Detecting Evasion Attacks in Deployed Tree Ensembles

Laurens Devos, Lorenzo Perini, Wannes Meert, Jesse Davis
laurens.devos@kuleuven.be
@laudevs

1 Tree Ensembles Can Be **Mislead**

- Susceptible to **Evasion Attacks**
  - Adversarial examples at test time
  - Small carefully crafted changes to inputs fooling the model

Evasion attack

$T($  $) = 7$

$T($  $) = 5?$

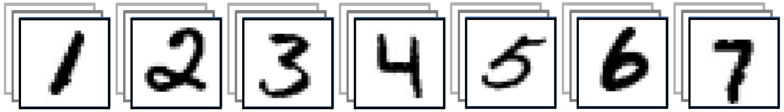
- Many performant attacks exist

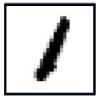
MILP	LT-Attack	Veritas
Kantchelian et al. ICML'16	Chen et al. NeurIPS'19	Devos et al. ICML'21

What we know:

2 Models **Always Make A Prediction**

Model T trained on data:



$T($  $) = 1$

$T($  $) = 7$

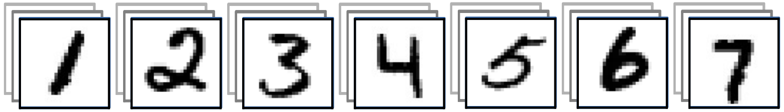
$T($  $) = 3$

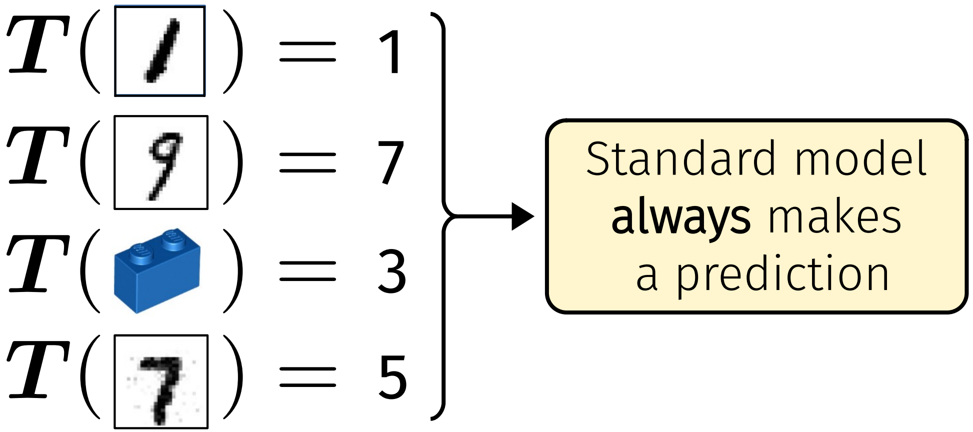
$T($  $) = 5$

What we know:

2 Models **Always Make A Prediction**

Model T trained on data:



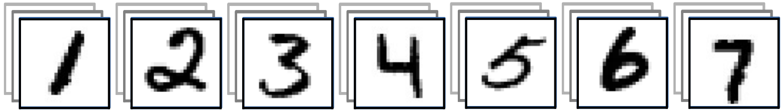


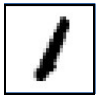
Maybe we should **abstain from making a prediction?**

What we know:

2 Models **Always Make A Prediction**

Model T trained on data:



$T($  $) = 1$

$T($  $) = 7$

$T($  $) = 3$

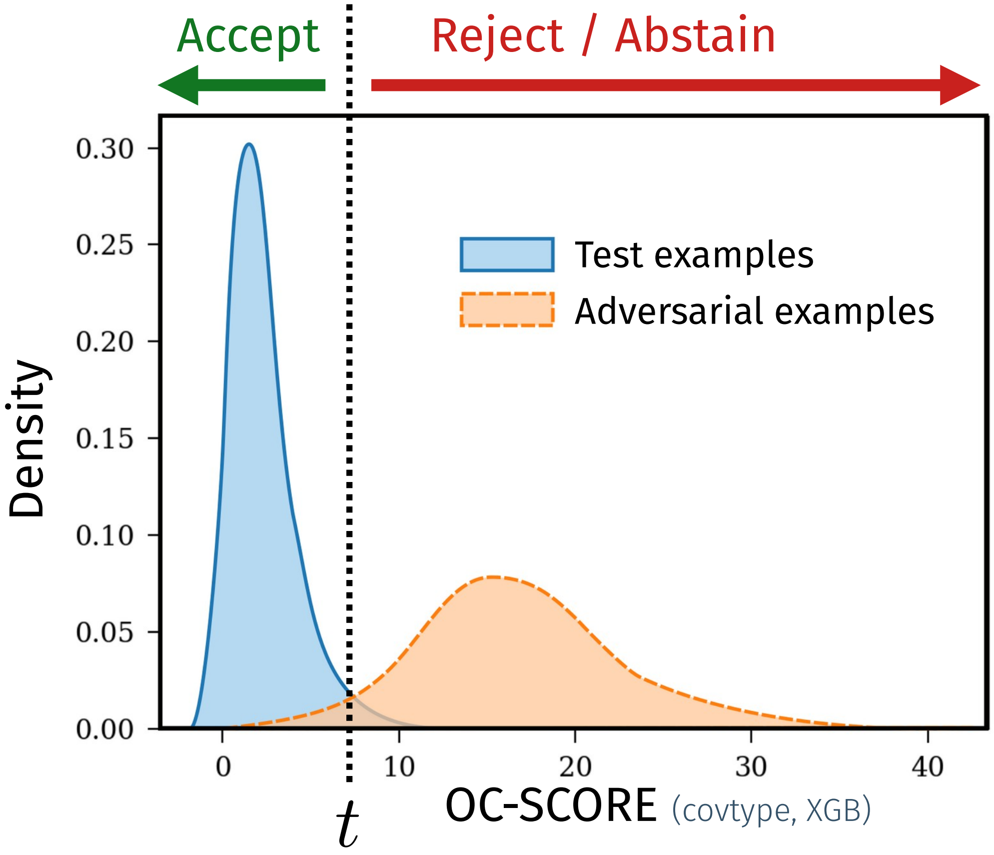
$T($  $) = 5$

Standard model **always** makes a prediction

Maybe we should **abstain from making a prediction?**

This paper **OC-SCORE** identifies suspicious examples

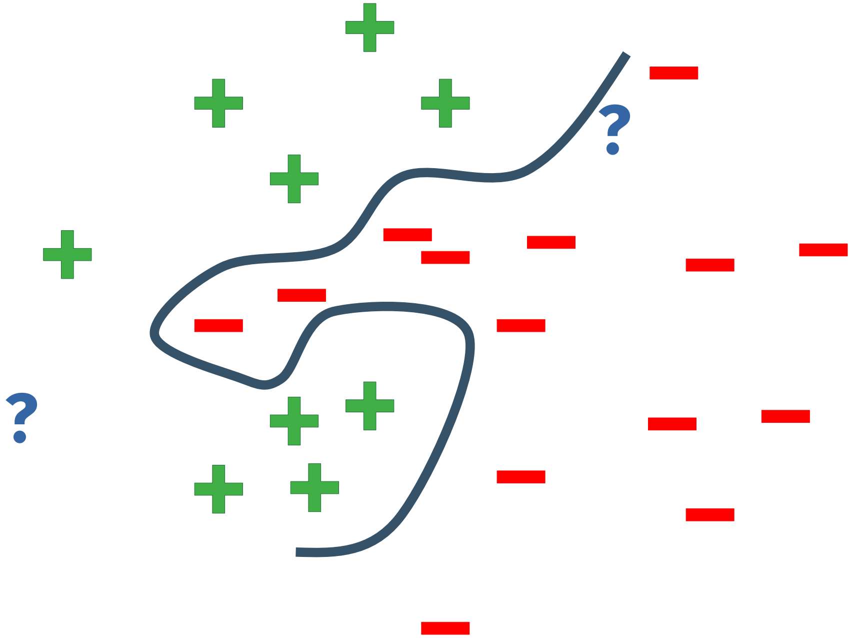
This paper OC-SCORE identifies suspicious examples



