Towards Quantifying the Effect of Datasets for Benchmarking: A Look at Tabular Machine Learning

Ravin Kohli* RKOHLI@CS.UNI-FREIBURG.DE
Albert-Ludwigs-Universität Freiburg, Germany

Matthias Feurer* MATTHIAS.FEURER@STAT.UNI-MUENCHEN.DE
Ludwig-Maximilians-Universität München, Germany
Munich Center for Machine Learning, Germany

Katharina Eggensperger KATHARINA.EGGENSPERGER@UNI-TUEBINGEN.DE
University of Tübingen, Germany

Bernd Bischl BERND.BISCHL@STAT.UNI-MUENCHEN.DE
Ludwig-Maximilians-Universität München, Germany
Munich Center for Machine Learning, Germany

Frank Hutter FH@CS.UNI-FREIBURG.DE
Albert-Ludwigs-Universität Freiburg, Germany

Reviewed on OpenReview: *https://openreview.net/forum?id=ACLLU9nQ2E*

Abstract

Data in tabular form makes up a large part of real-world ML applications, and thus, there has been a strong interest in developing novel deep learning (DL) architectures for supervised learning on tabular data in recent years. As a result, there is a debate as to whether DL methods are superior to the ubiquitous ensembles of boosted decision trees. Typically, the advantage of one model class over the other is claimed based on an empirical evaluation, where different variations of both model classes are compared on a set of benchmark datasets that supposedly resemble relevant real-world tabular data. While the landscape of state-of-the-art models for tabular data changed, one factor has remained largely constant over the years: *The datasets*. Here, we examine 30 recent publications and 187 different datasets they use, in terms of age, study size and relevance. We found that the average study used less than 10 datasets and that half of the datasets are older than 20 years. Our insights raise questions about the conclusions drawn from previous studies and urge the research community to develop and publish additional recent, challenging and relevant datasets and ML tasks for supervised learning on tabular data.

1 Introduction

Empirical evaluations are crucial to studying algorithms under varying conditions and measuring progress (McGeoch, 2012; Johnson, 2002) and, thus, are a fundamental part of rigorous, data-driven ML research (Sculley et al., 2018). Here, we focus on supervised learning on tabular data, which is an omnipresent data type in many domains, such as in medicine, finance and industry (Chui et al., 2018; Müller, 2023). The availability of a large number of representative, realistic datasets is a necessary requirement for running conclusive

*. Equal Contribution.

©2024 Kohli, Feurer, Eggensperger, Bischl and Hutter.

experiments to assess the performance and usefulness of specific ML algorithms in a given domain, and to drive forward the development of new state-of-the-art models. Arguably, the more mature ML becomes as a discipline and the more common its application becomes in a certain domain, the larger and more detailed our benchmarks should be, when we move from proofs-of-concepts to a more thorough type of analysis. However, currently, there is a risk of low comparability and contradicting results due to studies using different benchmark datasets and a risk of misleading conclusions if the selected ML tasks are not representative.

Tree-based ensembles have been the dominating model class for tabular supervised learning due to their flexibility and scalability, especially variants of gradient-boosted decision trees (GBDT; Friedman, 2001; Chen and Guestrin, 2016; Ke et al., 2017; Prokhorenkova et al., 2018) and random forests (Breiman, 2001; Wright and Ziegler, 2017). Driven by the triumph of DL in many other domains, DL researchers started to also focus on tabular data. Consequently, which of the two model classes performs better is a heavily and often passionately debated question, with publications favoring one or the other based on *empirical evaluations* (Gorishniy et al., 2021; Kadra et al., 2021; Borisov et al., 2022; Shwartz-Ziv and Armon, 2022; Grinsztajn et al., 2022; Hollmann et al., 2023; McElfresh et al., 2023). Since the landscape of relevant models has changed in the past few years, our primary aim is to scrutinize a fundamental design decision of every empirical study: *The choice of datasets*. Despite recent advances in providing large collections of datasets for empirical comparisons (Bischl et al., 2021; McElfresh et al., 2023), the choice of benchmark datasets often appears to be either an afterthought or a nuisance for model-centric research.

After reviewing the state-of-the-art in tabular supervised learning in Section 2, we present our main three contributions:

1. We review 30 recently published papers studying DL and GBDT models for tabular data and collect the superset of 187 datasets used in these works, now available on `Openml.org` (Section 3).

2. We evaluate a set of 11 popular algorithms (3 DL models, 4 GBDT models and 4 baseline models) on this superset (Section 3).

3. We discuss the reviewed studies and our experimental results with respect to study size (Section 4), dataset age (Section 5) and dataset relevance (Section 6).

Afterwards, we propose several ways forward to improve the experimental design in tabular classification in Section 7 before concluding the paper in Section 8.

2 Background and Related Work

Now, we briefly summarize recent methods for supervised learning on tabular data and prior work evaluating the current state of empirical research.

For over a decade, tree-based ensembles, and especially GBDTs, have been considered to be state-of-the-art models for tabular data (Fernández-Delgado et al., 2014; Wainberg et al., 2016). In the last few years, DL researchers have started to target tabular data as a new domain and suggested new architectures and methods specifically tailored toward tabular data. Popular methods are often based on transformers (Arik and Pfister, 2020; Chen et al., 2022; Kossen et al., 2021; Somepalli et al., 2022; Gorishniy et al., 2022; Hollmann et al.,

2023), GAMs (Agarwal et al., 2021; Dubey et al., 2022; Radenovic et al., 2022), standard feed-forward and ResNet architectures (Gorishniy et al., 2021; Kadra et al., 2021) or a hybrid of a tree-style model with a network (Popov et al., 2020; Sarkar, 2022; Yoon et al., 2020; Joseph and Raj, 2022). Furthermore, several researchers have critically examined these proposed advances and have conducted neutral and independent studies, i.e., not comparing against a model that was developed by the authors of the study themselves (Boulesteix et al., 2013), often finding that many neural networks do not work as well as proclaimed by the publications introducing them (Borisov et al., 2022; Grinsztajn et al., 2022; Gorishniy et al., 2021; Shwartz-Ziv and Armon, 2022; McElfresh et al., 2023) but also showing positive results for some (McElfresh et al., 2023).

Shwartz-Ziv and Armon (2022) go one step further and also demonstrate that an ensemble of novel DL techniques with GBDT models can indeed give better results, similar to the known fact that standard feed-forward networks can be combined with tree-based methods to achieve better performance (Zimmer et al., 2021). More recently, McElfresh et al. (2023) conducted a large-scale study aiming to identify dataset characteristics that make a dataset more favorable for tree-based or NN-based models, finding that on a large number of datasets, both approaches perform en-par and that tree-based methods can handle irregular features (e.g., those with a skewed range or standard deviation of features) better than some NN-based models.

