TABULAR DATA: DEEP LEARNING IS NOT ALL YOU NEED

Yarden Fogel

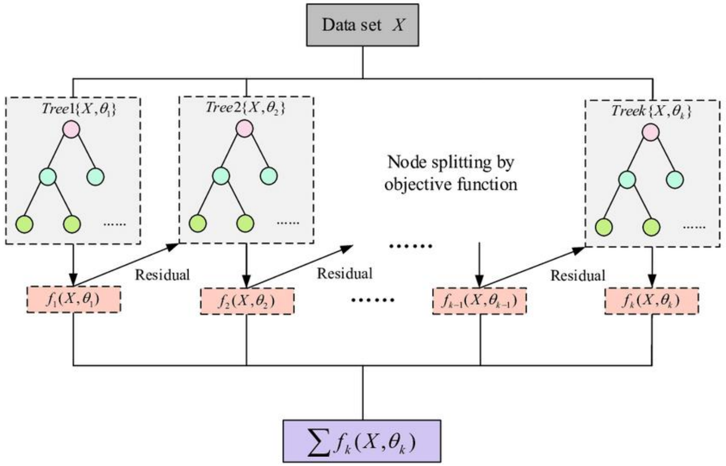
Roni Ben Dom

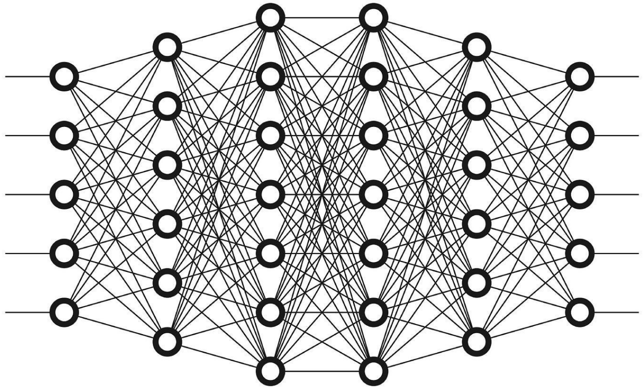
AGENDA

Background

- Ensemble algorithms
- Gradient boosting
- XGBoost

Paper Review





INTRODUCTION

The paper's main purpose is to compare performance and resource-requirements of recently-proposed and highly-touted deep learning models for tabular datasets against XGBoost and ensemble approaches, and to establish tabular data best practices.

RECAP – ENSEMBLE LEARNING

Ensemble learning is a technique that combines the decisions from multiple models in order to to improve overall performance

RECAP – GRADIENT BOOSTING

Gradient boosting is one of the variants of ensemble methods where multiple weak models are combined to improve performance.

By iteratively improving upon the mistakes made by previous models, gradient boosting creates a powerful ensemble model that can capture complex patterns and make accurate predictions. It is known for its effectiveness in handling both numerical and categorical data, as well as its ability to handle missing values.

XGBOOST

XGBoost (eXtreme Gradient Boosting) is an optimized and highly-efficient implementation of the gradient boosting algorithm.

XGBoost builds on the principles of gradient boosting but incorporates several enhancements to improve performance and scalability.

The main difference is that XGBoost uses a more regularized model, which helps to prevent overfitting (more on that later).

TABULAR DATA: DEEP LEARNING IS NOT ALL YOU NEED

The paper's primary objective was to systematically compare recent deep learning architectures for tabular data, as there are no standard benchmark datasets and different models were not equally optimized in the various papers

DEEP LEARNING MODELS FOR TABULAR DATA – CHALLENGES

DL Models: Challenges with Tabular Data

- **Lack of locality:** DL models typically learn global representations of data, less effective at capturing local feature relationships.
- **Data sparsity:** Sensitive to data sparsity, not enough data to learn accurate representations of the data.
- **Mixed feature types:** Difficult to train on data with mix of numerical, ordinal, and categorical features; presents scaling issues
- **Lack of prior knowledge:** Don't take advantage of prior knowledge about data structure, like order of or relationships between features.
- **Interpretability:** Difficult to interpret, learn complex representations of the data that are not easily understood by us mere mortals.

DEEP LEARNING MODELS TESTED

TabNet

Uses an encoder and sparsemax layer which forces a smaller feature set, but not all or none with soft thresholds

Neural Oblivious Decision Ensembles (NODE)

An ensemble model comprised of oblivious decision trees (ODT's). Only one feature is chosen at each level, resulting in a balanced ODT that can be differentiated.

