DOFEN: Deep Oblivious Forest Ensemble

Kuan-Yu Chen[1], Ping-Han Chiang[1], Hsin-Rung Chou[1], Chih-Sheng Chen[1], Tien-Hao Chang[1 2]

1 SinoPac Holdings, Taipei, Taiwan
2 Department of Electronic Engineering, National Cheng Kung University, Tainan, Taiwan

Background and Motivation

- **<u>What's missing in current Deep Tabular Neural Networks (DTNN) ?</u>**

  => Sparse selection of columns in tree-based models

  - Only a limited number of features are used when constructing each tree
  - This Increases feature diversity and helps mitigate overfitting

- **<u>Existing DTNNs cannot achieve "sparse selection of columns"</u>**

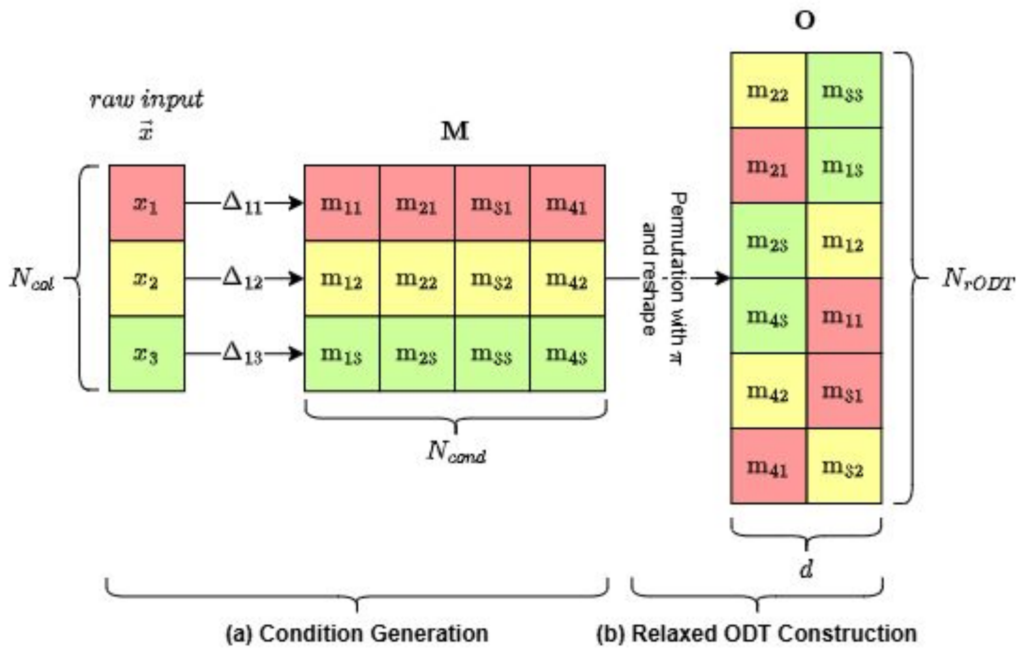
  => Directly generate a sparse matrix for on-off column selection is non-differentiable

  - Attention-based models (e.g. SAINT, FT-Transformer, Trompt) uses softmax result in dense selection of columns
  - Tree-inspired networks (e.g. TabNet and NODE) uses entmax or sparsemax to enhance sparsity but still only achieve near-sparse effect

DOFEN proposes a novel two-step workaround process

[STEP 1] <u>Enumerating as many sparse selections of columns as possible</u>

=> Condition Generation and Construct relaxed Oblivious Decision Trees (rODTs)