By the very nature of established review procedures and the biases they cause, all works introducing a novel method provide empirical results that demonstrate the superior performance of that method; however, several works have criticized specifically the field of ML for testing novel ideas only on a small set of datasets that might not be representative of the real world (Saitta and Neri, 1998; Wagstaff, 2012; Raji et al., 2021) or an experimental setup that fails to identify the source of empirical gains (Lipton and Steinhardt, 2019). While it can be an important "sanity check" to ensure that an algorithm is implemented correctly and working as intended, picking a small number will usually not cover the space of realistic datasets well and can therefore lead to spurious and potentially misleading or unreliable results, which might not hold under qualitative replication (Macià et al., 2013; Macià and Bernadó-Mansilla, 2014; Muñoz et al., 2018). Nevertheless, small numbers of datasets are used in many empirical studies; Macià et al. (2013) demonstrated this for papers from 15 years ago[1] and we demonstrate in this paper that this still holds for more recent work.

3 Subject of this Study: Commonly used Datasets for Prototypical Studies in Tabular ML

We base our selection of datasets on the well-known blog post by Sebastian Raschka[2] that surveys 30 papers which evaluate or propose DL for tabular data. First, we briefly discuss the algorithms and datasets we considered, starting with a summary of the publications, our procedure to collect datasets and summary statistics.

1. Macià et al. (2013) studied 215 papers from the journal *Pattern Recognition* and the *International Conference on Machine Learning* that have been published between 2008 and 2010, and which compared at least two classifiers and contained the word classifier either in the title or abstract, finding that more than 75% of the works use 10 or less datasets.
2. `https://sebastianraschka.com/blog/2022/deep-learning-for-tabular-data.html`, accessed on Feb 4th, 2024 [last update on Jan 23rd, 2023]

We chose to consider the papers from this blog post as a concise collection of recent work that avoids injecting our own biases; we list all[3] of the papers it considers in Table 3 in Appendix A. In addition, we consider two OpenML benchmarking suites (Bischl et al., 2021) as two popular dataset collections used to evaluate and compare ML models: the *OpenML-CC18* (Bischl et al., 2021) and the *AMLB* (Gijsbers et al., 2022) suite. This gives us a total of 31 entries in the table, out of which we consider 26 in this work.[4] Overall, we found 211 classification and 54 regression datasets. We restricted ourselves to supervised classification to have a manageable experimental setup. After screening the classification datasets, we are left with 187 that we consider in our work, and we describe the datasets we had to leave out in Appendix A.3. [5]

To facilitate a comprehensive and informative comparison, we evaluated a selection of 11 commonly used ML models: *Logistic Regression (Linear)*, *Random Forests (RF)* and two simple *multi-layer Perceptrons (MLP sklearn/MLP Pytorch)* as baseline algorithms, 4 *tree-based boosting algorithms (CatBoost, XGB, LGBM, HGBM)* and 3 recent *advanced DL architectures (ResNet, SAINT and FT-Transformer)*. For each method, we used random search (Bergstra and Bengio, 2012) for 100 hyperparameter configurations using a train-valid-test split using ROC AUC and used the test performance of the configuration with the best validation performance. We provide the details of these methods, search spaces and the experimental protocol in Appendix C.

We note that the main goal of our work is not to provide an overview of, or compare, methods but to focus on the *most commonly used datasets* in *prototypical studies for supervised tabular ML* and how the changing model landscape impacts the results. For this reason, and in order to conduct a neutral comparison (Boulesteix et al., 2013), we also leave out two methods that some of us co-authored: TabPFN (Hollmann et al., 2023) and regularization cocktails (Kadra et al., 2021).

4 Part I: A Look at Study Sizes

We start our assessment by looking at the size of the studies conducted in these papers. For this assessment, we consider all datasets, including classification, regression and other tasks, as shown in Table 3 in Appendix A. Classification seems to be by far the most prominent task type; papers roughly use 3.5 times as many classification datasets as regression datasets in their work. This is also underlined by the fact that we found a total of 187 classification datasets compared to 54 regression datasets in these papers.

3. We do not list the entry *Denoising Autoencoders (DAEs) for Tabular Data* because it refers to Kaggle Notebooks and not an experimental study.
4. Out of the 31 papers, we do not consider two papers introducing data generators (Borisov et al., 2023; Kotelnikov et al., 2023) and three further studies using datasets and tasks that fall beyond the scope of this paper: one paper only uses regression datasets (Zhu et al., 2021), one paper uses pre-training models on a multi-label dataset (Levin et al., 2023), one paper only contained preprocessed datasets where the pre-processing is not clearly described (Cai et al., 2021). This leaves us with a total of 26 collections of datasets that we study in this paper.
5. Overall, we have managed to collect datasets for 14 papers, while for the remaining 12 papers, at least one dataset is missing (as we describe in the Appendix A).

#data sets	**1**	**2 − 10**	**11 − 30**	**> 30**	**Avg**
#Papers using #datasets (main exp.)	2(7.41%)	17(62.96%)	6(22.22%)	2 (7.41%)	11.74
#Papers using #datasets (all)	1(3.70%)	11(40.74%)	10(37.04%)	5(18.51%)	22.37
Macià et al. (2013)	25.5%	55.3%	16.7%	2.3%	8

Table 1: Number of datasets used in the publications considered in this work (see Table 3 in Appendix A) (n=27) in relation to the numbers provided by Macià et al. (2013). *(main exp.)* only refers to classification datasets used in the main experiments, *(all)* considers all datasets used in a paper.

Following Macià et al. (2013), we analyze the number of datasets per paper used in the main experiments (and overall) and present the results in Table 4. Our collection of publications roughly follows the same pattern as identified by Macià et al. (2013), who considered 215 general ML papers eleven years ago: Only very few papers use only a single dataset (7.41%), and most papers use between two and ten datasets for their main experiment(62.96%). When considering all datasets used in a paper (not just in the main results), we also find a large number of papers using between 11 and 30 datasets. This is also in line with the more recent results by Bouthillier and Varoquaux (2020), who surveyed more than 300 submissions to NeurIPS'19 (and ICLR'19), concluding that 44% of the NeurIPS papers (and 50% of the ICLR papers) used $3 - 5$ datasets.

To visualize the impact of the study size, we compute the rank of the performance of ML models on increasing subsets of datasets in Figure 1 (left-hand side). First, we consider randomly chosen subsets (thin lines) and compare them to the overall rank across all possible subsets (dotted lines). Additionally, some datasets are used more often than others, and we show the rank across increasing subsets of the most popular datasets (solid lines), i.e. the line shows the rank across the three (four, five, six, ...) most often used datasets. While we do not want to discuss the ranking of individual models, this visualization shows that there is a lot of variability in the subsets and that especially a small number of datasets can lead to large instability and contradictory conclusions based on the data subset sample.[6]

5 Part II: A Look at Dataset Age

Next, we study the age of the used datasets and whether and how it impacts the results. First of all, we observe that while the number of novel methods increases quickly, the number of new datasets does not. We show the empirical distribution of the dataset age in Figure 2 (left) and observe that overall the datasets are relatively old; not a single dataset is newer than from 2021. Concretely, the 75th Percentile is 2012, i.e., 75% of the datasets were created before the deep learning boom started with the release of AlexNet (Krizhevsky et al., 2012). In addition, in Figure 2 (middle and right) we show the dataset distribution over

6. At the same time, we note that a study might have good reasons to choose a subset to assess algorithms under specific conditions, e.g., small datasets or only continuous features. Such experiments are important to understand the behaviour of algorithms, and also do not yield contradictory results (it may well be that algorithm A is better than algorithm B on small datasets and vice versa on large datasets).

