Karlsruhe Institute of Technology

# Leveraging Plasticisty in Incremental Decision Trees

**Marco Heyden*, Heitor Murilo Gomes, Edouard Fouché, Bernhard Pfahringer, Klemens Böhm**

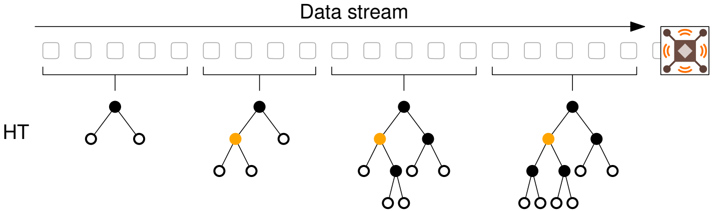
ECML PKDD 2024 | 12. 09. 2024

### Data streams

- A data stream $S = x_1, x_2, \ldots, x_t, \ldots$ is a sequence of observations that arrive over time

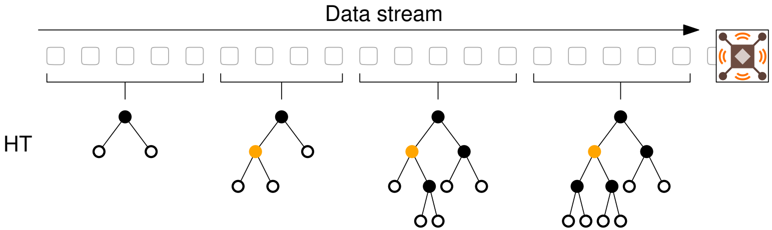
### Incremental decision trees

- Learn from each instance at a time
- Rely on statistical test to decide about node split
- Hoeffding trees (HT) [DH00] and Extremely Fast Decision Trees [MWS18] are popular choices



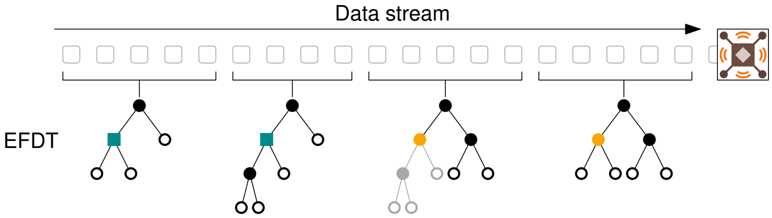
# Hoeffding Trees
## For each leaf, identify best split with high probability

**procedure** HT LEAF NODE
    **for each** new instance $x_i$ arriving at leaf $l$ **do**
        Use $x_i$ to update counters $n_{ijk}$ in $l$ ▷ One counter for each attribute $i$, value $j$, class label $k$
        $a_1 \leftarrow$ attribute with the highest $IG$
        $a_2 \leftarrow$ attribute with the 2nd-highest $IG$
        **if** $IG(a_1) > IG(a_2)$ with probability $> 1 - \delta$ **then** ▷ Requires many data points until a split is performed
            Split $l$ at $a_1$



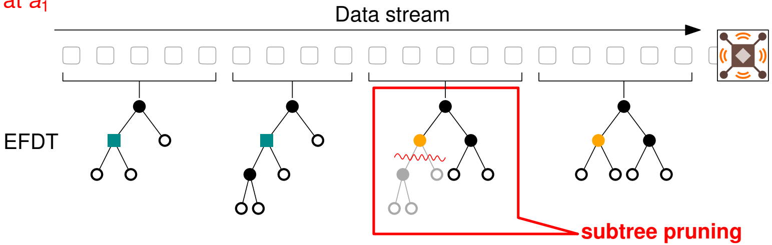
# Extremely Fast Decision Trees
## Relaxed split criterion $\rightarrow$ learn from less data

**procedure** EFDT LEAF NODE
  **for each** new instance $x_i$ arriving at leaf node $l$ **do**
    Use $x_i$ to update counters in $l$
    $a_1 \leftarrow$ attribute with the highest $IG$
    **if** $IG(a_1) > 0$ with probability $> 1 - \delta$ **then** ▷ Relaxed splitting criterion
      Split $l$ at $a_1$



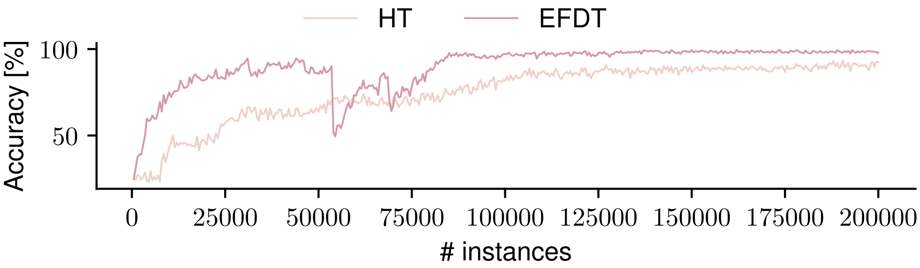
# Extremely Fast Decision Trees
## Reevaluate if split was optimal