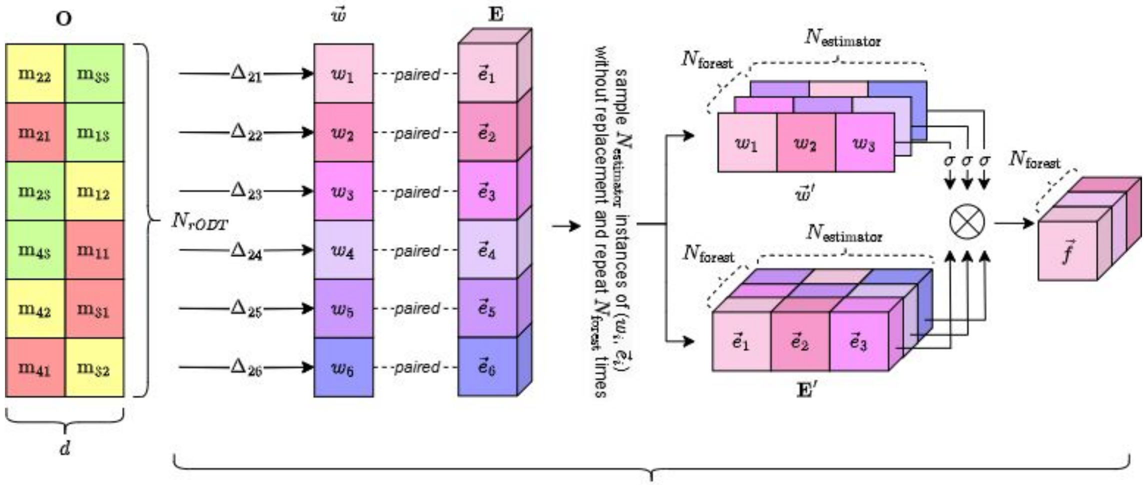
1. Generate $\boldsymbol{N_{cond}}$ conditions for each column
2. Random select ***d*** conditions to form an depth ***d*** rODT
3. Will generate a total of $\boldsymbol{N_{rODT}}$ rODTs

Each rODT can be refer to sparse selections of columns, as each rODT uses only ***d*** columns

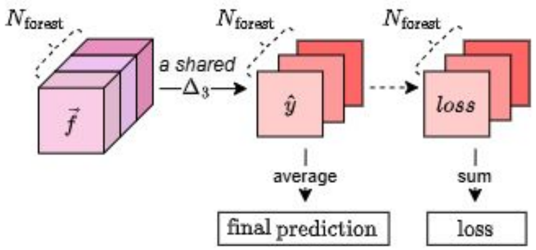
DOFEN proposes a novel two-step workaround process

[STEP 2] Weighting the importance of these sparse selections and aggregate them

=> Two-level rODT Forest Ensemble



(c) Forest Construction



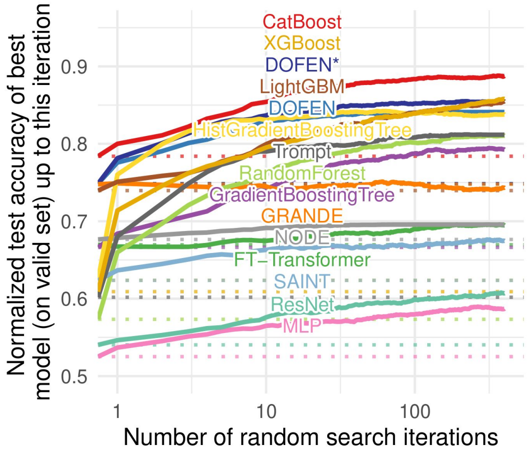
Level 1: Forest Construction

1. Each rODT goes through its own weighting network Δ_2
2. Randomly aggregate $N_{estimator}$ rODTs to form an rODT forest
3. Will conduct a total of N_{forest} rODT forests

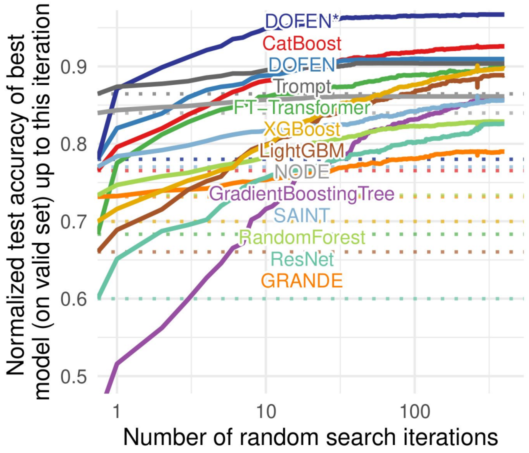
Level 2: Forest Ensemble

1. Each rODT forest gives prediction through a shared network
2. Calculate loss for each forest individually
3. Average forest predictions to form a final prediction

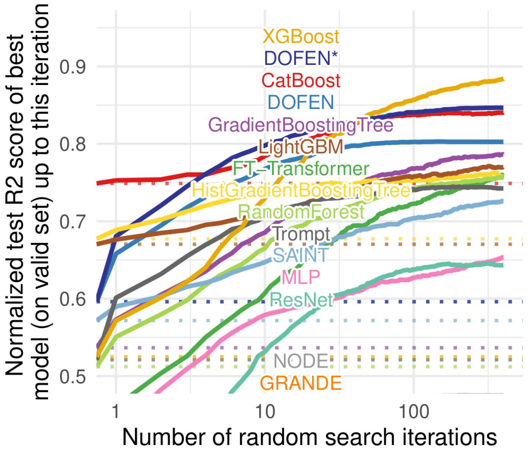
DOFEN reaches SOTA performance on Tabular Benchmark