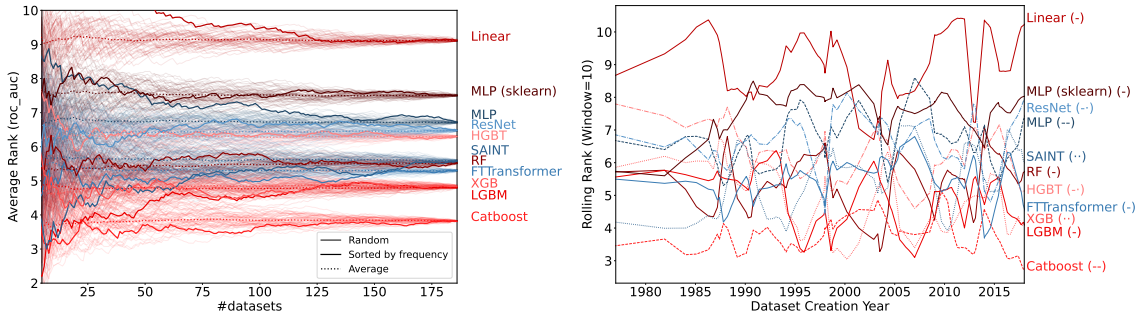

Figure 1: (Left) Ranking for increasing subsets of datasets. The thin, semi-transparent lines are computed on randomly sampled collections, dashed lines show the overall average rank over these randomly sampled collections, and the think lines correspond to the rank across subsets of popular datasets (e.g. the rank across the 3 most often used datasets). (Right) Ranking of the methods for rolling windows of 10 datasets, sorted by their age.

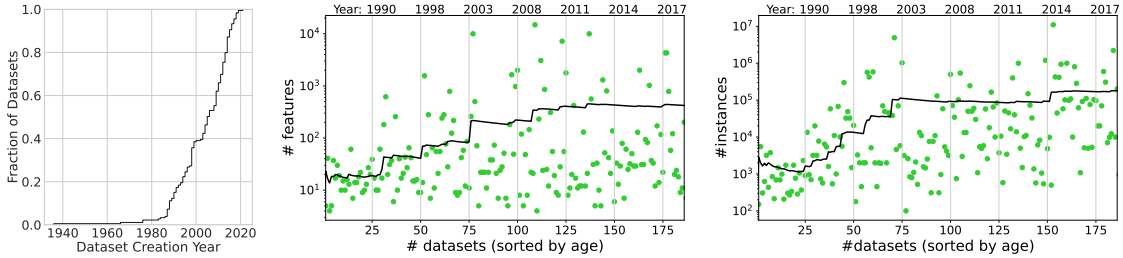

Figure 2: Cumulative distribution of dataset creation over years (left). Distribution of features (middle) and instances (right) according to the number of datasets (X-axis). The green dots depict individual datasets, and the black line is the mean number of features and instances, respectively. We also give the rough years at the top of the X-axis and apply a log transformation to the y-axis.

time. Here we can observe that datasets appear to increase in size over time, which would justify the development of new methods.

While the age of a dataset on its own does not indicate its relevance for nowadays tasks, it can still yield interesting insights, as we show next. For this, we sort the datasets by age and compute the ranking of the algorithms for rolling windows of datasets, see Figure 1 (right). We find that except for CatBoost, the ranks fluctuate quite a bit, and also that CatBoost is not always the best method. Indeed, around 2014, one could consider FT-Transformer to be the best model, and between 1998 and 2012 there is also a lot of fluctuation with four different tree-based ensembles ranking first at different points in time.

OpenML ID	Count	Name	Year
1596	12	Covertype	1998
1590	10	Adult	1996
45575	6	Epsilon	2008
45570	6	Higgs	2014
4538	5	Gesture Phase Segmentation	2014
45062	5	Churn Modelling (shrutime)	2019
23512	5	Higgs Small	2014
31	5	German Credit	1994
14674	4	Blood Transfusion Service Center	2008
42397	4	Credit Fraud	2015
1494	4	QSAR Biodeg	2013
37	4	Diabetes	1988
45556	4	Click	2012
45554	4	Fico	2018
40975	4	Car	1988

Table 2: Most frequently used datasets that have been used in at least four different suites. For each dataset, we list its OpenML dataset ID (**ID**), by how many papers it was used (**Count**), and its creation year (**Year**).

6 Part III: A Look at Individual Datasets

Finally, we are also interested in the datasets themselves and what kind of tasks they represent. In the following, we share some of the peculiarities we found while studying the datasets; for a full list, see Appendix A.5.

First, some datasets are used more often than others. We list datasets that were used in four or more papers in Table 6 and highlight several interesting findings: (a) The two most often used datasets, *Covertype* (Blackard and Dean, 1999) and *Adult* (Kohavi, 1996), are more than 20 years old, (b) out of the datasets that were used at least four times, 5/15 are older than 20 years, 11/15 are 10 years or older, and only 4 are younger than ten years, (b) the third most often used dataset, *Epsilon* (Sonnenburg and Franc, 2008; Boullé, 2009), is an artificial dataset from the Pascal Large Scale Learning Challenge 2008, and (c) the Higgs dataset appears twice in this ranking: once in its original size with 11M samples and once in a small version (whereas both are often referred to as *Higgs*).[7] Second, among the total number of 187, we find several datasets that are commonly referred to as "image" datasets: *MNIST*, *Fashion-MNIST*, *CIFAR-10*, *Devnagari script*, *mfeat-pixel* and *optdigits*; but also several datasets that constitute feature extractors from images, such as *mfeat-fourier* and variants of that dataset constructed using other feature extraction techniques. Moreover, among the most-used datasets, we also find one dataset that contains features extracted from a video (Gesture Phase Segmentation). We argue that image and video data (as well as their extracted features) should no longer be used as tabular data because using their raw representation inside a deep neural network has led to drastically improved performance

7. We even found a third dataset named *Higgs* which originates from the popular Higgs Boson Kaggle competition (Adam-Bourdarios et al., 2015).

over tabular classification methods. Third, reproducibility is a main concern for empirical experiments. However, we found two regularly used datasets (*Cardiovascular Disease* and *Tours and Travels Customer Churn Prediction*) where we could not find the origin, including whether they are real-world or synthetic datasets. Additionally, for example, *FICO/HELOC* exists in several variations and has been used under the same name. Finally, we also found several datasets (*adult*, *German Credit*, *Diabetes*, *COMPAS* and *iris*) that are *retired* or *retracted* and, thus, should not be used anymore (Ding et al., 2021; Bao et al., 2021; Poisot, 2020; Radin, 2017; Grömping, 2019). These issues question the relevance of existing empirical evaluations for current, real-world tabular learning scenarios and urge the usage of unique dataset identifiers, such as OpenML dataset IDs.

