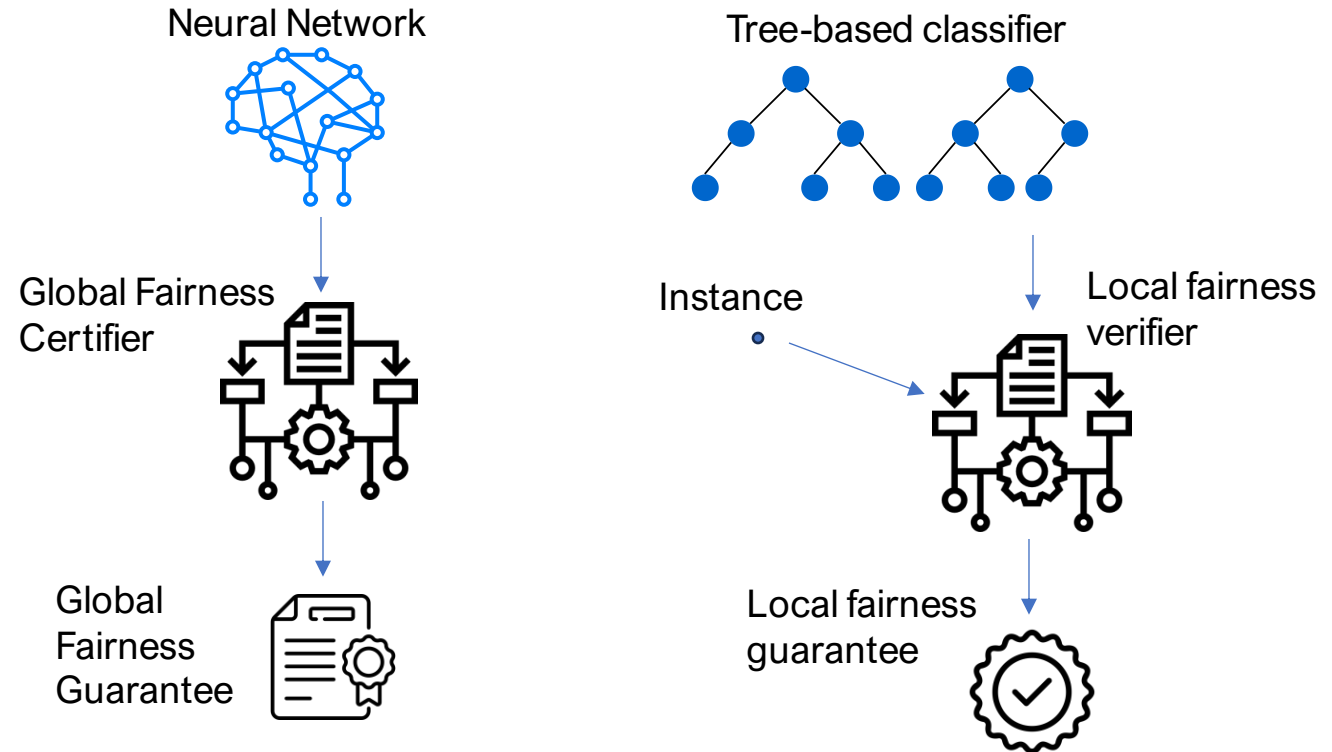
The **explainability** of the **guarantees** is **usually neglected...**

Fairness Testing*1



Under-approximated analysis!

Formal Fairness Verification*2-3



Only for Neural Networks!

Support only local properties!

