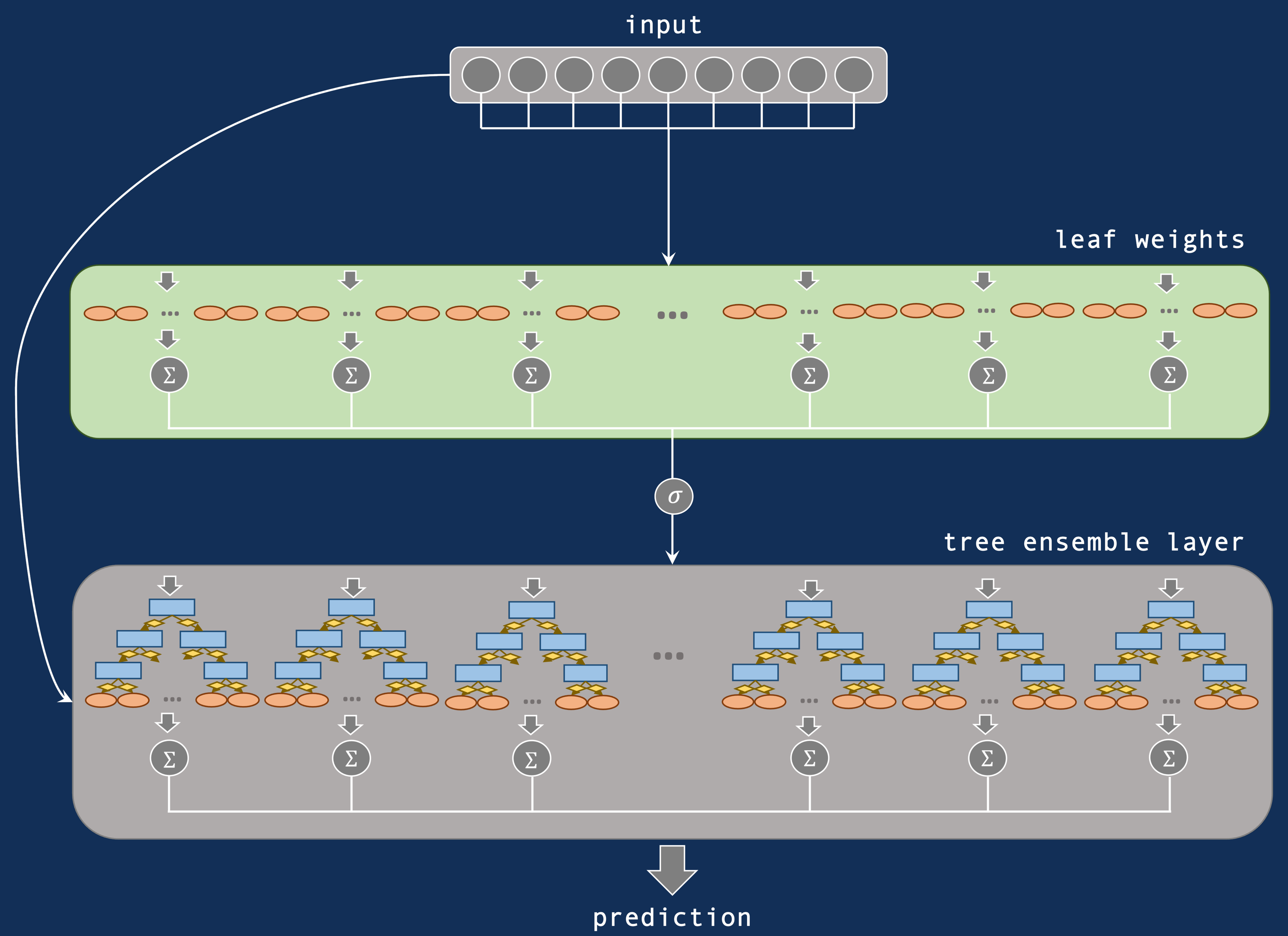
GRANDE: Gradient-Based Decision Tree Ensembles for Tabular Data

Combining a **gradient-based** optimization with the inductive bias of **axis-aligned** splits



https://github.com/s-marton

Motivation

Tabular data is most frequent type of data:

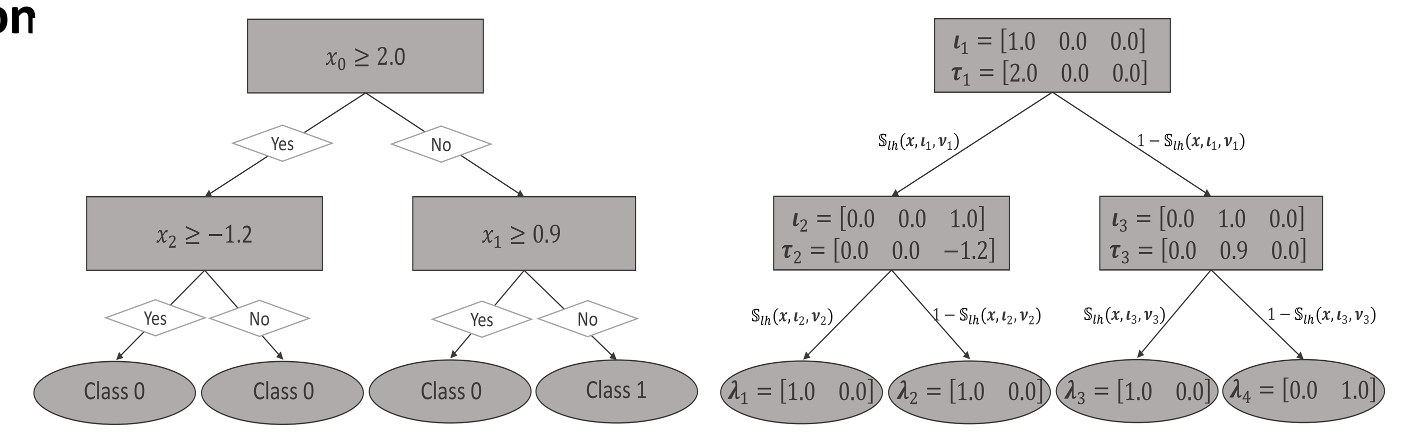
- Comes with several challenges (heterogeneous data, class imbalance,…)
- Last "unconquered castle" for DL methods → GBDTs are SOTA
  - One reason is the inductive bias of axis-aligned splits: Tabular data typically has irregular target functions
    - DL methods favor overly smooth solutions → not well-suited for irregular target functions
    - Tree-based methods learn piecewise-constant functions → well-suited for irregular target functions
- High need for gradient-based methods → Flexibility

→ **GRANDE = gradient-based optimization + inductive bias of axis-aligned splits**

GradTree: Gradient-Based Decision Trees

Dense DT Representation

- Relaxing the split indices and split thresholds
  →Allow reasonable optimization with gradient descent



(a) Vanilla DT Representation (b) Dense DT Representation

Straight-Through Operator for non-differentiable operations

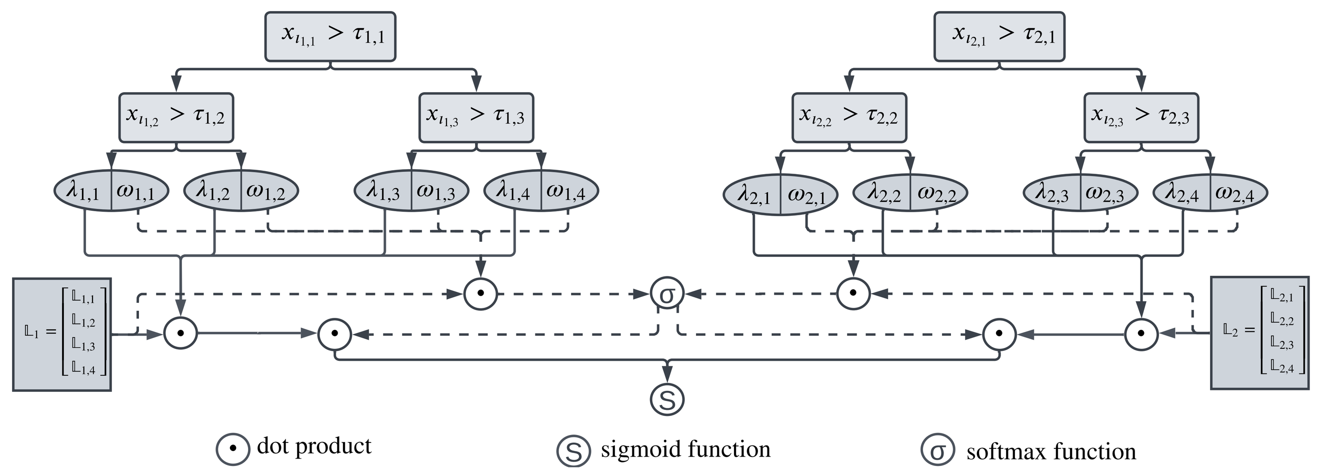
(1) Hardmax function to enforce one-hot encoded split vectors → **univariate, axis-aligned DTs**