DNF-Net

The idea is based on disjunctive normal formulas (DNF). The complete model is an ensemble of disjunctive normal neural form (DNNF) and fully connected layers

1D-CNN

The model is based on the idea that the CNN structure performs well in feature extraction. FC layer used to create a larger feature set with locality characteristics, followed by several 1D-Conv layers with shortcut-like connections

STUDY EXPERIMENTS – DATASETS

To address the fact that there is no standard data benchmark, the authors took all the datasets from the tested models' papers (3 from each of 3 studies) along with 2 more unseen datasets to evaluate the models' performance

Dataset	Features	Classes	Samples	Source	Paper
Gesture Phase	32	5	9.8k	OpenML	DNF-Net
Gas Concentrations	129	6	13.9k	OpenML	DNF-Net
Eye Movements	26	3	10.9k	OpenML	DNF-Net
Epsilon	2000	2	500k	PASCAL Challenge 2008	NODE
YearPrediction	90	1	515k	Million Song Dataset	NODE
Microsoft (MSLR)	136	5	964k	MSLR-WEB10K	NODE
Rossmann Store Sales	10	1	1018K	Kaggle	TabNet
Forest Cover Type	54	7	580k	Kaggle	TabNet
Higgs Boson	30	2	800k	Kaggle	TabNet
Shrutime	11	2	10k	Kaggle	New dataset
Blastchar	20	2	7k	Kaggle	New dataset

STUDY EXPERIMENTS – OPTIMIZATION

The goal of the authors was to equally optimize all tested models, so they created a tuning protocol which was applied to all models:

- Used the HyperOpt library (Bayesian optimization)
- Hyperparameter search run for 1000 steps (parameters combos) on each dataset
- Each model was optimized over 6-9 main hyperparameters while the rest of the hyperparameter values were taken from their respective original papers
- **Failure point in our view: Not the same partitioning of all the datasets**

EXPERIMENTAL RESULTS

The first test was to gauge how the DL models performed (on CE/RMSE) when trained on datasets that were not included in their original papers, and comparing them to XGBoost and various ensembles

Model Name	Rossman	CoverType	Higgs	Gas	Eye	Gesture
XGBoost	490.18 ± 1.19	3.13 ± 0.09	21.62 ± 0.33	2.18 ± 0.20	**56.07**±0.65	80.64 ± 0.80
NODE	488.59 ± 1.24	4.15 ± 0.13	21.19 ± 0.69	2.17 ± 0.18	68.35 ± 0.66	92.12 ± 0.82
DNF-Net	503.83 ± 1.41	3.96 ± 0.11	23.68 ± 0.83	**1.44** ±0.09	68.38 ± 0.65	86.98 ± 0.74
TabNet	**485.12**±1.93	3.01 ± 0.08	**21.14**±0.20	1.92 ± 0.14	67.13 ± 0.69	96.42 ± 0.87
1D-CNN	493.81 ± 2.23	3.51 ± 0.13	22.33 ± 0.73	1.79 ± 0.19	67.9 ± 0.64	97.89 ± 0.82
Simple Ensemble	488.57 ± 2.14	3.19 ± 0.18	22.46 ± 0.38	2.36 ± 0.13	58.72 ± 0.67	89.45 ± 0.89
Deep Ensemble w/o XGBoost	489.94 ± 2.09	3.52 ± 0.10	22.41 ± 0.54	1.98 ± 0.13	69.28 ± 0.62	93.50 ± 0.75
Deep Ensemble w XGBoost	485.33 ± 1.29	**2.99** ±0.08	22.34 ± 0.81	1.69 ± 0.10	59.43 ± 0.60	**78.93** ±0.73
	TabNet			DNF-Net		

Model Name	YearPrediction	MSLR	Epsilon	Shrutime	Blastchar
XGBoost	77.98 ± 0.11	55.43±2e-2	11.12±3e-2	13.82 ± 0.19	20.39 ± 0.21
NODE	76.39 ± 0.13	55.72±3e-2	**10.39**±1e-2	14.61 ± 0.10	21.40 ± 0.25
DNF-Net	81.21 ± 0.18	56.83±3e-2	12.23±4e-2	16.8 ± 0.09	27.91 ± 0.17
TabNet	83.19 ± 0.19	56.04±1e-2	11.92±3e-2	14.94±, 0.13	23.72 ± 0.19
1D-CNN	78.94 ± 0.14	55.97±4e-2	11.08±6e-2	15.31 ± 0.16	24.68 ± 0.22
Simple Ensemble	78.01 ± 0.17	55.46±4e-2	11.07±4e-2	13.61±, 0.14	21.18 ± 0.17
Deep Ensemble w/o XGBoost	78.99 ± 0.11	55.59±3e-2	10.95±1e-2	14.69 ± 0.11	24.25 ± 0.22
Deep Ensemble w XGBoost	**76.19** ±0.21	**55.38**±1e-2	11.18±1e-2	**13.10**±0.15	**20.18**±0.16
	NODE			New datasets	

EXPERIMENTAL RESULTS – COMMENTARY

Key Findings:

- No deep learning model consistently outperformed the others.
- The XGBoost model generally outperformed the deep models. For 8 of the 11 datasets, XGBoost outperformed the deep models that did not appear in the relevant original paper.
- The ensemble of deep models and XGBoost outperformed the other models in most cases. For 7 of the 11 datasets, the ensemble of deep models or XGBoost was significantly better than the single deep models.