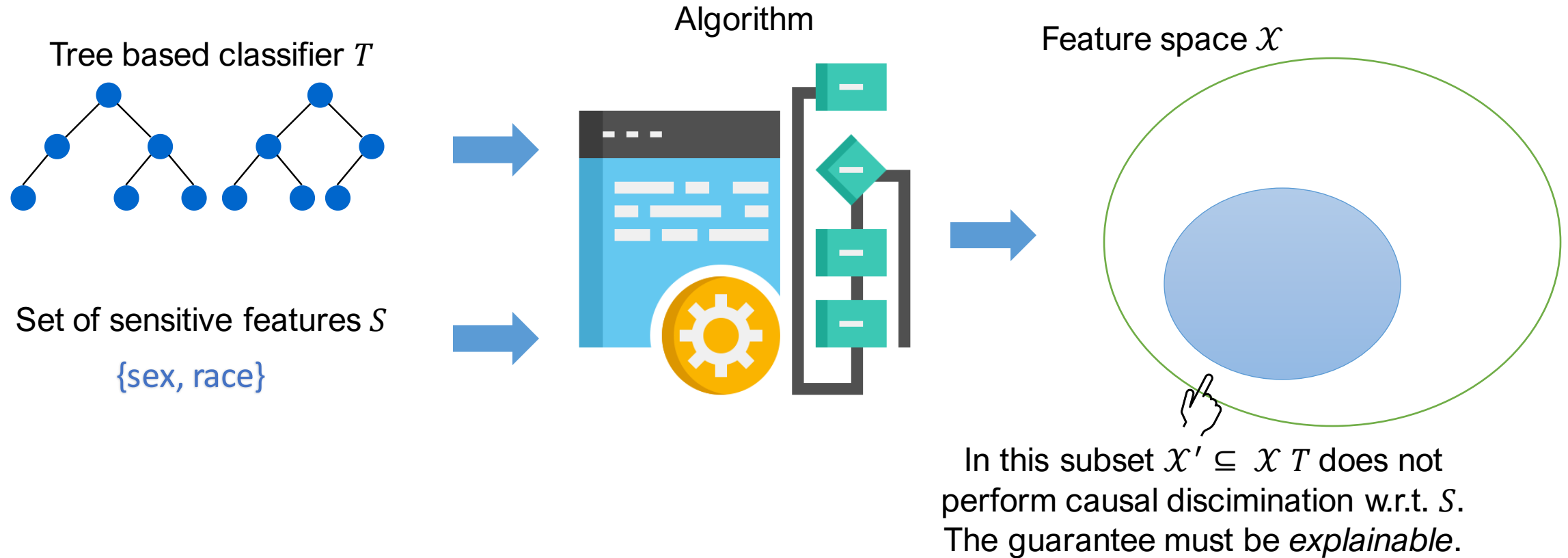
*1A. Aggarwal, P. Lohia, S. Nagar, K. Dey, and D. Saha, *Black-box fairness testing of machine learning models*, ESEC/SIGSOFT FSE 2019.

*2H. Khedr and Y. Shoukry, *Certifair: A framework for certified global fairness of neural networks*, 2022.

*3F. Ranzato, C. Urban and M. Zanella, *Fairness-aware training of decision trees by abstract interpretation*, CIKM '21 (2021).

Research problem

Problem:



Lack of Causal Discrimination and Stability

Lack of causal discrimination is connected to the **stability** property:

- Suppose to have an instance $\vec{x} \subseteq X$ and a set of possible adversarial manipulations $A(\vec{x})$;
- f is *stable* on \vec{x} if and only if $\forall \vec{z} \in A(\vec{x}): f(\vec{z}) = f(\vec{x})$. It's a **local** property.
- Lack of causal discrimination: changes to the sensitive features in S must not affect the predictions of the classifier.