7 Discussion

First of all, while we do not aim to compare the different algorithms in terms of performance, we find CatBoost (Prokhorenkova et al., 2018) to obtain a strong average ranking, which is in line with the results of other comparisons (McElfresh et al., 2023; Zhu et al., 2023).

Based on the three angles from which we looked at the datasets used in empirical comparisons of supervised classification models for tabular data (study size, dataset age and individual datasets), we conclude the following:

1. We need to separate the representation of datasets from the information a dataset represents. A dataset should not be considered tabular because it is stored in tabular form but because of its content (e.g., the pixels of an image can, in principle, be stored in a table, but we would not consider the resulting dataset to be tabular).

2. For several areas, extracting features from raw data and using tabular classification methods on the outcome yields competitive results (e.g., EEG classification (Gemein et al., 2020) or particle detection in a physics experiment (Adam-Bourdarios et al., 2015)). However, for other domains (e.g., image data) this representation is no longer relevant for state-of-the-art research. Therefore, as a community, we must come up with a clear definition of when data should be used in its tabular representation, contrary to other forms.

3. We need to be able to define what makes problems relevant. While we can define the space of problems to measure if we can solve datasets in every part of the space (Muñoz et al., 2018), we should also focus on problems considered relevant by a large community and prospective users of tabular ML methods.

4. Finally, and most importantly, there has been a non-negligible shift in the model landscape, and we must change our benchmark dataset landscape accordingly to no longer measure progress in the past.

Based on these, we will be able to come up with a crisp definition of what constitutes a tabular classification task and we make a concrete proposal in Appendix B, based on which we can define new standard benchmark datasets. In order to allow the ML community to make progress, we call for the creation of such benchmark suites.

As a first concrete step, we propose to deprecate datasets for which we are very certain that they should not be used in their tabular representation: images, videos, and datasets

extracted from those modalities. We are not aware of any recent results that demonstrate that any model class besides deep learning models tailored towards those modalities achieves competitive performance.

8 Conclusion and Future Work

In this work, we surveyed the state of datasets used in recent papers on novel architectures for tabular deep learning and works that conduct independent empirical comparisons of supervised classification models. We found that the selection of datasets can lead to different rankings of models and, thereby, also different conclusions. We found that datasets of different ages can lead to different rankings. Last but not least, we found that most datasets used are rather old. When introducing new solutions, we would have expected the works to also use rather recent datasets. As a concrete first step forward, we suggested dropping image, video and datasets extracted from these modalities. We hope that our work starts a discussion about the datasets used to benchmark algorithms for tabular data, which hopefully leads to better practices in the comparison of machine learning algorithms for tabular data, and to more people making current tabular datasets available.

In the future, it would be great to take a closer look at how the different data types and the dataset age impact the performance of the models under consideration. Finally, the community needs to discuss how to incentivize creating and providing datasets on platforms such as OpenML in order to encourage the sharing of relevant new challenges.

Broader Impact Statement

We conducted a meta-study reusing the same datasets used in previous papers to maximize the alignment of our results with the original results. This includes several ethically questionable datasets, such as *COMPAS* (Bao et al., 2021) and *Iris* (Poisot, 2020) and retired datasets, such as *adult* (Ding et al., 2021). Using these datasets might send the signal that they can be used; however, we do not endorse the usage of these datasets and urge researchers to exclude them from future work (as discussed in Section 6).

Reproducibility Statement

We provide results, the analysis scripts and our code for the experiments at `https://github.com/automl/dmlr-iclr24-datasets-for-benchmarking`. Furthermore, we refer the reader to Appendix C for a discussion of our setup, and Appendix A for a discussion of the datasets we used.

Acknowledgments

The authors thank Jan N. van Rijn and Pieter Gijsbers for their help in uploading datasets to OpenML and Sebastian Fischer for his help in an early review of the blog post of Sebastian Raschka. Katharina Eggensperger was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC number 2064/1 – Project number 390727645. This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant number 417962828. We

acknowledge funding by the European Union (via TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215, and ERC Consolidator Grant DeepLearning 2.0, grant no. 101045765). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

Funded by the European Union

References

C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kégl, and D. Rousseau. The Higgs boson machine learning challenge. In G. Cowan, C. Germain, I. Guyon, B. Kégl, and D. Rousseau, editors, *Proceedings of the NIPS 2014 Workshop on High-energy Physics and Machine Learning*, volume 42 of *Proceedings of Machine Learning Research*, pages 19–55, 2015.

R. Agarwal, L. Melnick, N. Frosst, X. Zhang, B. Lengerich, R. Caruana, and G. Hinton. Neural additive models: Interpretable machine learning with neural nets. In M. Ranzato, A. Beygelzimer, K. Nguyen, P. Liang, J. Vaughan, and Y. Dauphin, editors, *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*, pages 4699–4711. Curran Associates, 2021.

S. Arik and T. Pfister. TabNet: Attentive interpretable tabular learning. In F. Rossi, V. Conitzer, and F. Sha, editors, *Proceedings of the Thirty-Fourth Conference on Artificial Intelligence (AAAI'20)*. Association for the Advancement of Artificial Intelligence, AAAI Press, 2020.

D. Bahri, H. Jiang, Y. Tay, and D. Metzler. Scarf: Self-supervised contrastive learning using random feature corruption. In *Proceedings of the International Conference on Learning Representations (ICLR'22)*, 2022. Published online: `iclr.cc`.

M. Bao, A. Zhou, S. Zottola, B. Brubach, B. Brubach, S. Desmarais, A. Horowitz, K. Lum, and S. Venkatasubramanian. It's compaslicated: The messy relationship between RAI datasets and algorithmic fairness benchmarks. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. Curran Associates, 2021.

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

B. Bischl, G. Casalicchio, M. Feurer, F. Hutter, M. Lang, R. Mantovani, J. van Rijn, and J. Vanschoren. OpenML benchmarking suites. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. Curran Associates, 2021.

J. Blackard and D. Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, 24(3):131–151, 1999.

V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2022.

V. Borisov, K. Sessler, T. Leemann, M. Pawelczyk, and G. Kasneci. Language models are realistic tabular data generators. In *International Conference on Learning Representations (ICLR'23)*, 2023. Published online: `iclr.cc`.

A.-L. Boulesteix, S. Lauer, and M. Eugster. A plea for neutral comparison studies in computational sciences. *PLOS One*, 8(4), 2013.

M. Boullé. A parameter-free classification method for large scale learning. *Journal of Machine Learning Research*, 10(46):1367–1385, 2009.

X. Bouthillier and G. Varoquaux. Survey of machine-learning experimental methods at NeurIPS2019 and ICLR2020. Research report [hal-02447823], Inria Saclay Ile de France, 2020.