**procedure** EFDT INTERNAL NODE
  **for each** new instance $x_i$ arriving at internal node $s$ **do**
    Use $x_i$ to update counters in $s$
    $a_1 \leftarrow$ attribute with the highest $IG$
    $a_c \leftarrow$ current split attribute
    **if** $IG(a_1) > IG(a_c)$ with probability $> 1 - \delta$ **then** ▷ Split revision: prune subtree if better split was found
      Remove subtree below $s$
      Split $s$ at $a_1$



# HT vs. EFDT
## EFDT learns faster than HT but suffers from accuracy drops

- Toy example on synthetic data (details in the paper)

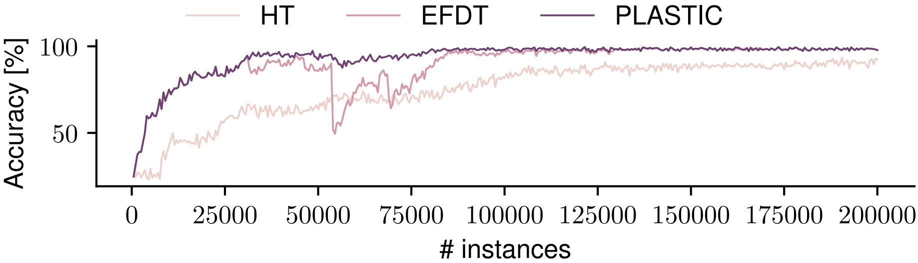



**Can we maintain EFDT's fast learning but avoid the accuracy drops?**

# Our Approach
## PLASTIC

Karlsruhe Institute of Technology

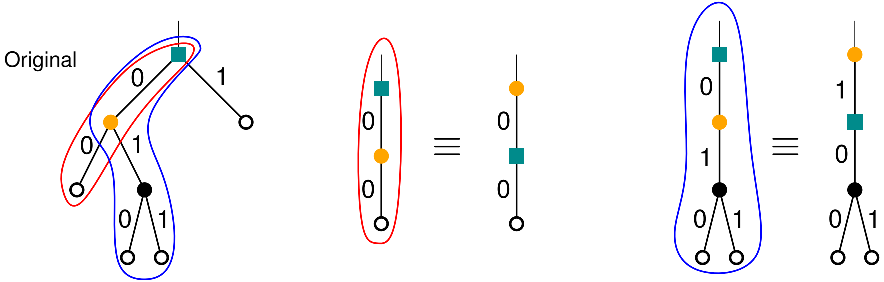
**Yes! By exploiting *decision tree plasticity* during split reevaluation.**



# PLASTIC
## Decision tree plasticity

- Alter tree structure without affecting predictions
- In the left-most branch, any instance with attribute values ■ = 0 and ● = 0 will arrive at ○
- Hence, from the viewpoint of the leaf, ■—●—○ ≡ ●—■—○

⇒ Change the **current split attribute** (■) to the **desired split attribute** (●) by *restructuring* all branches under ■

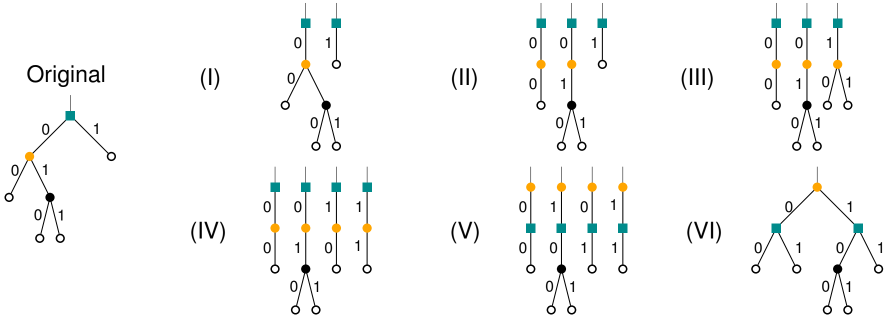


# Algorithm — Decision tree restructuring

I–IV. Decouple the branches of the tree

V. Reorder each branch

VI. Re-build tree



# PLASTIC
## Numerical and binary categorical splits

### Numerical splits

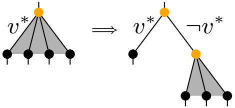
- For numerical splits, split thresholds are typically different