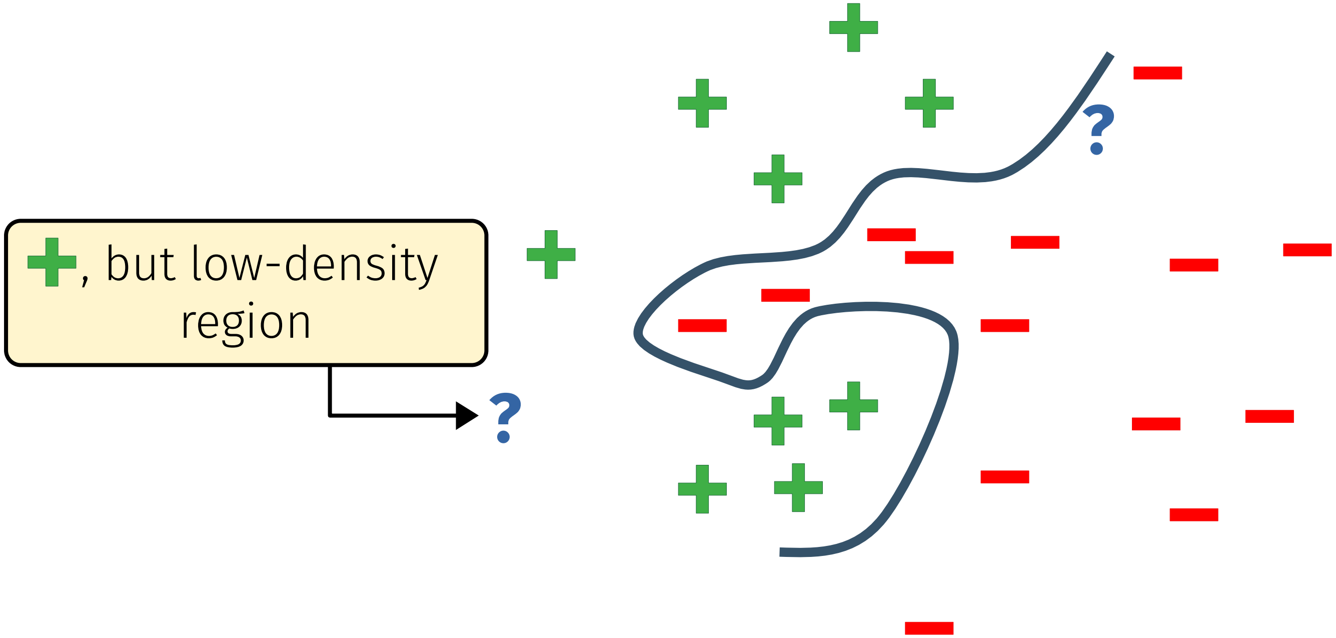
Given a model and a set of ‘*normal*’ examples

