

Tree-Ensemble Repair Through Constrained Optimization

Lorenzo Cascioli, Chaahat Jain, Marcel Steinmetz, Jörg Hoffmann and Jesse Davis

Many ML models must satisfy certain **properties beyond accuracy**



Fairness

A loan approval model should never discriminate based on sex or race.



Safety

A self-driving car should never accelerate at a red light.

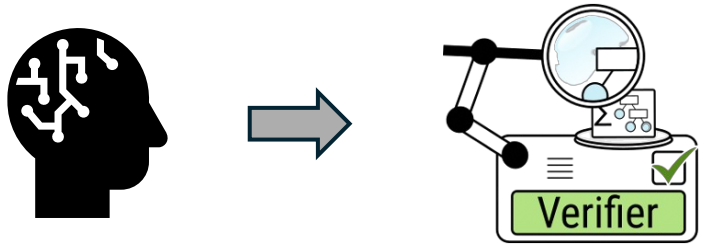
…



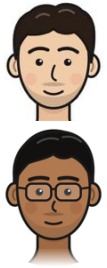
But usually they don’t!

This limits the applicability of ML in sensitive use cases.

Verification identifies violations as concrete counterexamples:



Verification:
post-hoc analysis to discover
a model's undesirable behaviors.



	29	PhD student	~35k €/year
	29	PhD student	~35k €/year

loan accepted

loan denied

unfair model!

This tells us that our model violates the property,
but **it doesn't tell us how to fix the issue!**

Our goal

Slightly **modify a learned model** to **fix** its **behavior on the counterexamples**.

Model Repair

Given: model T, training data D, set of counterexamples E

Do: produce modified model T' that

- does not violate property on each $e \in E$
- has similar performance to T on D



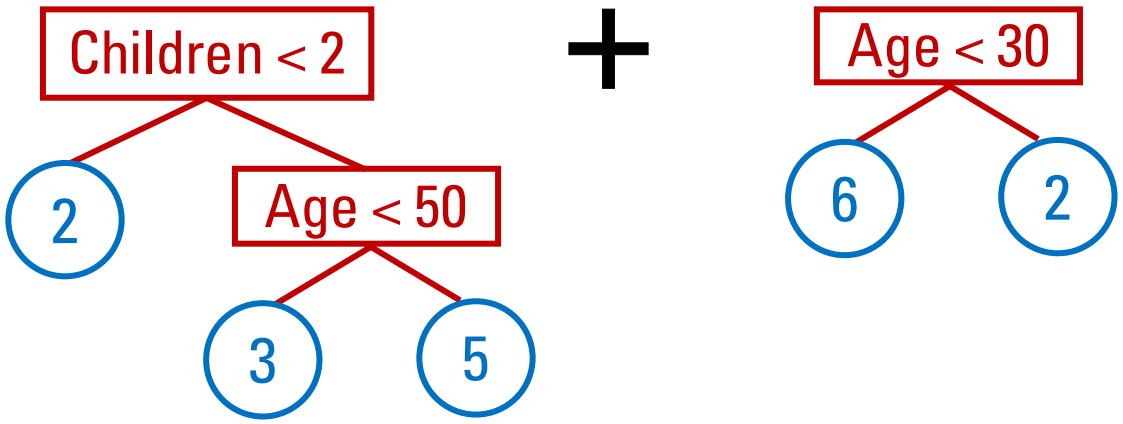
We specifically study how to repair tree ensembles:

Random Forest

dmlc XGBoost

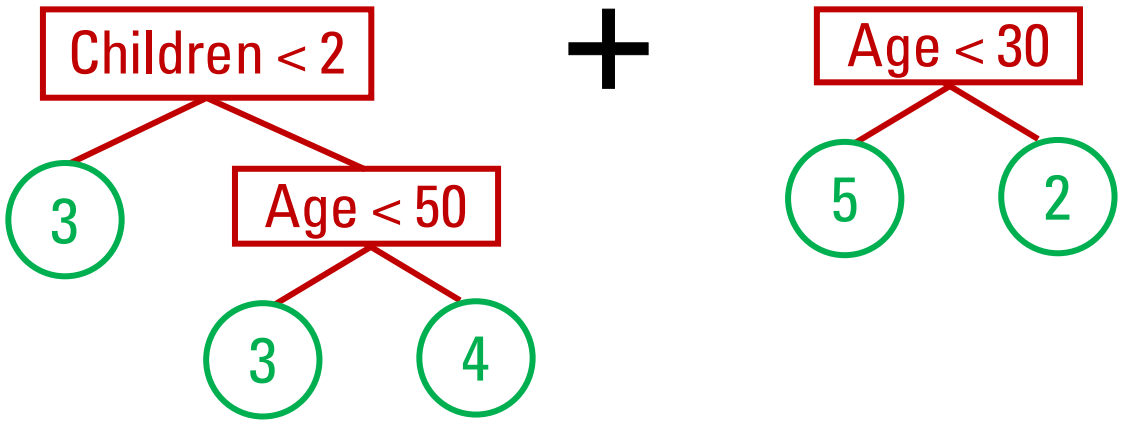
LightGBM

We encode tree ensemble repair as a **leaf refinement task**



- keep trees structure fixed
- slightly **change leaf values**

We encode tree ensemble repair as a **leaf refinement task**



- keep trees structure fixed
- slightly **change leaf values** $\nu \rightarrow \nu'$

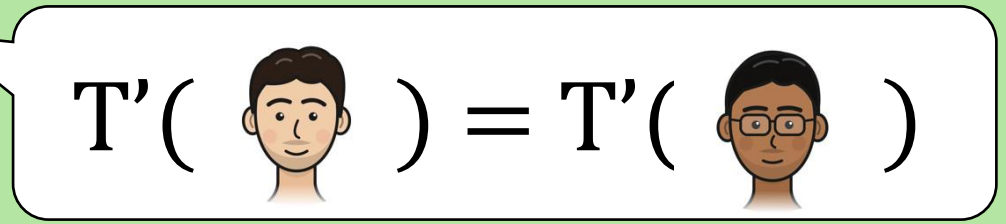
Contribution 1

A **MILP encoding to repair a tree ensemble**

We obtain the new leaf values ν' for the ensemble through:

$$\text{minimize}_{\nu'} \quad \text{PerformanceDrop}\,(\nu, \nu', D) \;+\; \lambda \cdot ||\nu' - \nu||$$

$$\text{s.t.} \quad \text{property holds } \forall e \in E$$



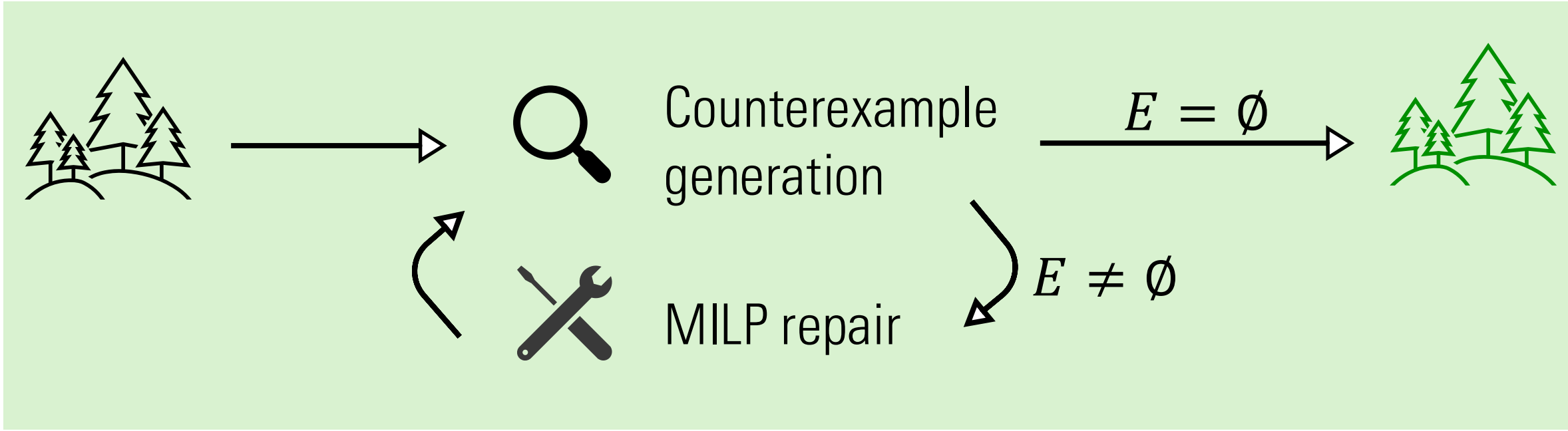


One hard constraint per each counterexample,
so MILP solution is guaranteed to fix all identified violations.