⇒ Adjust split threshold prior to restructuring

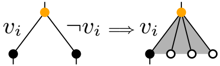
⇒ Remove unreachable subtree

### Binary categorical splits

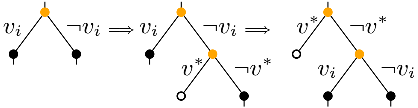
multiway – binary




binary – multiway




binary – binary



# Experiments
## Setup and competitors

### Experiments

**1. Comparison with EFDT**

- Evaluates the effect of **decision tree restructuring**
- Comparison of PLASTIC and EFDT
- We use our own implementation of EFDT (based on the same code as PLASTIC)

**2. Comparison with HT, EFDT and EFHAT**

- Evaluation against **state of the art** decision trees
- We add a simple adaptive version of PLASTIC called PLASTIC-A
  - Trains a background tree when accuracy drops
  - Replaces current tree once it is more accurate

### Data streams

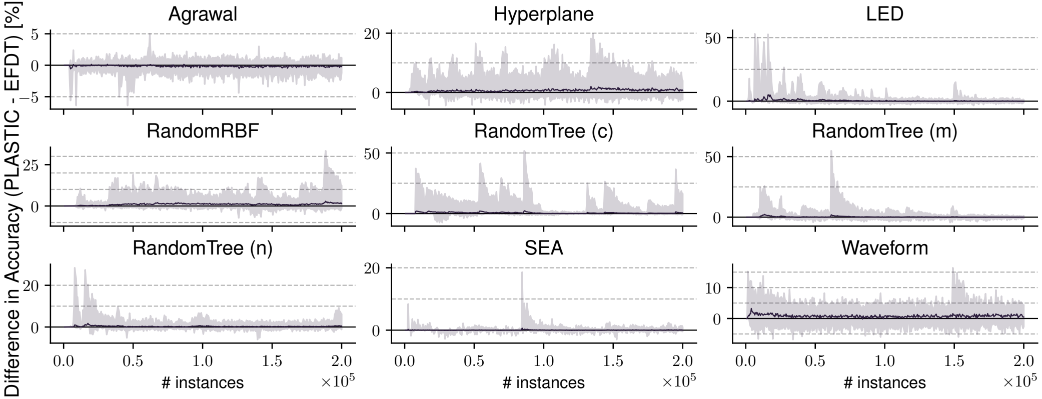
- 9 synthetic, 15 real-world data streams
- 200,000 instances on synthetic data
- Up to 15 million instances on real world data

### Evaluation methodology

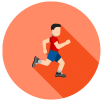
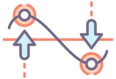
- Test-then-train evaluation
- Accuracy in sliding window of size 500 (synthetic data) and 1000 (real world data)

# Experiments
## Comparison to EFDT (synthetic data)

- Graphs show difference in accuracy between PLASTIC and EFDT
- Shaded area shows maximum difference across experiment repetitions



# Experiments
## Results on real-world data streams

| Approach | HT | EFDT | EFHAT | PLASTIC | PLASTIC-A | NC |
|---|---|---|---|---|---|---|
| RIALTO | 24.2 | 37.8 | 42.3 | **49.2** | 47.4 | 0.0 |
| SENSORS | 15.8 | 38.2 | 42.7 | **48.1** | 47.1 | 0.1 |
| COVTYPE | 68.3 | 77.4 | 79.6 | **82.1** | 81.3 | 95.1 |
| HARTH | 79.5 | 86.5 | 89.2 | 88.3 | **90.9** | 99.9 |
| PAMAP2 | 58.4 | 94.5 | 98.3 | 96.6 | **98.6** | 99.9 |
| WISDM | 65.6 | 80.6 | 89.0 | 82.6 | **93.1** | 99.9 |
| ... | | | | | | |
| Accuracy | 64.8 | 74.2 | 76.4 | 76.7 | **77.8** | 61.4 |
| Rank | 4.21 | 3.71 | 2.50 | **2.29** | **2.29** | – |
| Runtime | **61.8** | 110.6 | 198.6 | 141.6 | 175.0 | 27.1 |

# Wrapping up

Karlsruhe Institute of Technology

## Summary

- We propose PLASTIC, an **incremental decision tree algorithm** that uses **decision tree plasticity** to overcome forgetting in EFDT
- During split re-evaluation, PLASTIC restructures the otherwise pruned subtree
- Simple adaptive version, PLASTIC-A, starts growing a background tree once a change was detected

## In the paper

- Pseudocode
- Additional experiments

**Paper and code:**

- doi.org/10.1007/978-3-031-70362-1_3
- github.com/heymarco/PLASTIC

Paper

GitHub

# References I

[1] Pedro M. Domingos and Geoff Hulten. "Mining high-speed data streams". In: *KDD*. ACM, 2000, pp. 71–80.

[2] Chaitanya Manapragada, Geoffrey I. Webb, and Mahsa Salehi. "Extremely fast decision tree". In: *KDD*. ACM, 2018, pp. 1953–1962. DOI: 10.1145/3219819.3220005.