- **Assign** a score s to a new example
- **Reject** if $s > t$

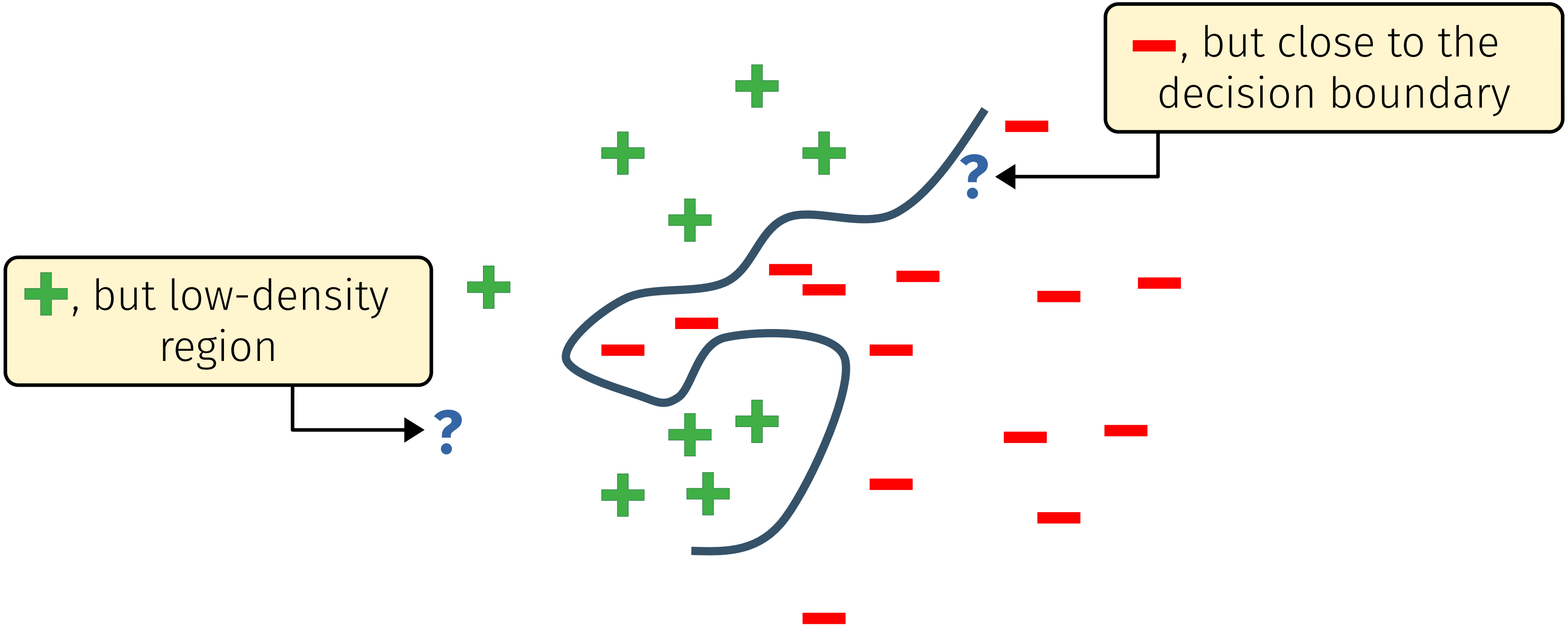
Need Insights into **Why Non-Robust**



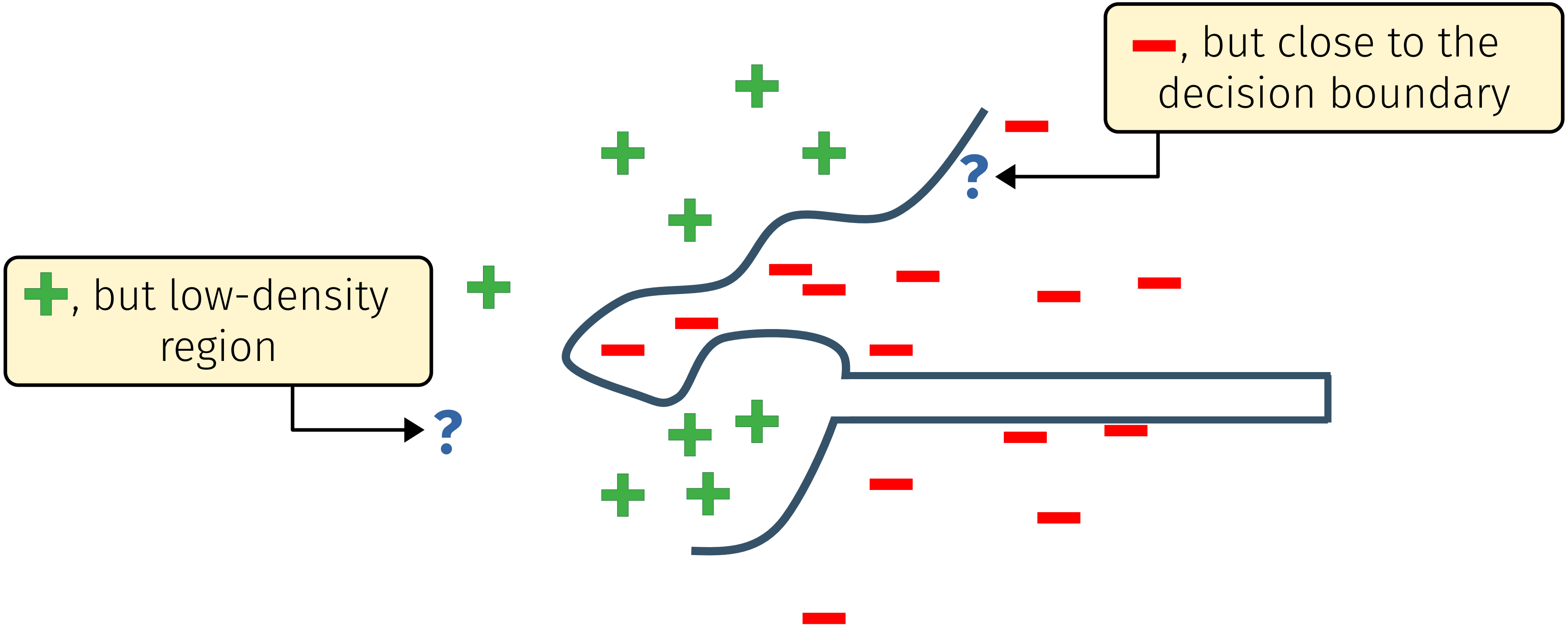
Need Insights into **Why Non-Robust**



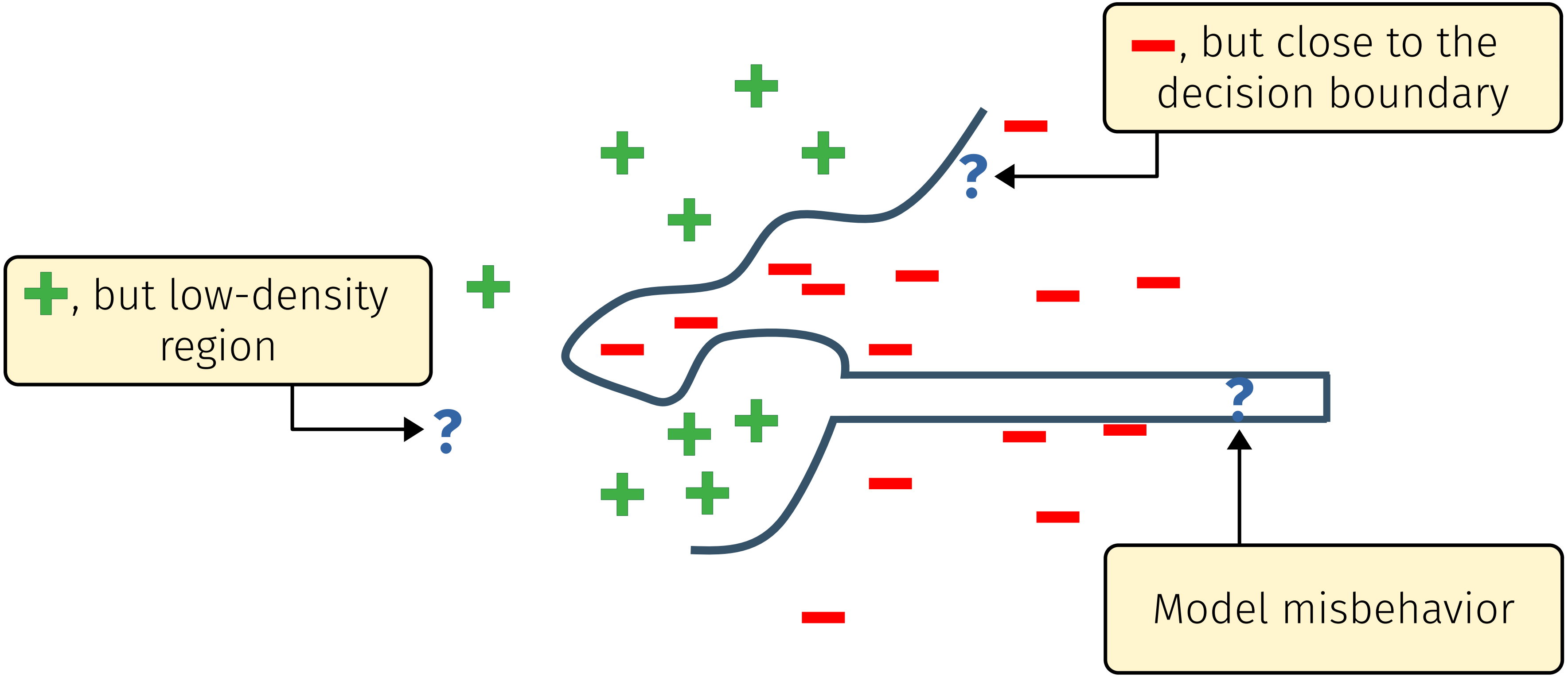
Need Insights into **Why Non-Robust**



Need Insights into **Why Non-Robust**



Need Insights into **Why Non-Robust**



Possible ways to **Deal With Evasion Attacks**

Possible ways to **Deal With Evasion Attacks**

- Collect more data
  - By hand → have fun collecting it
  - Hardening: generate data automatically
    [Goodfellow et al., ICLR'15, Kantchelian et al., ICML'16]

Possible ways to **Deal With Evasion Attacks**

- Collect more data
  - By hand → have fun collecting it
  - Hardening: generate data automatically
    [Goodfellow et al., ICLR'15, Kantchelian et al., ICML'16]
- Robust tree learners
  [Chen et al. ICML'19, Calzavara et al. DMKD'20, Vos & Verwer, ICML'21]

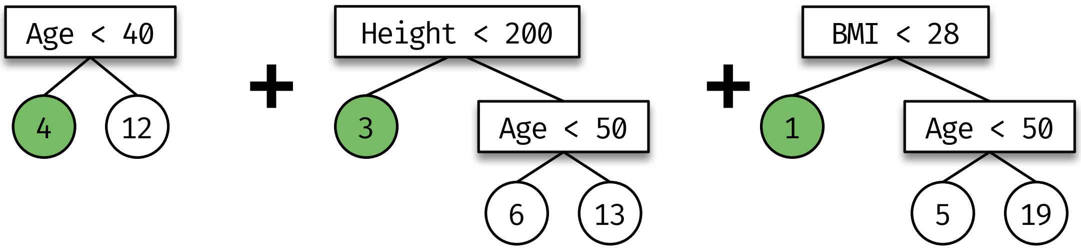
Possible ways to **Deal With Evasion Attacks**

- Collect more data
  - By hand → have fun collecting it
  - Hardening: generate data automatically
    [Goodfellow et al., ICLR'15, Kantchelian et al., ICML'16]
- Robust tree learners
  [Chen et al. ICML'19, Calzavara et al. DMKD'20, Vos & Verwer, ICML'21]
- **Post-deployment detection → OC-SCORE**