X. Bouthillier, P. Delaunay, M. Bronzi, A. Trofimov, B. Nichyporuk, J. Szeto, N. Mohammadi Sepahvand, E. Raff, K. Madan, V. Voleti, S. Ebrahimi Kahou, V. Michalski, T. Arbel, C. Pal, G. Varoquaux, and P. Vincent. Accounting for variance in machine learning benchmarks. In A. Smola, A. Dimakis, and I. Stoica, editors, *Proceedings of Machine Learning and Systems 3*, volume 3, pages 747–769, 2021.

L. Breiman. Random forests. *Machine Learning Journal*, 45:5–32, 2001.

L. Buturović and D. Miljković. A novel method for classification of tabular data using convolutional neural networks. *bioRxiv*, 2020.

S. Cai, K. Zheng, G. Chen, H. Jagadish, B. Ooi, and M. Zhang. Arm-net: Adaptive relation modeling network for structured data. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, page 207–220. Association for Computing Machinery, 2021.

M. Carlisle. racist data destruction?, 2019. URL `https://medium.com/@docintangible/racist-data-destruction-113e3eff54a8`.

J. Chen, K. Liao, Y. Wan, D. Chen, and J. Wu. Danets: Deep abstract networks for tabular data classification and regression. In K. Sycara, V. Honavar, and M. Spaan, editors, *Proceedings of the Thirty-Sixth Conference on Artificial Intelligence (AAAI'22)*, pages 3930–3938. Association for the Advancement of Artificial Intelligence, AAAI Press, 2022.

T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In B. Krishnapuram, M. Shah, A. Smola, C. Aggarwal, D. Shen, and R. Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*, pages 785–794. ACM Press, 2016.

M. Chui, J. Manyika, M. Miremadi, N. Henke, R. Chung, P. Nel, and S. Malhotra. Notes from the AI frontier: insights from hundreds of use cases, 2018.

F. Ding, M. Hardt, J. Miller, and L. Schmidt. Retiring adult: New datasets for fair machine learning. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, pages 6478–6490. Curran Associates, 2021.

A. Dubey, F. Radenovic, and D. Mahajan. Scalable interpretability via polynomials. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Proceedings of the 36th International Conference on Advances in Neural Information Processing Systems (NeurIPS'22)*, pages 36748–36761. Curran Associates, 2022.

M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15(90):3133–3181, 2014.

J. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

L. Gemein, R. Schirrmeister, P. Chrabaszcz, D. Wilson, J. Boedecker, A. Schulze-Bonhage, F. Hutter, and T. Ball. Machine-learning-based diagnostics of eeg pathology. *NeuroImage*, 220:117021, 2020.

P. Gijsbers, M. Bueno, S. Coors, E. LeDell, S. Poirier, J. Thomas, B. Bischl, and J. Vanschoren. Amlb: an automl benchmark. *arXiv:2207.12560*, 2022.

Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko. Revisiting deep learning models for tabular data. In M. Ranzato, A. Beygelzimer, K. Nguyen, P. Liang, J. Vaughan, and Y. Dauphin, editors, *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates, 2021.

Y. Gorishniy, I. Rubachev, and A. Babenko. On embeddings for numerical features in tabular deep learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Proceedings of the 36th International Conference on Advances in Neural Information Processing Systems (NeurIPS'22)*, pages 24991–25004. Curran Associates, 2022.

L. Grinsztajn, E. Oyallon, and G. Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, pages 507–520, 2022.

U. Grömping. South german credit data: Correcting a widely used data set. Technical report, Beuth Hochschule Für Technik Berlin, 2019. URL `http://www1.beuth-hochschule.de/FB_II/reports/Report-2019-004.pdf`.

I. Guyon, L. Sun-Hosoya, M. Boullé, H. Escalante, S. Escalera, Z. Liu, D. Jajetic, B. Ray, M. Saeed, M. Sebag, A. Statnikov, W. Tu, and E. Viegas. Analysis of the AutoML Challenge Series 2015-2018. In F. Hutter, L. Kotthoff, and J. Vanschoren, editors,

Automated Machine Learning: Methods, Systems, Challenges, chapter 10, pages 177–219. Springer, 2019. Available for free at `http://automl.org/book`.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR'16)*, pages 770–778. Computer Vision Foundation and IEEE Computer Society, IEEE, 2016.

N. Hollmann, S. Müller, K. Eggensperger, and F. Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations (ICLR'23)*, 2023. Published online: `iclr.cc`.

X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arxiv:2012.06678 [cs.LG]*, 2020.

D. Johnson. A theoretician's guide to the experimental analysis of algorithms. *Proceedings of the 5th and 6th DIMACS implementation challenges*, 59:215–250, 2002.

M. Joseph and H. Raj. Gate: Gated additive tree ensemble for tabular classification and regression. *arXiv:2207.08548v4 [cs.LG]*, 2022.

A. Kadra, M. Lindauer, F. Hutter, and J. Grabocka. Well-tuned simple nets excel on tabular datasets. In M. Ranzato, A. Beygelzimer, K. Nguyen, P. Liang, J. Vaughan, and Y. Dauphin, editors, *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates, 2021.

G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Proceedings of the 31st International Conference on Advances in Neural Information Processing Systems (NeurIPS'17)*. Curran Associates, 2017.

R. Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 202–207. AAAI Press, 1996.

J. Kossen, N. Band, C. Lyle, A. Gomez, T. Rainforth, and Y. Gal. Self-attention between datapoints: Going beyond individual input-output pairs in deep learning. In M. Ranzato, A. Beygelzimer, K. Nguyen, P. Liang, J. Vaughan, and Y. Dauphin, editors, *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*, pages 28742–28756. Curran Associates, 2021.

A. Kotelnikov, D. Baranchuk, I. Rubachev, and A. Babenko. Tabddpm: Modelling tabular data with diffusion models. *OpenReview*, 2023. URL `https://openreview.net/forum?id=EJka_dVXEcr`.

A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Proceedings of the 26th International Conference on Advances in Neural Information Processing Systems (NeurIPS'12)*, pages 1097–1105. Curran Associates, 2012.

R. Levin, V. Cherepanova, A. Schwarzschild, A. Bansal, C. Bruss, T. Goldstein, A. Wilson, and M. Goldblum. Transfer learning with deep tabular models. In *International Conference on Learning Representations (ICLR'23)*, 2023. Published online: `iclr.cc`.

Z. Lipton and J. Steinhardt. Troubling trends in machine learning scholarship: Some ML papers suffer from flaws that could mislead the public and stymie future research. *Queue*, 17(1):45–77, 2019. ISSN 1542-7730. doi: 10.1145/3317287.3328534.

N. Macià and E. Bernadó-Mansilla. Towards uci+: A mindful repository design. *Information Sciences*, 261:237–262, 2014.

N. Macià, E. Bernadó-Mansilla, A. Orriols-Puig, and T. Kam Ho. Learner excellence biased by data set selection: A case for data characterisation and artificial data sets. *Pattern Recognition*, 46(3):1054–1066, 2013.