RESULTS – RELATIVE PERFORMANCE

For each dataset the relative performance of each model was calculated and compared to the best model for that dataset. The table below presents the averaged relative performance per model on all its unseen datasets, with the ensemble of all models as the standout.

Name	Average Relative Performance (%)
XGBoost	3.34
NODE	14.21
DNF-Net	11.96
TabNet	10.51
1D-CNN	7.56
Simple Ensemble	3.15
Deep Ensemble w/o XGBoost	6.91
Deep Ensemble w XGBoost	**2.32**

OUR CRITIQUES OF THE PAPER

- **Not comprehensive enough:** he paper's conclusions are based on a limited selection of tabular datasets, DL models, and tasks, and could've explored ensembles more.
- **Secondary endpoints lacking in specifics:** The authors stated that DL models are "challenging to optimize" and that they explored the tradeoffs of compute vs. accuracy, but details were "sparse".
- **No code repo:** Industry critics suggest that the study's findings must be viewed with skepticism as a result.
- **Food for thought – importance of performance on unseen data?** Why must models be generalizable to different datasets? Why can't they be optimized for the task at hand and generalizable to unseen data of that specifc type and domain?

IMPLICATIONS FOR THE FIELD OF MACHINE LEARNING

Suggests that DL models may not be the best choice for all tabular data tasks, and that simple statistical models and traditional ML models like XGBoost should not be overlooked.

Clearly more research is needed to understand the limitations of DL models for tabular data, to improve DL model tuning, and to further study which types of tabular data are most appropriate for different models and architectures.

The paper also highlights the potential of ensemble methods, combining traditional and DL models, to improve performance on tabular datasets.

SUMMARY

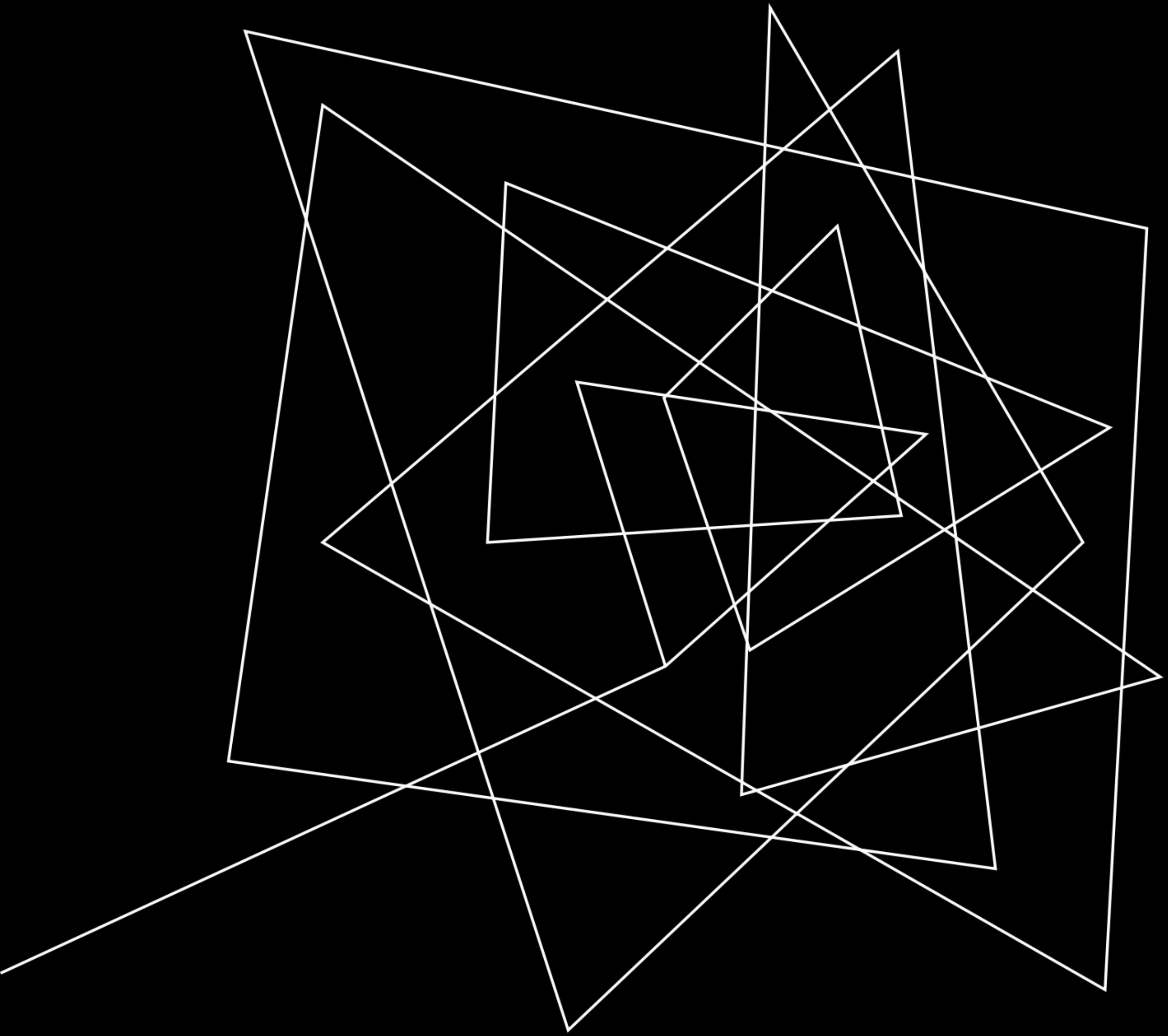
- Deep learning models can come up short vs XGBoost on unseen data
- When possible tradeoffs between performance, computational inference cost, and hyperparameter optimization time are explored, it shows that we must take the reported deep models' performance on accuracy with a grain of salt