This paper

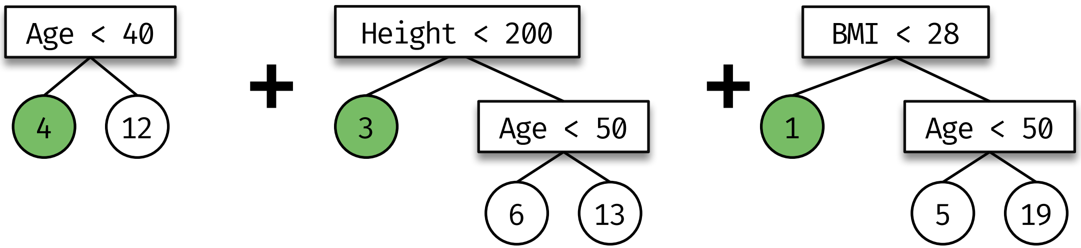
Post-Deployment Detection

The **OC-SPACE**



Age	**Height**	**BMI**
32	176	22

The **OC-SPACE**



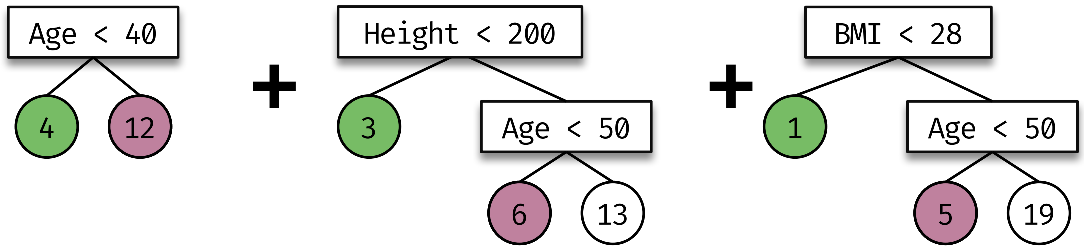
Age	Height	BMI
32	176	22

`OC({A=32, H=176, B=22}) = (4,3,1)`

- **OC** = tuple of compatible leaves

L. Devos, L. Perini, W. Meert, J. Davis – DTAI, KU Leuven

The **OC-SPACE**

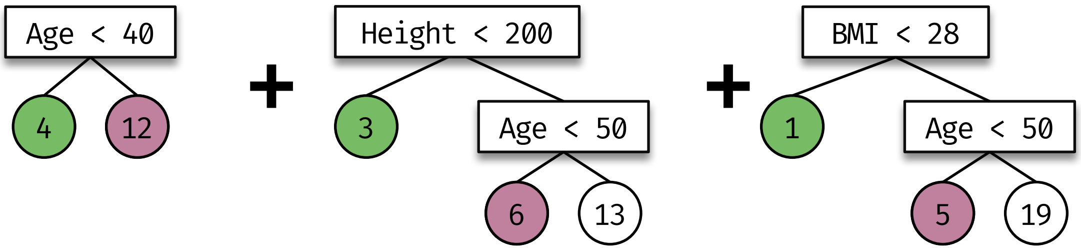


Age	Height	BMI
32	176	22
55	201	29

```
OC({A=32, H=176, B=22}) = (4,3,1)
OC({A=55, H=201, B=29}) = (12,6,5)
```

- **OC** = tuple of compatible leaves

The **OC-SPACE**



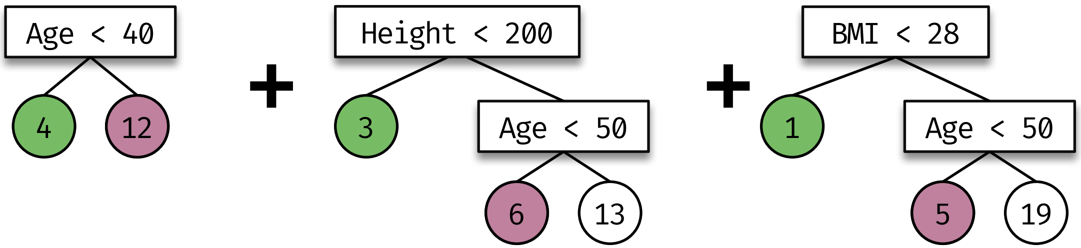
Age	**Height**	**BMI**
32	176	22
55	201	29

OC({A=32, H=176, B=22}) = (4,3,1)

OC({A=55, H=201, B=29}) = (12,6,5)

- **OC** = tuple of compatible leaves
- **OC-SPACE** = set of all possible OCs

The **OC-SPACE**



Age	Height	BMI
32	176	22
55	201	29

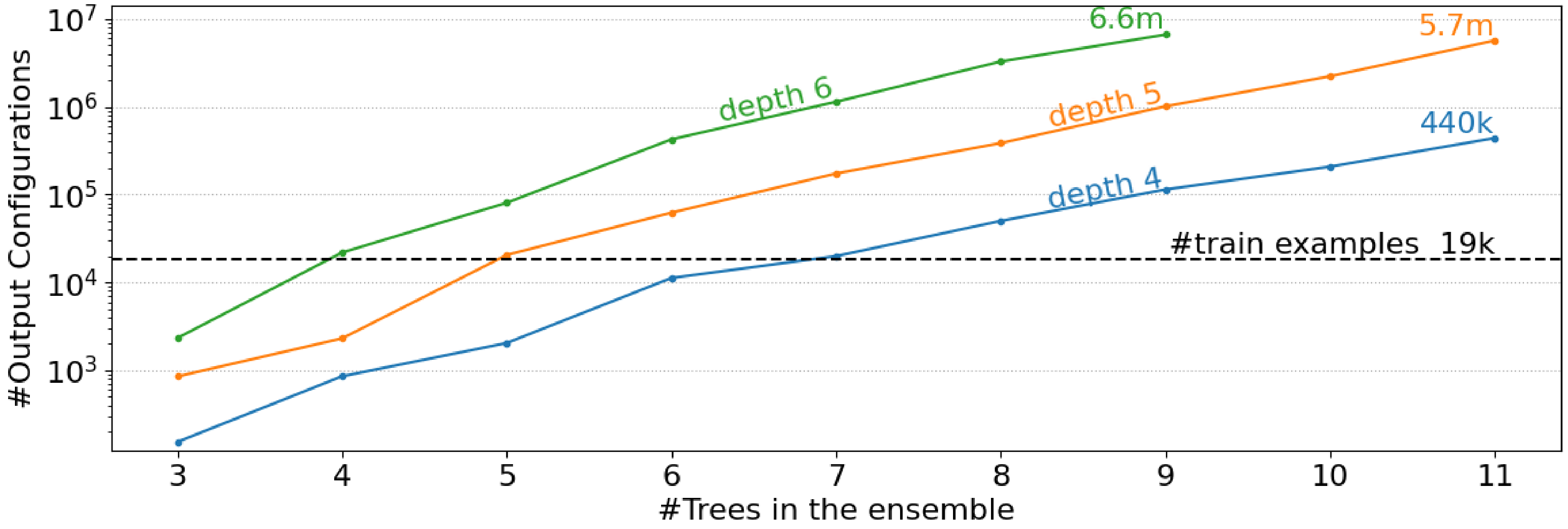
OC({A=32, H=176, B=22}) = (4,3,1)

OC({A=55, H=201, B=29}) = (12,6,5)

- **OC** = tuple of compatible leaves
- **OC-SPACE** = set of all possible OCs

How big is **OC-SPACE**?

OC-SPACE Explodes



Vast majority of OCs **never visited** by a training example

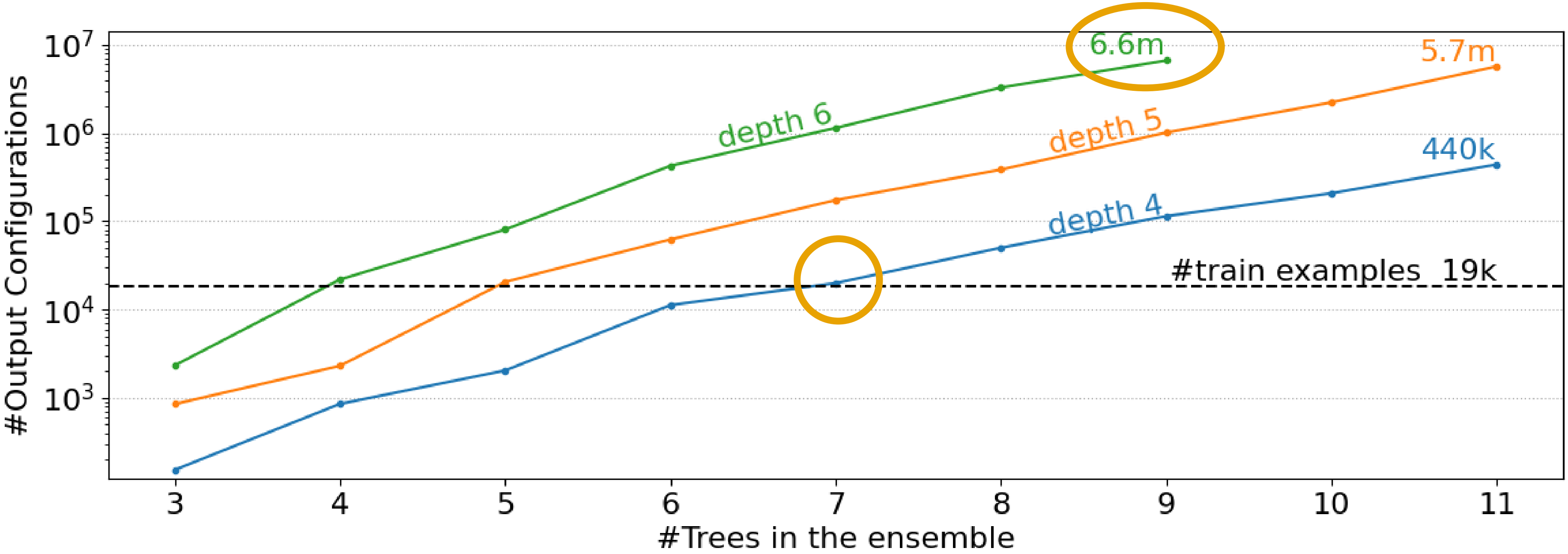
Computed using **Veritas**, XGBoost, California housing; 8 features, 19k examples