D. McElfresh, S. Khandagale, J. Valverde, V. Prasad, B. Feuer, C. Hegde, G. Ramakrishnan, M. Goldblum, and C. White. When do neural nets outperform boosted trees on tabular data? In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2023.

C. McGeoch. *A Guide to Experimental Algorithmics*. Cambridge University Press, 2012.

M. Muñoz, L. Villanova, D. Baatar, and K. Smith-Miles. Instance spaces for machine learning classification. *Machine Learning*, 107(1):109–147, 2018.

A. Müller. From automl to autods, 2023. URL `https://www.youtube.com/watch?v=jp_UZoM_OjE`. Industry Day Keynote at the 2nd AutoML conference.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

T. Poisot. Retiring iris, 2020. URL `https://armchairecology.blog/iris-dataset/`.

S. Popov, S. Morozov, and A. Babenko. Neural oblivious decision ensembles for deep learning on tabular data. In *Proceedings of the International Conference on Learning Representations (ICLR'20)*, 2020. Published online: `iclr.cc`.

L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, and A. Gulin. Catboost: Unbiased boosting with categorical features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Proceedings of the 31st International Conference on Advances in Neural Information Processing Systems (NeurIPS'18)*, page 6639–6649. Curran Associates, 2018.

F. Radenovic, A. Dubey, and D. Mahajan. Neural basis models for interpretability. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Proceedings of the 36th International Conference on Advances in Neural Information Processing Systems (NeurIPS'22)*, pages 8414–8426. Curran Associates, 2022.

J. Radin. "Digital Natives": How medical and indigenous histories matter for big data. *Osiris*, 32, 2017.

I. Raji, E. Bender, A. Paullada, E. Denton, and A. Hanna. Ai and the everything in the whole wide world benchmark. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. Curran Associates, 2021.

S. Raschka. A short chronology of deep learning for tabular data, 2022. URL `https://sebastianraschka.com/blog/2022/deep-learning-for-tabular-data.html`.

I. Rubachev, A. Alekberov, Y. Gorishniy, and A. Babenko. Revisiting pretraining objectives for tabular deep learning. *OpenReview*, 2023. URL `https://openreview.net/forum?id=kjPLodRa0n`.

L. Saitta and F. Neri. Learning in the "real world". *Mach. Learn.*, 30(2–3):133–163, 1998.

T. Sarkar. Xbnet: An extremely boosted neural network. *Intelligent Systems with Applications*, 15:200097, 2022.

B. Schäfl, L. Gruber, A. Bitto-Nemling, and S. Hochreiter. Hopular: Modern hopfield networks for tabular data. *OpenReview*, 2022. URL `https://openreview.net/forum?id=3zJVXU311-Q`.

D Sculley, J. Snoek, A. Wiltschko, and A. Rahimi. Winner's curse? on pace, progress, and empirical rigor. In *International Conference on Learning Representations Workshop track*, 2018. Published online: `iclr.cc`.

R. Shwartz-Ziv and A. Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

G. Somepalli, A. Schwarzschild, M. Goldblum, C. Bruss, and T Goldstein. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *OpenReview*, 2022. URL `https://openreview.net/forum?id=nL2lDlsrZU`.

S. Sonnenburg and V. Franc. Large scale learning - challenge, 2008. URL `https://videolectures.net/icml08_sonnenburg_lsl/`.

B. Sun, L. Yang, W. Zhang, M. Lin, P. Dong, C. Young, and J. Dong. Supertml: Two-dimensional word embedding for the precognition on structured tabular data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2973–2981, 2019.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Proceedings of the 31st International Conference on Advances in Neural Information Processing Systems (NeurIPS'17)*. Curran Associates, Inc., 2017.

K. Wagstaff. Machine learning that matters. *arXiv:1206.4656 [cs.LG]*, 2012.

M. Wainberg, B. Alipanahi, and B. Frey. Are random forests truly the best classifiers? *Journal of Machine Learning Research*, 17(110):1–5, 2016.

M. Wright and A. Ziegler. ranger: A fast implementation of random forests for high dimensional data in c++ and r. *Journal of Statistical Software*, 77(1):1–17, 2017.

J. Yoon, Y. Zhang, J. Jordon, and M. van der Schaar. Vime: Extending the success of self- and semi-supervised learning to tabular domain. In H. Larochelle, M. Ranzato, R. Hadsell, M.-F. Balcan, and H. Lin, editors, *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*, pages 11033–11043. Curran Associates, 2020.

B. Zhu, X. Shi, N. Erickson, M. Li, G. Karypis, and M. Shoaran. XTab: Cross-table pretraining for tabular transformers. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 43181–43204. PMLR, 23–29 Jul 2023.

Y. Zhu, T. Brettin, F. Xia, A. Partin, M. Shukla, H. Yoo, Y. Evrard, J. Doroshow, and R. Stevens. Converting tabular data into images for deep learning with convolutional neural networks. *Scientific Reports*, 11(11325), 2021.

L. Zimmer, M. Lindauer, and F. Hutter. Auto-Pytorch: Multi-fidelity metalearning for efficient and robust AutoDL. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:3079–3090, 2021.

Appendix A. Considered Works and Datasets Used

A.1 Dataset Collection Procedure

In the following, we discuss all dataset collections we consider and list them in Table 3. The use of datasets serves different purposes in the considered papers: (a) to visually demonstrate specific behaviour or capability on an exemplary dataset, e.g., the mushroom dataset used to study interpretability in Arik and Pfister (2020) (b) to study generalization to other dataset types or algorithm components, e.g., mid-sized datasets in Schäfl et al. (2022) or convolution layers in (Kossen et al., 2021) and (c) to compare the performance of algorithms. Here we solely focus on the datasets used for (c), the so-called *main experiment* comparing algorithms w.r.t. their performance. We describe the considered and left-out datasets per paper in Appendix A.4. To ensure reproducibility and to follow prior efforts, we used OpenML for handling the datasets so that (a) all datasets and their metadata are available in a uniform and machine-readable format, (b) datasets are stored in a single place and are accessible via APIs and (c) data splits can be stored alongside the data. In addition, using OpenML overcomes the criticism of UCI as it provides access in a uniform manner and does not require laborious manual inspection (Macià and Bernadó-Mansilla, 2014). This is a great advancement over previous work, where the dataset origins were provided as OpenML task and dataset ids, links to Kaggle competitions and notebooks, links to UCI, repositories on GitHub and links to CSV files uploaded on self-hosted websites. To follow a structured

procedure, we first manually gathered all combinations of dataset names and papers. Then, we identified the actual data [8] used and followed these steps:

1. We used the OpenML dataset ID or task ID where available.
2. If these were unavailable, we tried to find the dataset on OpenML via the name, description and other information and verified a match with the number and type of attributes and instances.
3. In case of no match, we tried to track down the original source and uploaded the dataset to OpenML to make it accessible for future researchers.