REFERENCES

- History of DL for tabular data – with chronological links – https://sebastianraschka.com/blog/2022/deep-learning-for-tabular-data.html
- TabNet – TabNet: Attentive Interpretable Tabular Learning https://arxiv.org/pdf/1908.07442.pdf
- LANGUAGE MODELS ARE REALISTIC TABULAR DATA GENERATORS Oct2022 updated 2023 – https://arxiv.org/pdf/2210.06280.pdf
- Well-tuned Simple Nets Excel on Tabular Datasets https://arxiv.org/pdf/2106.11189.pdf
- TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second – https://arxiv.org/pdf/2207.01848.pdf
- SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training – https://arxiv.org/pdf/2106.01342.pdf
- TabDDPM: Modelling Tabular Data with Diffusion Models 2022-09 – https://arxiv.org/pdf/2209.15421.pdf
- Revisiting Pretraining Objectives for Tabular Deep Learning 2022-07 https://arxiv.org/pdf/2207.03208.pdf
- Transfer Learning with Deep Tabular Models (2022-06) https://arxiv.org/pdf/2206.15306.pdf
- Hopular: Modern Hopfield Networks for Tabular Data (2022-06) (a form of RNNs) – https://arxiv.org/pdf/2206.00664.pdf
- On Embeddings for Numerical Features in Tabular Deep Learning (2022-03) https://arxiv.org/pdf/2203.05556.pdf
- DANets: Deep Abstract Networks for Tabular Data Classification and Regression (2022-09) https://arxiv.org/pdf/2112.02962.pdf

REFERENCES (CONT'D)

- Deep Neural Networks and Tabular Data: A Survey (2021-10) – https://arxiv.org/pdf/2110.01889.pdf
- Revisiting Deep Learning Models for Tabular Data (2021-06) https://arxiv.org/pdf/2106.11959.pdf
- XBNet: An Extremely Boosted Neural Network (2021-06) https://arxiv.org/pdf/2106.05239.pdf
- Regularization Learning Networks: Deep Learning for Tabular Datasets https://arxiv.org/pdf/1805.06440.pdf
- XGBoost – XGBoost: Scalable GPU Accelerated Learning – https://arxiv.org/pdf/1806.11248.pdf
- Yandex 2021 Revisiting Topic – https://proceedings.neurips.cc/paper/2021/file/9d86d83f925f2149e9edb0ac3b49229c-Paper.pdf
- Deep Neural Networks and Tabular Data: A Survey – https://arxiv.org/pdf/2110.01889.pdf
- 2023 Review with links – https://www.strong.io/blog/deep-learning-for-tabular-data-an-overview
- July 2021 Review – https://m-clark.github.io/posts/2021-07-15-dl-for-tabular/
- May 2022 Update – https://m-clark.github.io/posts/2022-04-01-more-dl-for-tabular/
- Meta-analysis from late-2022 / early-2023 with tons of links – http://www.csam.or.kr/journal/view.html?uid=2036&&vmd=Full
- Detailing methods with reference links – https://wandb.ai/sauravm/RTDL/reports/Revisiting-Deep-Learning-Models-for-Tabular-Data--VmlldzoxNDE1Njk0
- Why do tree-based models still outperform deep learning on tabular data? 2022-07 https://arxiv.org/pdf/2207.08815.pdf
- Detailing taxonomy & 27 Methods – https://paperswithcode.com/methods/category/deep-tabular-learning



SUMMER BREAK
IS ALL YOU
REALLY NEED!