(2) Discretization of the split function (round the sigmoid output) → **hard splits**

GRANDE: Gradient-Based Decision Tree Ensembles

(1) Extend GradTree from individual trees to tree ensembles

(2) Introduction of instance-wise estimator weights



- End-to-end learnable weight parameters
- One weight per leaf instead of one weight per estimator
  → Weight is selected based on the path for a given sample

(3) Novel differentiable splitting function: softsign (instead of sigmoid)

→ Better gradient flow & more reasonable gradients

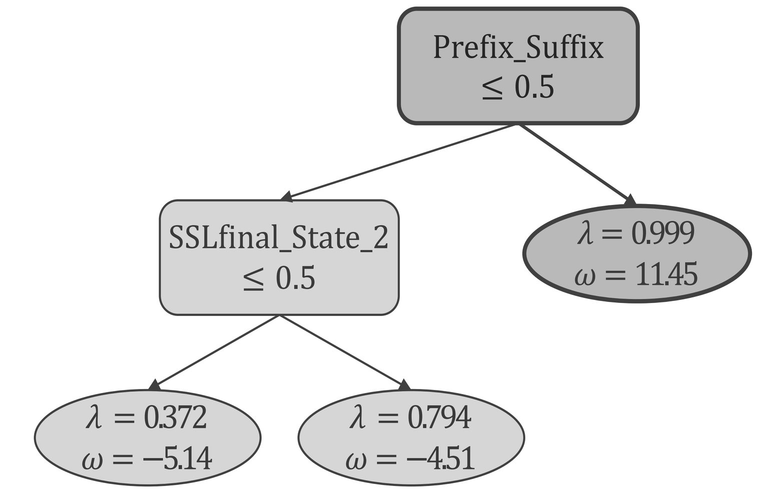
(4) Regularization: feature subset, data subset, dropout,…

→ Combine techniques of tree- and gradient-based methods

Case Study: PhishingWebsites Dataset

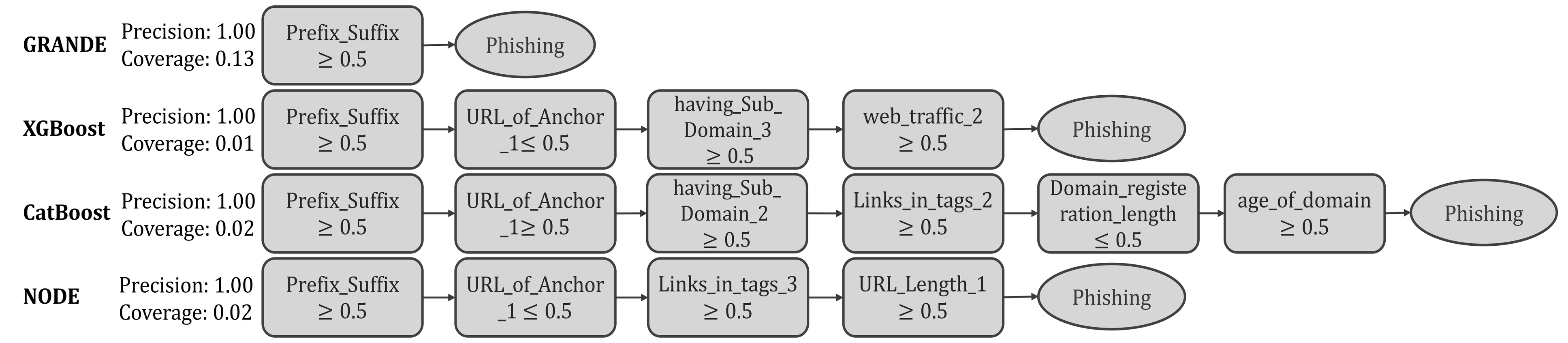
Task: identifying malicious websites based on meta- data and additional observable characteristics