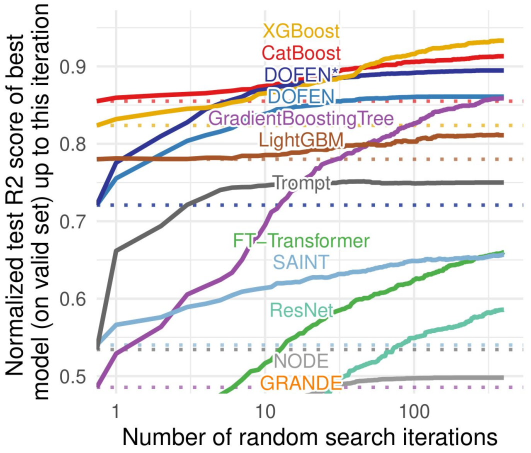
(a) Medium, Classification
(23 datasets)



(b) Large, Classification
(6 datasets)



(a) Medium, Regression
(36 datasets)



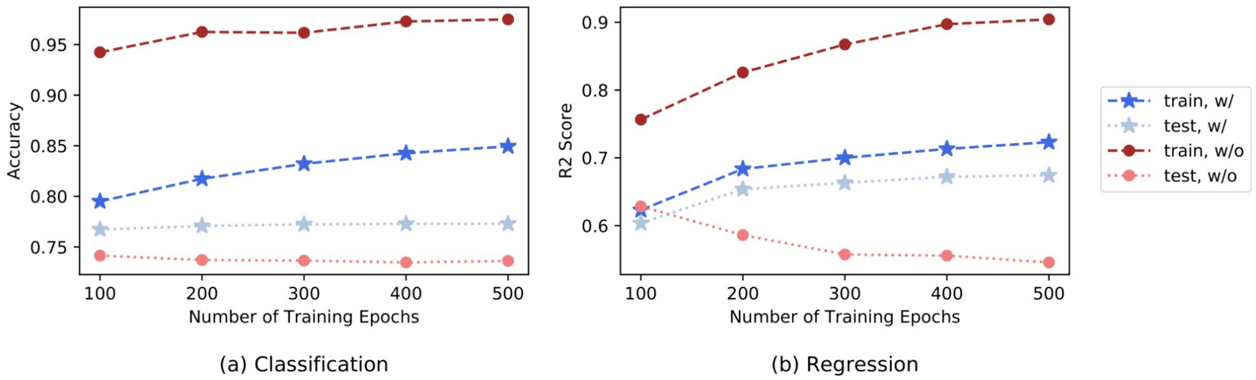
(a) Large, Regression
(8 datasets)

1. DOFEN is comparable to advanced Boosting Trees and sometimes surpasses them
2. DOFEN beats previous SOTA NN (e.g. FT-Transformer and Trompt)
3. DOFEN beats other tree-inspired NN (e.g. NODE and GRANDE)
4. DOFEN* is a multi-head extension of DOFEN, it shows even better performance !!!
   (this result will be added in the future version of our paper)

Two-Level rODT Forest Ensemble enhances performance and stability

<u>Sample rODTs to form forests mitigates overfitting and enchance performance</u>

	w/ sampling	w/o sampling
Classifcation	77.25	73.62
- numerical only	79.20	75.26
- heterogeneous	72.81	69.88
Regression	66.05	32.38
- numerical only	68.14	18.67
- heterogeneous	63.71	47.70



Performance drops drastically if we conduct a single rODT forest using all possible rODTs

The reason for the performance drop is overfitting

Two-Level rODT Forest Ensemble enhances performance and stability

Increase number of rODT forest enhance stability and performance

N_{forest}	1	10	20	50	**100 (default)**	400
Jannis (het-cls)	73.82 (0.60)	77.47 (0.19)	77.82 (0.15)	78.00 (0.06)	78.08 (0.07)	**78.14 (0.04)**
road-sofety (num-cls)	75.17 (1.18)	77.12 (0.10)	77.20 (0.07)	77.28 (0.04)	**77.32 (0.05)**	**77.32 (0.03)**
delay-zurich (het-rgr)	0.54 (0.33)	2.48 (0.09)	2.58 (0.03)	2.65 (0.03)	2.68 (0.03)	**2.70 (0.02)**
abalone (num-rgr)	54.69 (1.81)	58.10 (0.38)	58.46 (0.26)	58.62 (0.17)	58.68 (0.10)	**58.70 (0.04)**