OC-SPACE Explodes

Vast majority of OCs never visited by a training example

Computed using **Veritas,** XGBoost, California housing; 8 features, 19k examples

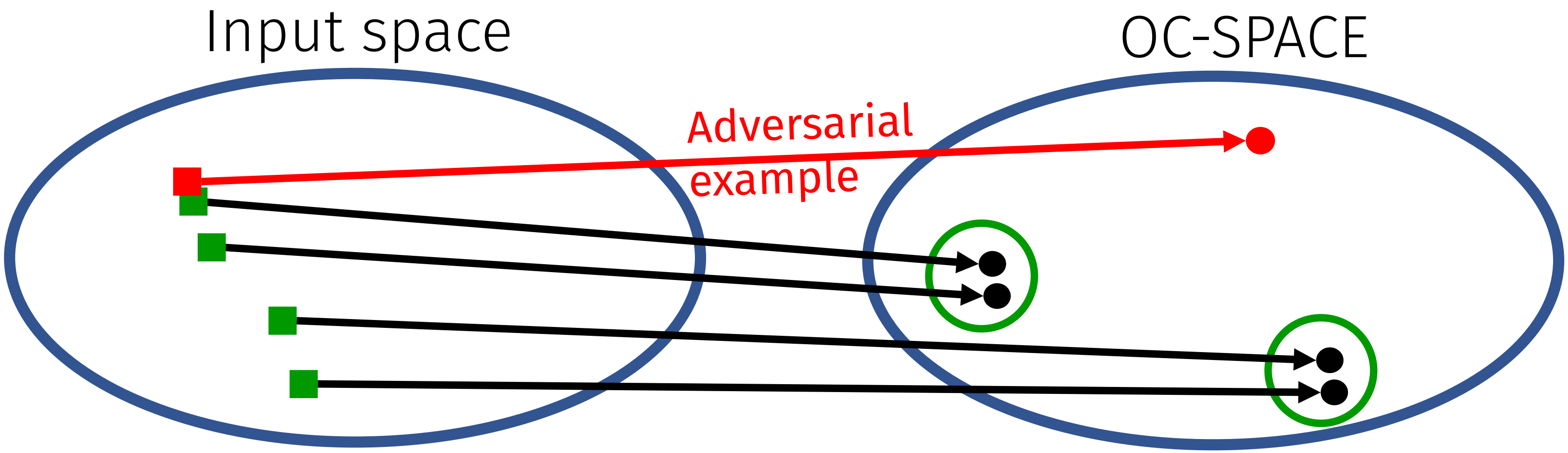
OC-SPACE Explodes



Vast majority of OCs **never visited** by a training example

Computed using **Veritas,** XGBoost, California housing; 8 features, 19k examples

OC-SPACE separates *normal* and *adversarial*



Adversarial example **close to *normal* example in input space**, but **far apart in OC-space**

OC-SCORE Algorithm:

Measuring an example's *adversarialness*

OC-SCORE Algorithm:
Measuring an example's *adversarialness*

- Assign each leaf node an identifier

OC-SCORE Algorithm:

Measuring an example's *adversarialness*

- Assign each leaf node an identifier
- Encode reference set examples by their identifiers: R

OC-SCORE Algorithm:

Measuring an example's *adversarialness*

- Assign each leaf node an identifier
- Encode reference set examples by their identifiers: R
- Post deployment when you receive an instance $\boldsymbol{x}$

OC-SCORE Algorithm:

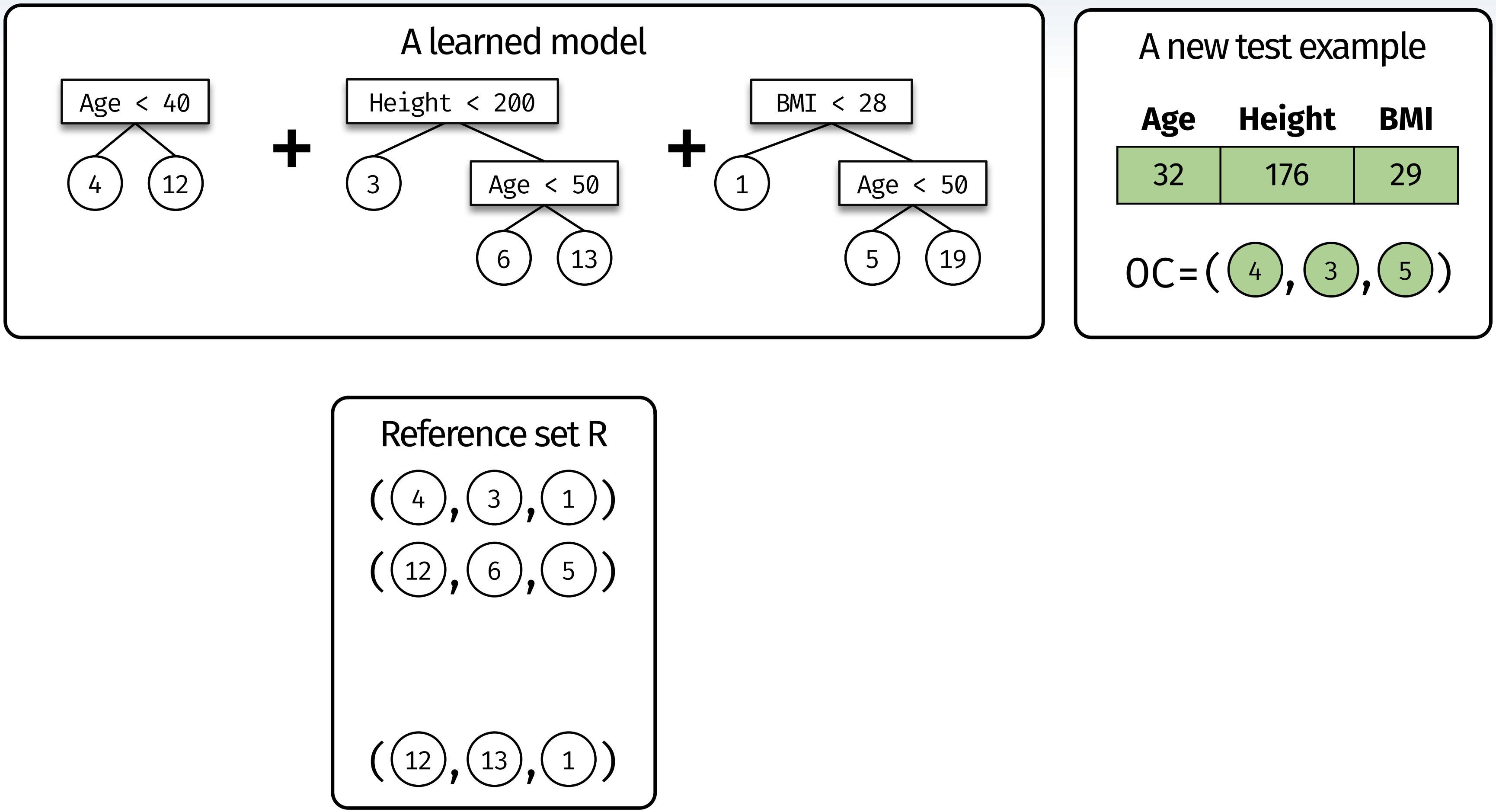
Measuring an example's *adversarialness*