We found a total of 187 classification datasets [9]. Most datasets were already available on OpenML, and we only had to upload or re-upload (to correct wrong metadata) 21 datasets to OpenML. Unfortunately, we could only obtain 187 out of 211 classification datasets: The missing datasets either were not publicly available (5 datasets), or it was unclear how a dataset was extracted from a gene database (6 datasets). We excluded six datasets for technical reasons, such as containing string or data features or having too few samples of one class (it had two samples, but our resampling procedure requires three), one dataset because we could not identify its source, and another 5 datasets because the data was already preprocessed according to a method introduced in the paper (Cai et al., 2021). Moreover, one dataset we only managed to upload after running the experiments, and it is therefore excluded from our further analysis. We give further details in Appendix A.3. Out of the 211 datasets, 19 datasets are also used in the OpenML-CC18 and the OpenML-AutoML benchmark.

A.2 Datasets that we did use

OpenML Dataset ID	Count	Name	Creation
2	1	anneal	1990
3	2	kr-vs-kp	1983
5	1	arrhythmia	1998
6	1	letter	1991
11	2	balance-scale	1976
12	2	mfeat-factors	1998
13	3	breast-cancer	1988
14	2	mfeat-fourier	1998
15	2	breast-w	1990
16	2	mfeat-karhunen	1998
18	2	mfeat-morphological	1998
22	2	mfeat-zernike	1998

8. We did not consider whether an experiment used a specific dataset split, e.g. cross-validation or holdout, to unify our study and to always have multiple test sets available (Bouthillier et al., 2021). We considered this to be more important than having access to the exact splits used in previous works.
9. We did not consider 54 regression datasets and plan to study this in the future.

23	2	cmc	1987
25	1	horse-colic	1989
28	2	digits	1995
29	3	credit-approval	1987
31	5	credit-g	1994
32	1	pendigits	1994
37	4	diabetes	1988
38	2	sick	1986
40	1	conn-bench-sonar-mines-rocks	1988
41	1	glass	1987
43	1	haberman-survival	1976
44	3	spambase	1999
46	1	splice	1991
49	1	heart-cleveland	1988
50	2	tic-tac-toe	1990
53	1	statlog-heart	1988
54	3	vehicle	1987
56	1	congressional-voting	1984
61	2	iris	1936
151	1	electricity	1999
171	1	primary-tumor	1988
182	1	satimage	1993
187	2	wine	1998
188	3	eucalyptus	1992
300	1	isolet	1991
307	1	vowel	1987
458	2	analcatdata_auth...	2003
469	2	analcatdata_dmft	2003
554	1	mnist	1998
1017	1	arrhythmia	1998
1044	1	eye movement	2005
1049	2	pc4	2004
1050	2	pc3	2004
1053	1	jm1	2004
1063	2	kc2	2004
1067	2	kc1	2004
1068	2	pc1	2004
1111	1	kddcup09-appetency	2009
1119	1	adult	1996
1219	1	click prediction	2012
1430	1	a9a	1998
1461	3	bank-marketing	2011
1462	2	banknote-authentication	2013
1464	4	blood-transfusion	2008
1468	2	cnae-9	2009

1475	1	first-order-theorem-proving	2013
1477	1	gas concentration	2012
1478	1	har	2012
1480	2	ilpd	2011
1483	1	ldpa	2010
1485	1	madelon	2003
1486	2	nomao	2008
1487	2	ozone-level-8hr	2005
1489	2	phoneme	1993
1494	4	qsar-biodeg	2013
1497	1	wall-robot-navigation	2009
1501	1	semeion	1994
1502	1	skin-segmentation	2009
1509	1	walking-activity	2012
1510	2	wdbc	1992
1523	1	vertebral-column3	2005
1524	1	vertebral-column2	2005
1567	2	pokerhand	2002
1590	10	adult roc	1996
1596	12	covertype	1998
4134	2	bioresponse	2011
4534	1	phishingwebsites	2012
4535	1	income	1995
4538	5	gesturephasesegmentationprocessed	2014
4541	1	diabetes 130us	2014
6332	3	cylinder-bands	1994
23381	2	dresses-sales	2014
23512	5	higgs small	2014
23517	2	numerai28.6	2016
40499	1	texture	1966
40664	1	car-evaluation	1988
40668	2	connect-4	1995
40670	1	dna	1991
40685	2	shuttle	1994
40687	1	solar-flare	1989
40701	1	churn	2012
40923	2	devnagari-script	2015
40966	2	miceprotein	2015
40975	4	car	1988
40978	1	internet-advertisements	1998
40979	1	mfeat-pixel	1998
40981	1	australian	1987
40982	2	steel-plates-fault	1998
40983	2	wilt	2013
40984	2	segment	1990

40994	2	climate-model-simulation-crashes	2013
40996	1	fashion-mnist	2017
41027	2	jungle_chess_2pcs_raw_endgame_complete	2014
41138	1	apsfailure	2016
41142	1	christine	2006
41143	2	jasmine	2009
41145	1	philippine	2009
41146	2	sylvine	1998
41147	1	albert	2014
41150	2	miniboone	2005
41162	1	kick	2012
41163	1	dilbert	2014
41164	2	fabert	2013
41166	3	volkert	2006
41167	1	dionis	2014
41168	3	jannis	2010
41169	3	helena	2010
42193	1	compas	2016
42396	2	aloi	2014
42397	4	credit fraud	2015
42477	1	default	2009
42733	1	click prediction small	2012
42734	1	okcupid-stem	2011
44089	1	jannis	2011
44120	1	electricity	1999
44121	1	covertype	1998
44122	1	pol	1995
44123	1	house_16h	1990
44125	1	magicaltelescope	2004
44126	1	bank-marketing	2011
44129	1	default_of-credit-card-clients	2014
44130	1	higgs small	2005
44156	1	electricity	1999
44157	1	eye movement	2005
44159	1	covertype	1998
45019	1	bioresponse	2011
45020	1	miniboone	2009
45021	1	diabetes130us	2010
45022	1	eye movement	2008
45028	1	credit	1997
45035	1	albert	2014
45036	1	default_of-credit-card-clients	2009
45038	1	road-safety	2015
45039	1	compas-two-years	2016
45062	5	churn modelling	2019

45069	1	diabetes	2008
45545	1	travel customers	2021
45547	2	cardio	1999
45548	2	otto group product classification	2015
45551	2	higgs kaggle	2014
45554	4	fico	2018
45556	1	click	2012
45557	1	mammographic	2006
45558	1	htru2	2010
45559	1	insurance_co	2000
45560	1	online_shoppers	2018
45562	1	seismicbumps	2010
45563	1	dota2games	2016
45565	1	1995_income	1996
45566	1	santander customer transaction prediction5	2019
45567	1	hcdr_main	2018
45568	3	telco-customer-churn	2018
45570	6	higgs	2014
45575	6	epsilon	2008
45579	1	microsoft	2013

A.3 Datasets that we did not use

- 20 Newsgroups, used by Dubey et al. (2022) and Radenovic et al. (2022). The 20 newsgroups dataset contains messages from 20 news groups with the task to classify which newsgroup a text belongs to. We considered uploading this dataset, but decided against doing so because we would train a representation on a training set, but in reality, we would use a 4-fold CV setup that does not align with this train/test split and would lead to spurious results.