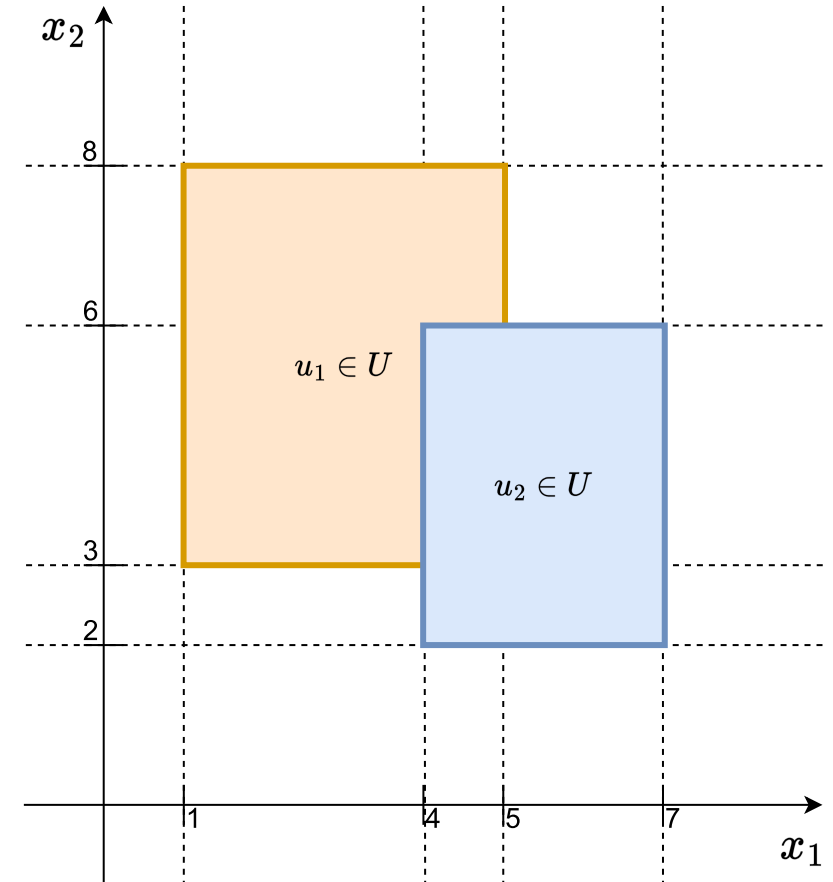
Data-Independent Stability Analysis

For tree-based models, exploit a **Data-Independent Stability Analysis algorithm (DISA)***:

- **Input:** tree-based model T and the definition of an attacker $A(\vec{x})$ (e.g., she manipulates the sensitive features of \vec{x}).
- **Output:** set of hyper-rectangles U that **over-approximates** the subsets of the feature space on which T is **unstable**.



T might perform **causal discrimination** on these subsets of the feature space!

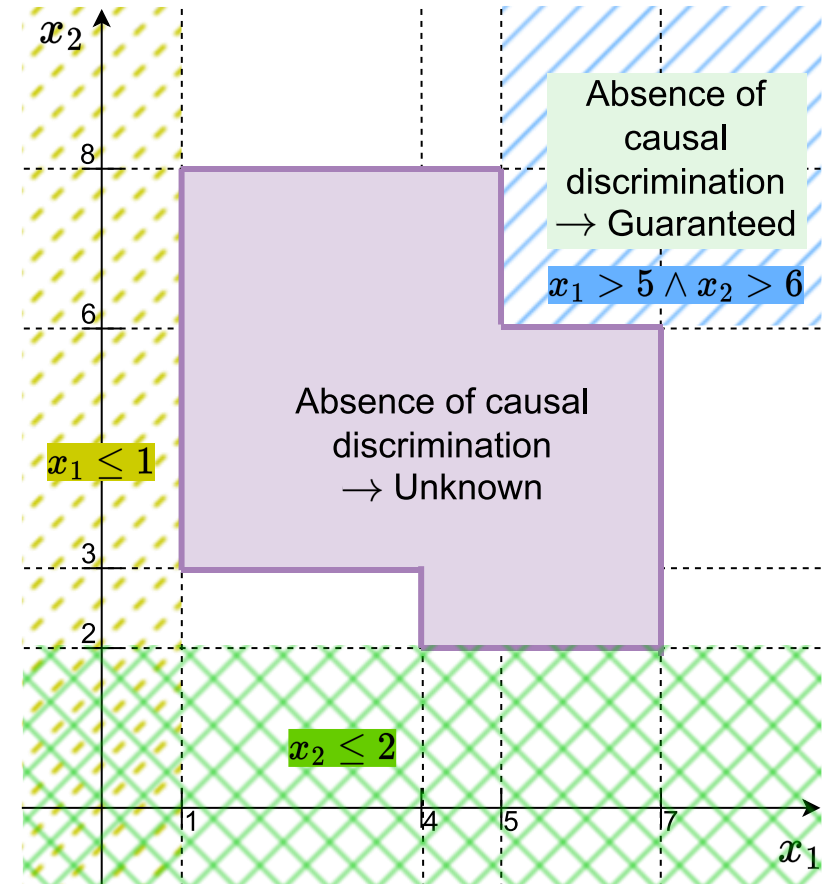


*S. Calzavara, L. Cazzaro, C. Lucchese, F. Marcuzzi, S. Orlando, *Beyond Robustness: Resilience Verification of Tree-Based Classifiers*, Computers&Security (2022)

Synthesis algorithm - Summary

Our analyzer (based on another analyzer*):

- Generates **increasingly complex sufficient conditions (logical formulas)** ensuring fairness.
- First iterations → formulas **easy to understand (explainable)**.
- The more computational resources are available, the more complex conditions may be generated.
- We measure the **precision and the performance** of the analyzer and the **explainability of the results** of the analysis.



Example

Our analysis synthesizes a set of sufficient conditions for fairness:

Conditions as **logical formulas**

Global conditions: predicate over the entire feature space

{age > 70 and job = «prof»,
credit_account < 4000 and age < 35 and housing = «rent»}

Explainable formulas: readily understandable

Our analysis is precise, explainable, reasonably efficient and proved sound and complete (details in the full paper)!