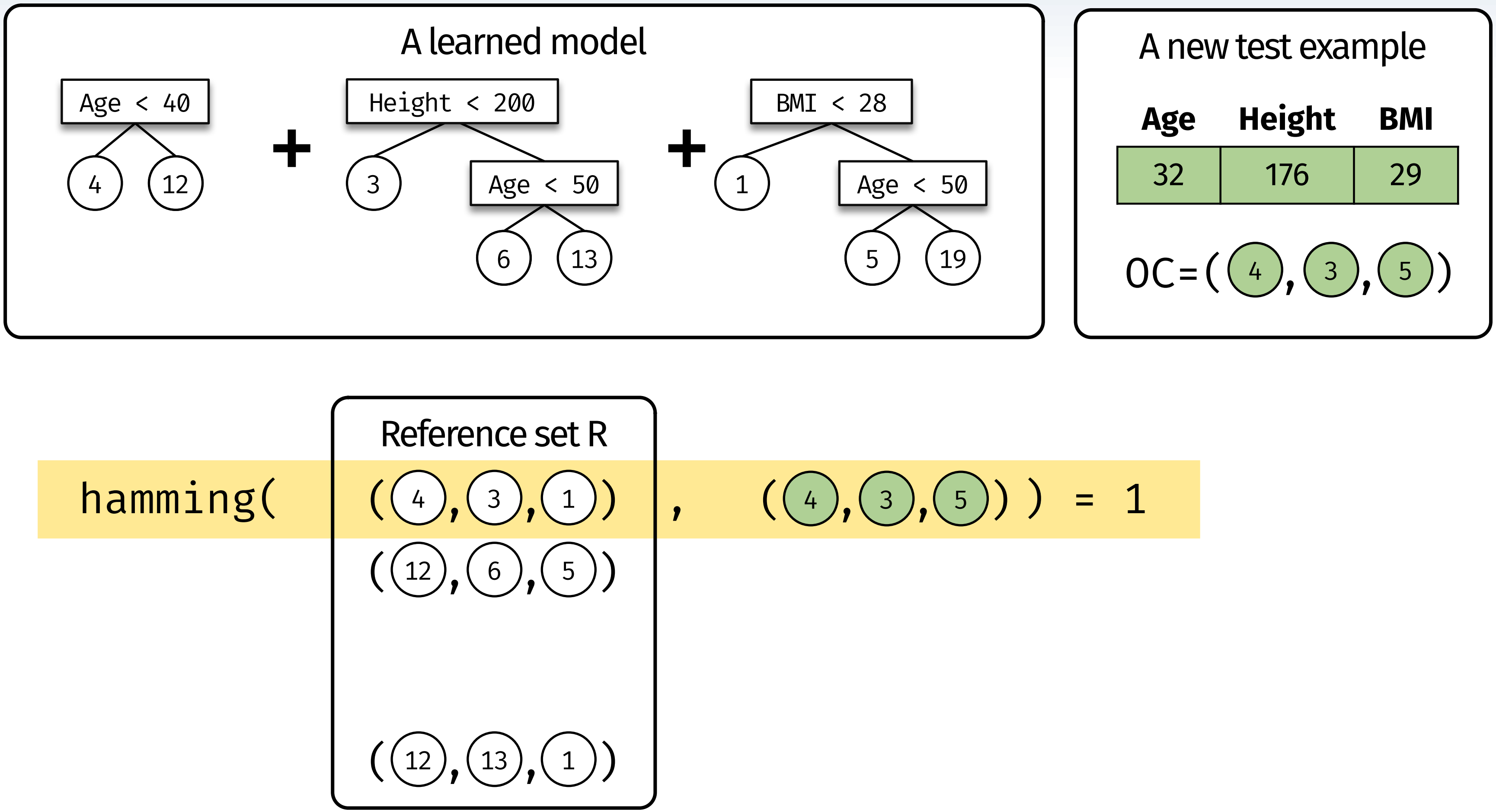
- Assign each leaf node an identifier
- Encode reference set examples by their identifiers: R
- Post deployment when you receive an instance $\boldsymbol{x}$
  - Execute ensemble to encode its reached identifiers: $\mathrm{OC}(\boldsymbol{x})$

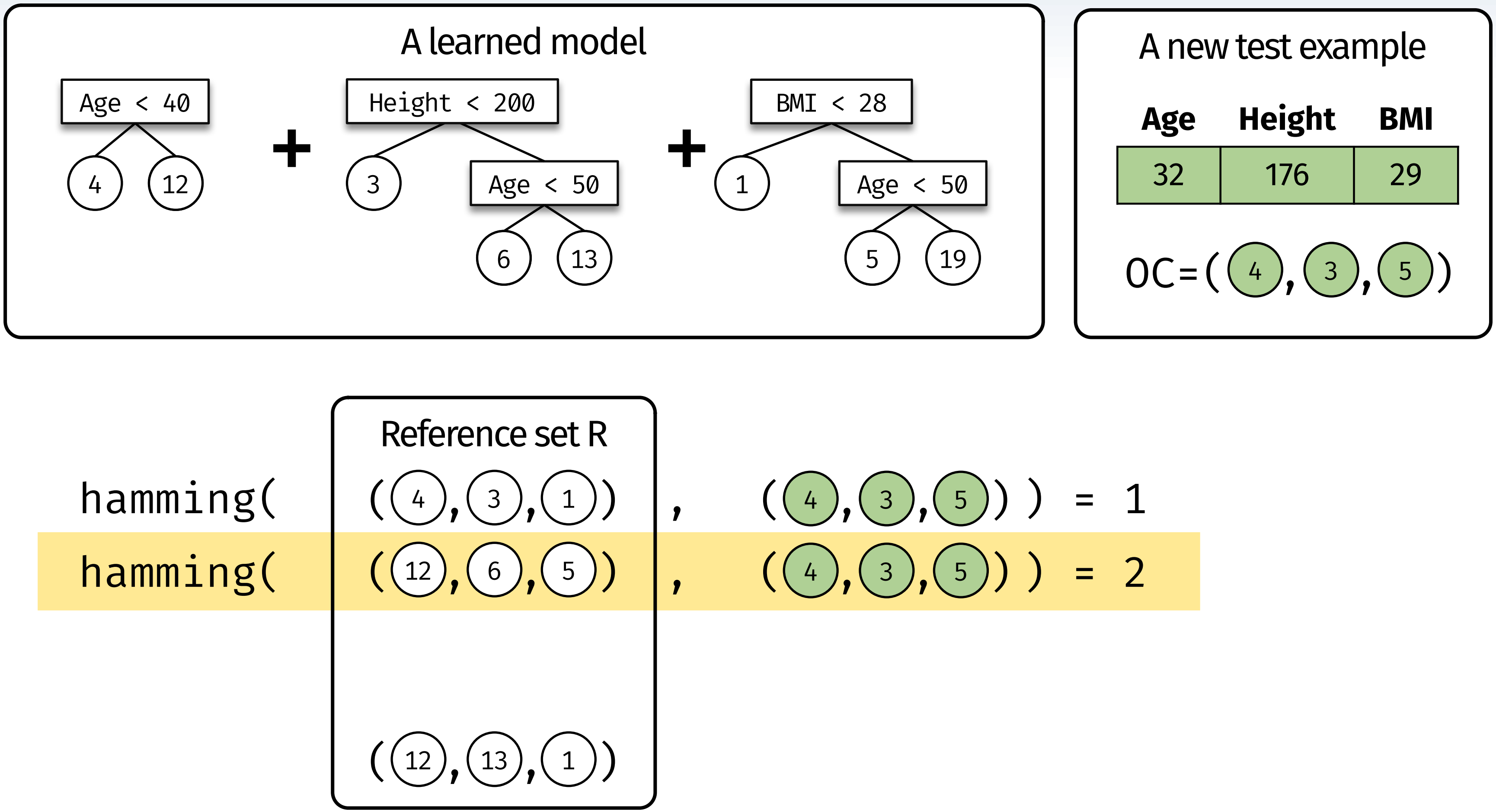
OC-SCORE Algorithm:

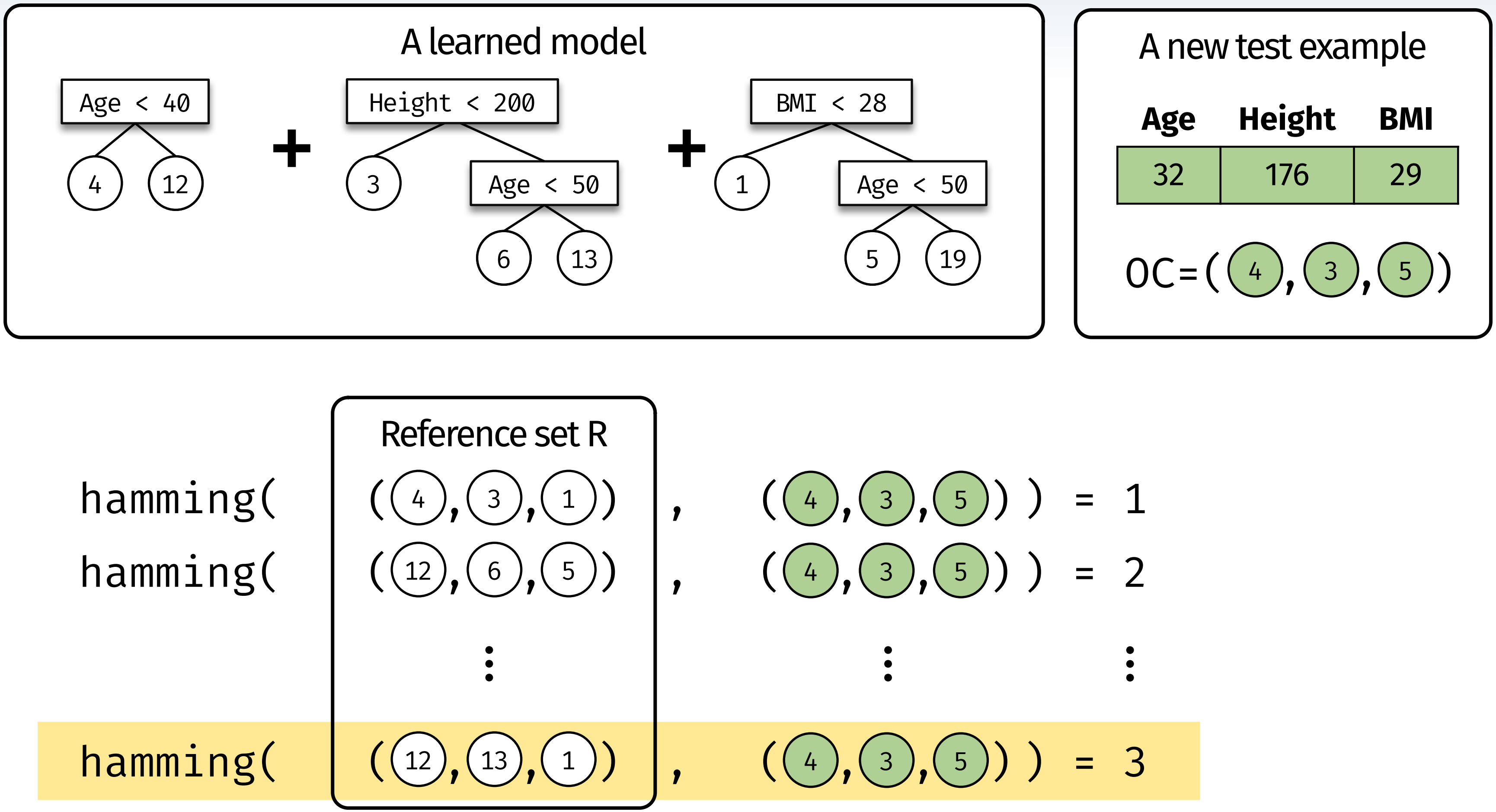
Measuring an example's *adversarialness*

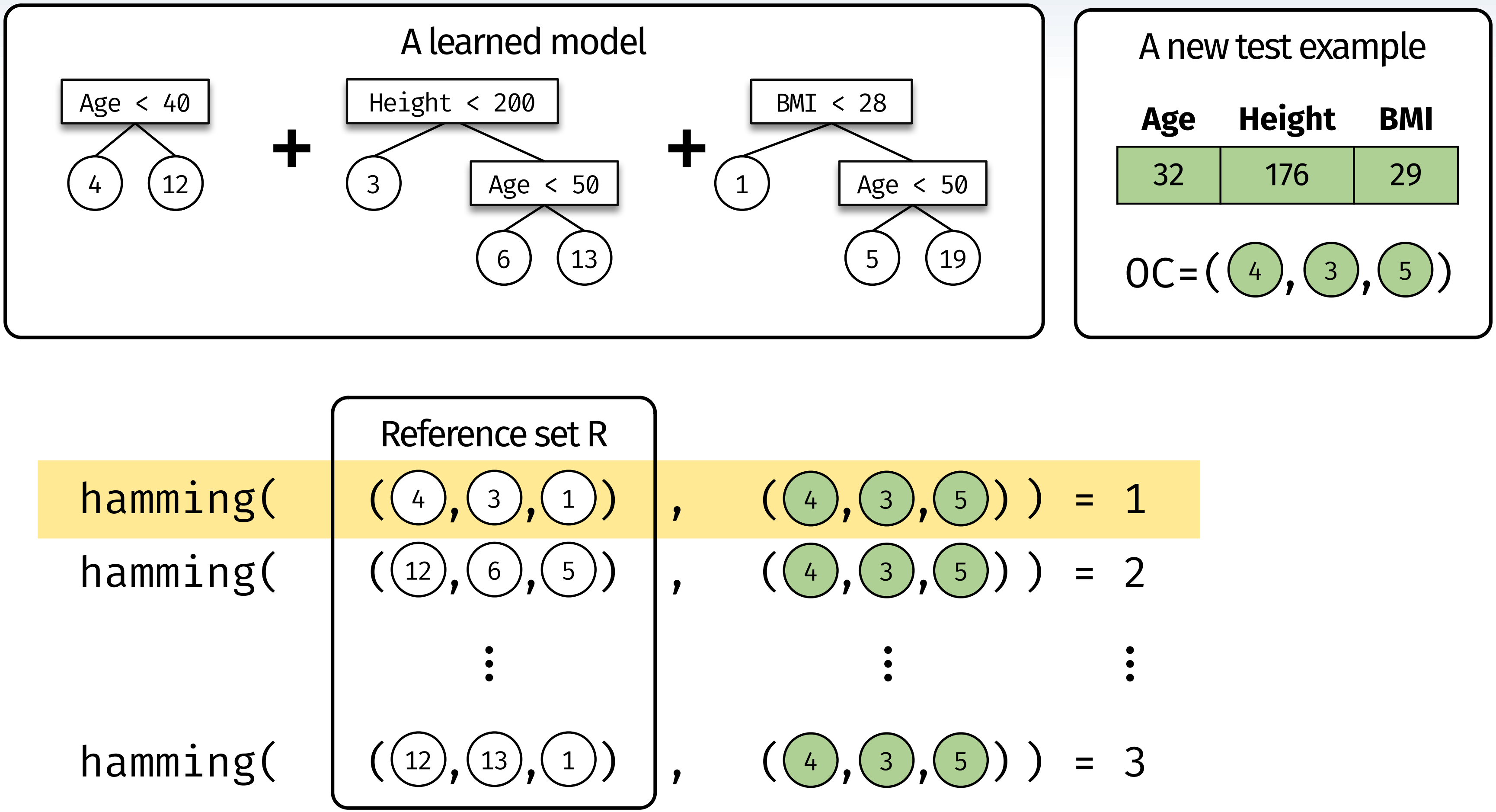
- Assign each leaf node an identifier
- Encode reference set examples by their identifiers: R
- Post deployment when you receive an instance $\boldsymbol{x}$
  - Execute ensemble to encode its reached identifiers: $\mathrm{OC}(\boldsymbol{x})$
  - Compute $\min\{\underbrace{\mathrm{hamming}(\mathrm{OC}(\boldsymbol{x}), \mathrm{OC}(\boldsymbol{x}'))}_{\text{Count how many leaves differ}} \mid \underbrace{\boldsymbol{x}' \in R}_{\text{Normal examples}}\}$