- Large performance gap between simple and complex models
- Simple rules exist, for instance:
  Is a prefix of suffix added to the domain name?
  → Yes: phishing (1)



Highest-Weighted Estimator. This figure visualizes the DT from GRANDE (1024 total estimators) which has the highest weight (normalized weight = 0.94) for an exemplary instance

- By assessing the instance-wise weights for an exemplary instance with Prefix_Suffix = 1, we can observe that the class predicted by the ensemble is derived completely from the depicted tree
  → GRANDE has learned a simple representation for a simple rule



Anchors Explanations. This figure shows the local explanations generated by Anchors for the given instance. The explanation for GRANDE only comprises a single rule. In contrast, the corresponding explanations for the other methods have significantly higher complexity, which indicates that these methods are not able to learn simple representations within a complex model.

Benchmark Results

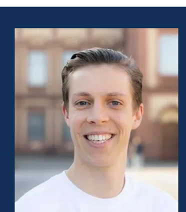
Performance Comparison. We report the test macro F1-score (mean for a 5-fold CV) with optimized parameters. The datasets are sorted based on the data size

	GRANDE	XGB	CatBoost	NODE
dresses-sales	**0.612 (1)**	0.581 (3)	0.588 (2)	0.564 (4)
climate-simulation-crashes	**0.853 (1)**	0.763 (4)	0.778 (3)	0.802 (2)
cylinder-bands	**0.819 (1)**	0.773 (3)	0.801 (2)	0.754 (4)
wdbc	**0.975 (1)**	0.953 (4)	0.963 (3)	0.966 (2)
ilpd	**0.657 (1)**	0.632 (3)	0.643 (2)	0.526 (4)
tokyo1	0.921 (3)	0.915 (4)	**0.927 (1)**	0.921 (2)
qsar-biodeg	**0.854 (1)**	0.853 (2)	0.844 (3)	0.836 (4)
ozone-level-8hr	**0.726 (1)**	0.688 (4)	0.721 (2)	0.703 (3)
madelon	0.803 (3)	0.833 (2)	**0.861 (1)**	0.571(4)
Bioresponse	0.794 (3)	0.799 (2)	**0.801 (1)**	0.780 (4)
wilt	0.936 (2)	0.911 (4)	0.919 (3)	**0.937 (1)**
churn	0.914 (2)	0.900 (3)	0.869 (4)	**0.930 (1)**
phoneme	0.846 (4)	0.872 (2)	**0.876 (1)**	0.862 (3)
SpeedDating	**0.723 (1)**	0.704 (4)	0.718 (2)	0.707 (3)
PhishingWebsites	**0.969 (1)**	0.968 (2)	0.965 (4)	0.968 (3)
Amazon_employee_access	0.665 (2)	0.621 (4)	**0.671 (1)**	0.649 (3)
nomao	0.958 (3)	**0.965 (1)**	0.964 (2)	0.956 (4)
adult	0.790 (4)	**0.798 (1)**	0.796 (2)	0.794 (3)
numerai28.6	**0.519 (1)**	0.518 (3)	0.519 (2)	0.503 (4)
Normalized Mean ↑	**0.776 (1)**	0.483 (3)	0.671 (2)	0.327 (4)
Mean Reciprocal Rank ↑	**0.702 (1)**	0.417 (3)	0.570 (2)	0.395 (4)

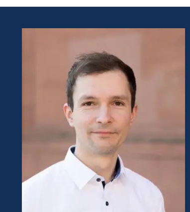
GRANDE in Action

- Easy-to-use implementation: **pip install GRANDE**

```
from GRANDE import GRANDE

params = {
        'depth': 5,
        'n_estimators': 512,
        'loss': 'crossentropy',
}

args = {
    'objective': 'binary',
}

model_grande = GRANDE(params=params, args=args)

model_grande.fit(X_train=X_train,
                 y_train=y_train,
                 X_val=X_valid,
                 y_val=y_valid)

preds_grande = model_grande.predict(X_test)
```

Sascha Marton
sascha.marton@uni-mannheim.de
University of Mannheim



Jun.-Prof. Dr. Stefan Lüdtke
stefan.luedtke@uni-rostock.de
Institute for Visual and Analytic Computing

Dr. Christian Bartelt
christian.bartelt@uni-mannheim.de
University of Mannheim

Prof. Dr. Heiner Stuckenschmidt
heiner.stuckenschmidt@uni-mannheim.de
University of Mannheim