The analyzer is available on Github: <https://github.com/LorenzoCazzaro/explainable-global-fairness-verification>

Take-away messages

1. **Testing does not allow to verify global properties.**
2. The guarantees provided by a verifier should be also easily interpretable by an human and informative about the ML classifier.
3. Tools for verifying robustness (resilience) may be used to verify fairness and viceversa.
4. Our analyzer returns global fairness guarantees that are explainable, since they are logical formulas.
5. Our synthesizer requires a lot of time to return all the possible fairness guarantees, but you need to run the analysis only once!

Final Remarks

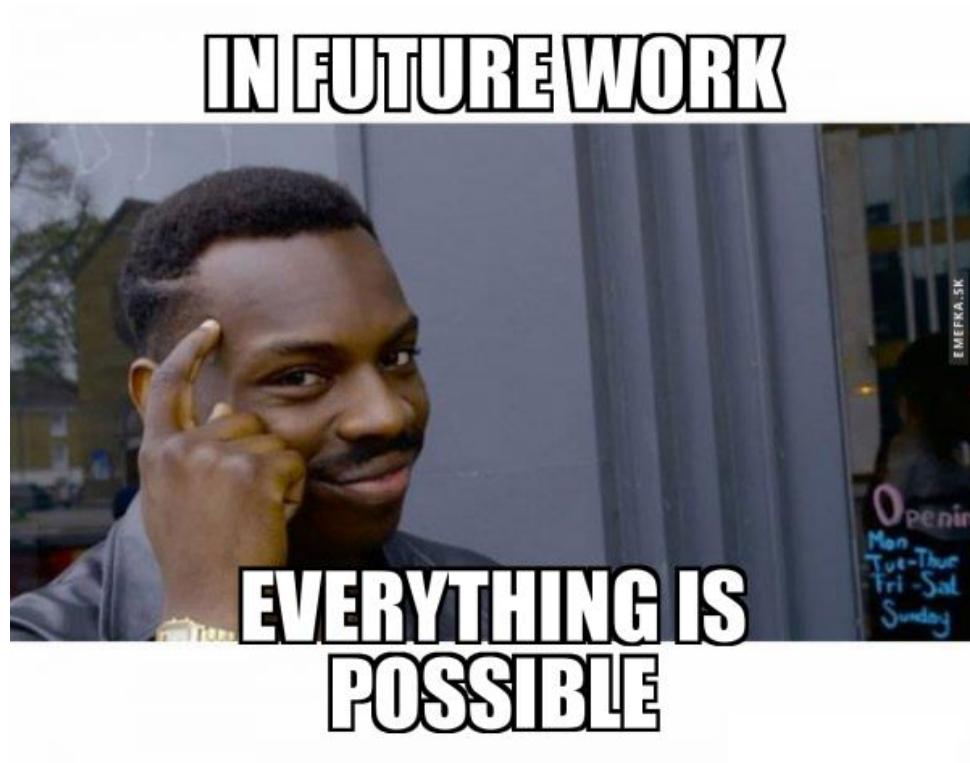
Conclusion – Efficient Verification

- The problem of verifying the security/fairness property of ML models may require **solving NP-hard problems**, so the efficiency of the verification algorithm may be sacrificed.
- Verifiable Learning proposes to train ML models that are **verifiable in an efficient way by design**.
- Enforcing the Large-Spread condition on a decision tree ensemble (with majority voting as aggregation scheme) enables robustness verification in **poly-time for any norm-based attacker (an NP-hard problem in general)**.
- Enforcing the Large-Spread condition on a decision tree ensemble trained through gradient boosting enables robustness verification in **poly-time for infinity and 0-norm attackers**.

Conclusion – Expressive Properties

- We need **expressive properties** for defining the trustworthy behaviour of ML models and **methods for verifying** them.
- Defining **data-independent properties** that depend only on the structure of the classifiers allows us to define properties that hold globally instead of holding locally.
- A **new security property, resilience**, enables a more conservative security assessment of the ML models.
- A **new tool for verifying fairness** of tree-based classifiers enables the verification of the global fairness instead of local fairness.
- The proposed analyses are computationally expensive, but the user needs to run the analysis only once.

Future Work





- Characterize better the set of possible neighbors of an instance.
- Train tree-based classifier that can exhibit a high resilience / global fairness.
- Generalize verifiable learning to other ML models, e.g., neural networks.

Lorenzo Cazzaro
Ph.D. student in Computer Science

 @LorenzoCazz

 lorenzo.cazzaro@unive.it

 Lorenzo Cazzaro

 LorenzoCazzaro

 <https://lorenzocazzaro.github.io/>



Ca' Foscari
University
of Venice



Thank you! Questions?