A learned model
Age < 40
4
12
+
Height < 200
3
Age < 50
6
13
+
BMI < 28
1
Age < 50
5
19
A new test example
Age
Height
BMI
32
176
29
OC=(4, 3, 5)
Reference set R
(4, 3, 1)
(12, 6, 5)
(12, 13, 1)


A learned model
Age < 40
4
12
+
Height < 200
3
Age < 50
6
13
+
BMI < 28
1
Age < 50
5
19
A new test example
Age
Height
BMI
32
176
29
OC=(4,3,5)
Reference set R
hamming((4,3,1) , (4,3,5)) = 1
(12,6,5)
(12,13,1)


A learned model
Age < 40
4
12
+
Height < 200
3
Age < 50
6
13
+
BMI < 28
1
Age < 50
5
19
A new test example
Age
Height
BMI
32
176
29
OC=(4,3,5)
Reference set R
hamming((4,3,1) , (4,3,5)) = 1
hamming((12,6,5) , (4,3,5)) = 2
(12,13,1)


A learned model
Age < 40
4
12
+
Height < 200
3
Age < 50
6
13
+
BMI < 28
1
Age < 50
5
19
A new test example
Age
Height
BMI
32
176
29
OC=(4, 3, 5)
Reference set R
hamming((4, 3, 1) , (4, 3, 5)) = 1
hamming((12, 6, 5) , (4, 3, 5)) = 2
hamming((12, 13, 1) , (4, 3, 5)) = 3


A learned model
Age < 40
4
12
+
Height < 200
3
Age < 50
6
13
+
BMI < 28
1
Age < 50
5
19
A new test example
Age
Height
BMI
32
176
29
OC=(4,3,5)
Reference set R
hamming((4,3,1) , (4,3,5)) = 1
hamming((12,6,5) , (4,3,5)) = 2
⋮
hamming((12,13,1) , (4,3,5)) = 3

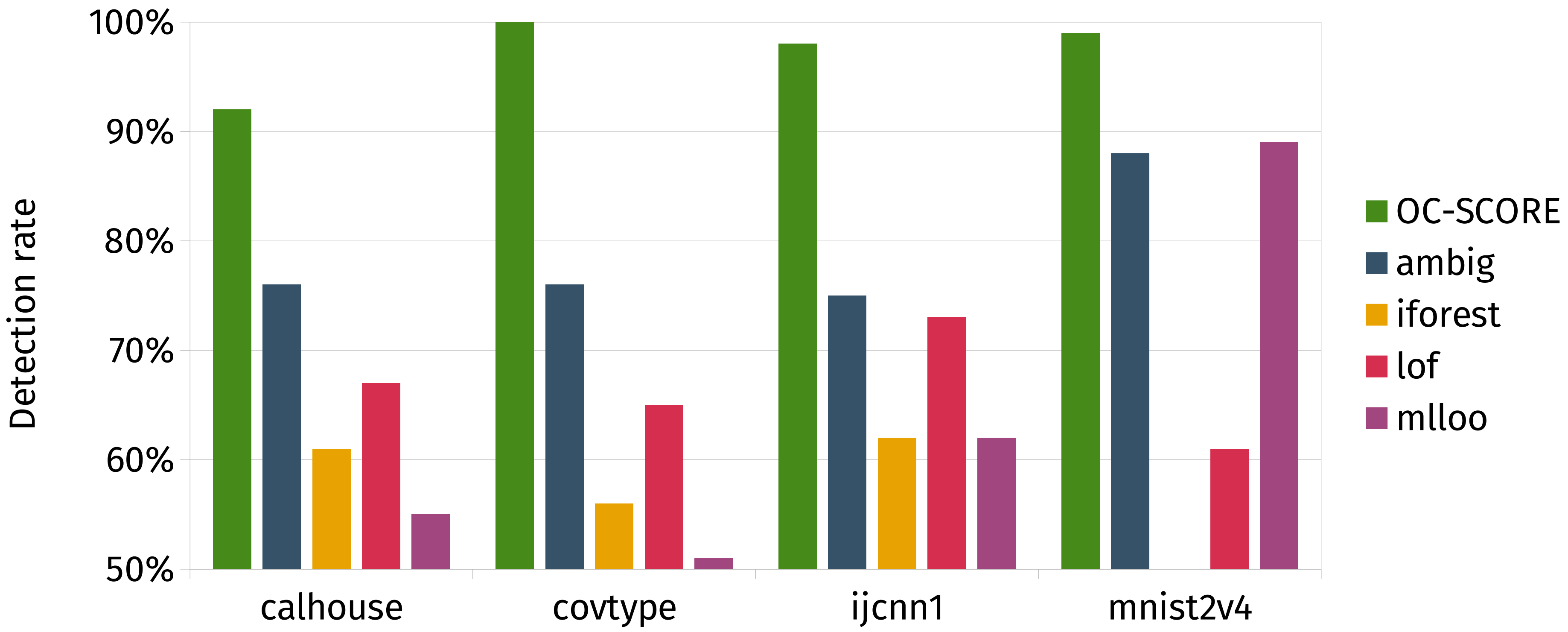
Experimental Setup

How well do we detect adversarial examples?

- **Task:** Distinguish adversarial from normal examples
- **8 dataset**: 4×500 adversarial vs. 2000 normal, 4 adversarial generation methods
- Compare **OC-SCORE** to 4 baselines
  - How accurately can the approaches distinguish between normal and adversarial examples?
  - Does **OC-SCORE** work on real-world data?

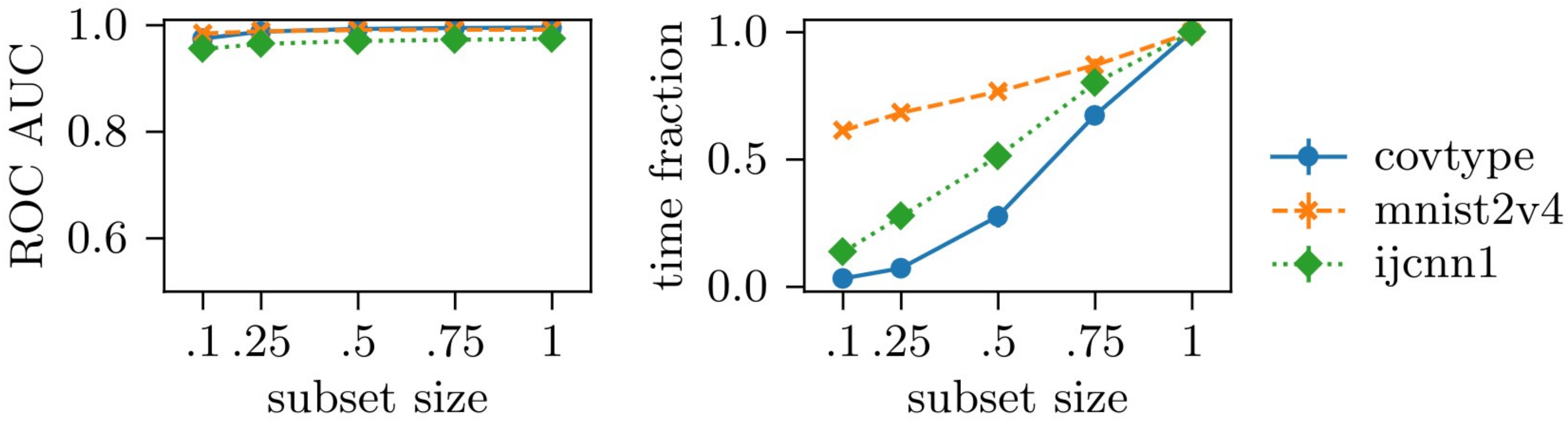
Experimental Results

Good Detection Rate



Reference Set *R* Need Not Be Large

- Random subsets of set of correctly classified training examples
- **Detection performance barely affected**



Applying **OC-SCORE** does not need to be expensive

L. Devos, L. Perini, W. Meert, J. Davis – DTAI, KU Leuven

Questions?

Detecting Evasion Attacks in Deployed Tree Ensembles

Laurens Devos, Lorenzo Perini, Wannes Meert, Jesse Davis
laurens.devos@kuleuven.be
@laudevs