Contribution 2

A **repair loop** to provably enforce the property

Repair could introduce new violations.
So we iterate until no violations exist:

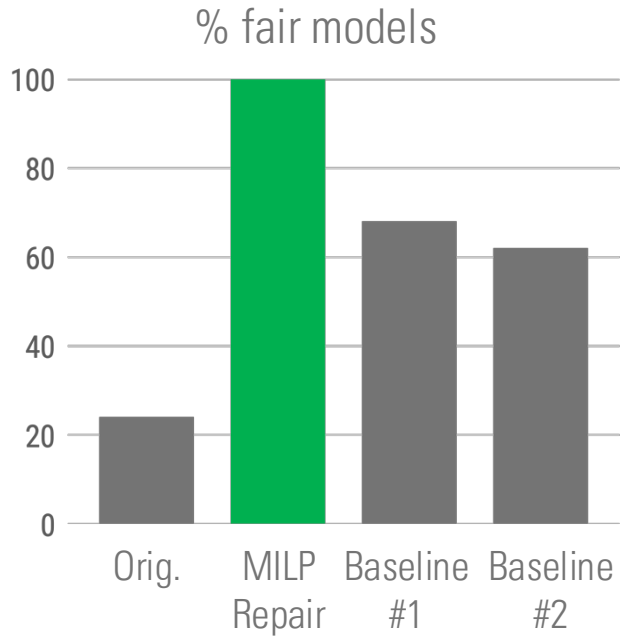


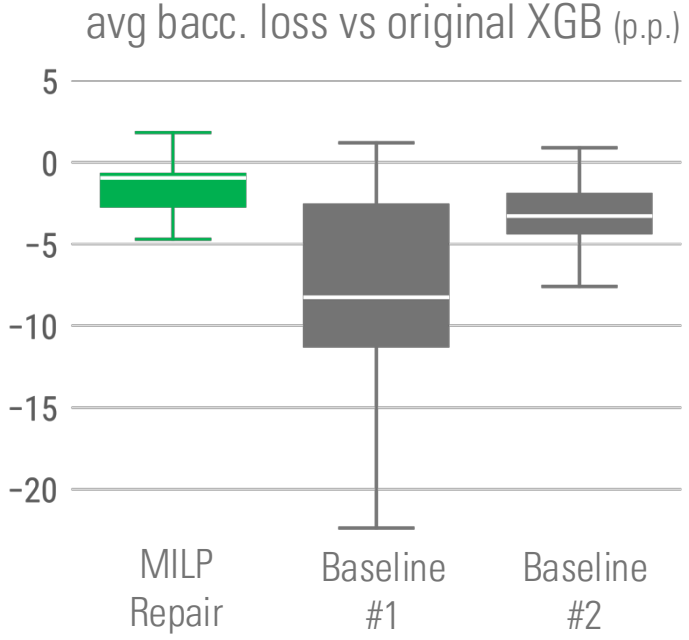
Our MILP repair loop **enforces properties** with **no performance drop**

e.g., repairing unfair XGBoost models

MILP repair: 100% fair models
Baselines: <70%

MILP repair: ~1.5 p.p. avg cost in bal. accuracy
Baselines: 3-9 p.p. avg cost





Baselines:
property-aware training.

More details in the paper:
- more properties
- more ensemble types

Tree-Ensemble Repair Through Constrained Optimization

ML models often need to satisfy properties beyond accuracy.

They usually don't, and verification finds concrete violations.

We propose a **MILP encoding to repair a tree ensemble** for given counterexamples.

We iterate verification and repair in a **repair loop** that guarantees property satisfaction.



Lorenzo Cascioli
@l_cascioli

Chaahat Jain

Marcel Steinmetz

Jörg Hoffmann

Jesse Davis

check out the paper!

KU LEUVEN LEUVEN.AI DTAI Declarative Languages & Artificial Intelligence fwo