1. Performance std are already small when N_{forest} = 1, and becomes even smaller when N_{forest} increase
2. Increase N_{forest} also improves performance

The decision making process of DOFEN is Interpretable

<u>We use the weights of rODTs to calculate a sample's feature importance</u>

1. Calculate how often a column is used by an rODT (conditions of columns form an rODT)
2. Weighted sum these column frequency by their corresponding weights $\mathbf{w}_i$ (from $\mathbf{\Delta}_2$)

	1st	2nd	3rd
Random Forest	alcohol (24.22%)	volatile acidity (12.44%)	free sulfur dioxide (11.78%)
XGBoost	alcohol (31.87%)	free sulfur dioxide (11.38%)	volatile acidity (10.05%)
CatBoost	alcohol (17.34%)	volatile acidity (12.07%)	free sulfur dioxide (11.47%)
Trompt	fixed acidity (10.91%)	volatile acidity (10.47%)	pH (10.37%)
DOFEN	alcohol (10.90%)	free sulfur dioxide (10.21%)	volatile acidity (10.01%)

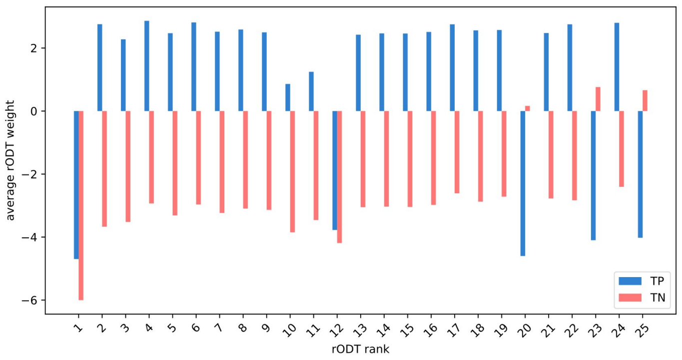
The feature importance ranked by DOFEN aligned closely with the ones ranked by tree-based models (table shows the result on white wine dataset)

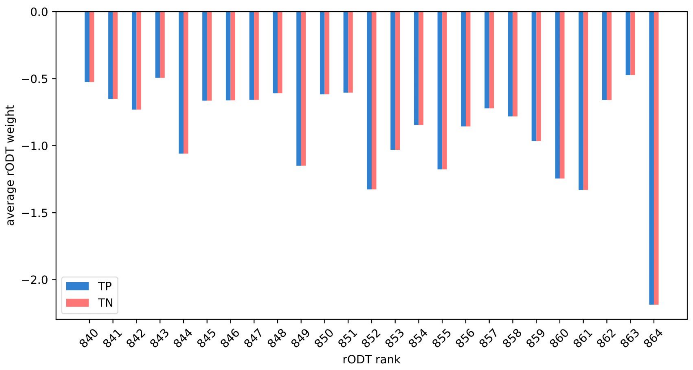
The decision making process of DOFEN is Interpretable

<u>rODTs with larger weight variation across samples are more crucial</u>

- rODTs with larger weight variantion activates differently on TP and TN samples
- rODTs with lower weight variantion shows same activation on all samples (i.e. redundant rODTs)

Activation of rODTs with larger weight variation





Activation of rODTs with smaller weight variation

- Carefully pruning "redundant" rODTs does not negative harm DOFEN's performance

Prune Ratio	0.0 (default)	0.02	0.1	0.2	by dataset
Classification	77.25	**<u>77.33</u>**	77.26	77.09	77.32
Regression	66.05	66.29	66.30	66.21	**<u>66.57</u>**

Prune P% of rODTs with smallest weight variation

Limitation and Conclusion

Limitation

1. **The inference time of DOFEN is relatively long**
   mainly caused by the group convolution operation for calculating weights for each rODT (this has already be solved in our official implementation on github)

2. **Training epochs of DOFEN are relatively large**
   randomization steps involved in DOFEN result in a slower convergence speed

Conclusion

1. DOFEN is a novel tree-inspired Tabular DNN that achieves on-off sparse selections of columns

2. DOFEN achieves SOTA results on the Tabular Benchmark, beating previous DNN-based models and is comparable to boosting tree methods

3. DOFEN's outstanding performance gives it the potential to serve as the backbone model for tabular data across various scenarios (e.g. self-supervised learning, multi-modal training)

Thank You