- adult, as used by Huang et al. (2020): This is a multi-label version of the famous adult data set (Kohavi, 1996) created for the ChaLearn AutoML challenge (Guyon et al., 2019, see online Appendix). We excluded this dataset because we do not consider multi-label problems. However, the suite used in this work already contains the a different version of the adult dataset, that only consists of the training portion of the original dataset, we do not deem this to be an issue.

- Avazu, used by Cai et al. (2021): We excluded this dataset because the paper provides the dataset only in the preprocessed version generated by the method described in the paper.

- Blog, used by Yoon et al. (2020): We excluded this dataset because it requires a non-i.i.d. split.

- Buddy, used by Kotelnikov et al. (2023). This dataset contains timestamps and it is therefore unclear how to use it in a standard supervised classification setup.

Table 3: Studied papers. We classify the papers from the blog post (Raschka, 2022) into three categories: (1) Works that study a large range of methods (**Comp**), (2) works that introduce a method (**New**), and (3) works that introduce data generation techniques (**Generative**; not considered). We also list two papers introducing OpenML benchmarking suites that are not part of the blog post (**OpenML**). Column **clf** shows the number of classification datasets used in the *main experiment* and **reg** shows the number of regression datasets, respectively. We also list the number of invalid datasets that we could not use (**inv**, see Appendix A.3; not considered), and the number of datasets that were used for experiments besides the *main experiment* (**side**, see Appendix A.4, not considered).

Type	**Method**	**Reference**	**clf**	**reg**	**inv**	**side**
Comp	-	Borisov et al. (2022)	4	1	-	-
	FT-Transformer	Gorishniy et al. (2021)	7	4	-	5
	-	Grinsztajn et al. (2022)	21	35	-	-
	-	Shwartz-Ziv and Armon (2022)	9	2	-	-
New	NAM	Agarwal et al. (2021)	2	2	1	-
	TabNet	Arik and Pfister (2020)	3	2	-	8
	SCARF	Bahri et al. (2022)	69	-	-	-
	TAC	Buturović and Miljković (2020)	1	-	1	-
	ARM-Net	Cai et al. (2021)	5	-	5	-
	DANET	Chen et al. (2022)	4	3	1	-
	SPAM	Dubey et al. (2022)	9	4	3	3
	multiple	Gorishniy et al. (2022)	7	4	-	1
	TabPFN	Hollmann et al. (2023)	30	-	-	170
	TabTransformer	Huang et al. (2020)	20	-	1	-
	GATE	Joseph and Raj (2022)	3	2	-	-
	RegCocktails	Kadra et al. (2021)	40	-	-	-
	NPT	Kossen et al. (2021)	6	4	-	2
	multiple	Levin et al. (2023)	1	-	1	2
	NODE	Popov et al. (2020)	3	3	1	-
	NBM	Radenovic et al. (2022)	8	4	3	3
	multiple	Rubachev et al. (2023)	6	5	-	1
	XGBNet	Sarkar (2022)	8	-	2	-
	Hopular	Schäfl et al. (2022)	16	-	1	4
	SAINT	Somepalli et al. (2022)	20	10	-	1
	SuperTML	Sun et al. (2019)	4	-	-	-
	VIME	Yoon et al. (2020)	11	-	9	-
	IGDT	Zhu et al. (2021)	-	2	-	-
	Summary	Mean	11.7	3.2	1.1	7.4
		Median	7	2	0	0
Generative	GReaT	Borisov et al. (2023)	-	-	-	6
	-	Kotelnikov et al. (2023)	-	-	-	16
OpenML	-	Bischl et al. (2021)	-	-	-	72
	-	Gijsbers et al. (2022)	-	-	-	71

- Click, used by Chen et al. (2022); Dubey et al. (2022); Popov et al. (2020); Radenovic et al. (2022) is a random subset of the size of 1.000.000 datapoints from the KDD Cup 2012. However, it contains more than 700.000 unique user IDs, which were too many to upload to OpenML as categories of a categorical feature. In the meantime we have manageded to upload the dataset, but only after running the experiments.
- Clinical UK: We excluded the dataset because it is not publicly available.
- Clinical US We excluded the dataset because it is not publicly available.
- Criteo, used by Cai et al. (2021): We excluded this dataset because the paper provides the dataset only in the preprocessed version generated by the method described in the paper.
- Diabetes130, used by Cai et al. (2021): We excluded this dataset because the paper provides the dataset only in the preprocessed version generated by the method described in the paper.
- Digit Completion, used by Sarkar (2022). Unfortunately, the paper does not contain enough information to identify this dataset. Furthermore, the provided code contains code to load a subset of MNIST or the digit dataset, which makes it unclear which of both might have been used. Moreover, there is no such dataset on the internet, which prevents us from using this dataset.
- ecoli, used by Schäfl et al. (2022). This dataset contains two classes that only have two samples each. This number of samples is too low for our train-valid-test procedure and we therefore exclude it.
- Frappe, used by Cai et al. (2021): We excluded this dataset because the paper provides the dataset only in the preprocessed version generated by the method described in the paper.
- MCH, used by Yoon et al. (2020): We excluded the dataset because it is not publicly available.
- MetaMIMIC, used by Levin et al. (2023): We excluded the dataset because it is not publicly available, and because it is a multi-label dataset.
- MIMIC2, used by Agarwal et al. (2021); Radenovic et al. (2022); Dubey et al. (2022): We excluded the dataset because it is not publicly available.
- MONO, used by Yoon et al. (2020): We excluded the dataset because it is not publicly available.
- MovieLens, used by Cai et al. (2021): We excluded this dataset because the paper provides the dataset only in the preprocessed version generated by the method described in the paper.
- MPV, used by Yoon et al. (2020): We excluded the dataset because it is not publicly available.

- MRV, used by Yoon et al. (2020): We excluded the dataset because it is not publicly available.
- PCT, used by Yoon et al. (2020): We excluded the dataset because it is not publicly available.
- RET, used by Yoon et al. (2020): We excluded the dataset because it is not publicly available.
- Rossmann Store Sales, used by Arik and Pfister (2020) and Shwartz-Ziv and Armon (2022): We excluded the dataset due to its time series nature.
- Syn1, Syn2, Syn3, Syn4, Syn5, and Syn6 (Arik and Pfister, 2020): We excluded these datasets due to their synthetic nature.
- Synthetic datasets used by Gorishniy et al. (2021): We excluded these datasets due to their purely synthetic nature. Nonetheless, we would like to point out the great study design: these datasets that were generated by gradually blending between neural networks and decision trees to study the inductive biases of the proposed FT-Transformer, showing where FT-Transformer improves over the previously used ResNet.
- Titanic as used by Sarkar (2022): We excluded the dataset due to its string features.

A.4 Datasets per Paper

In this section we briefly categorize the datasets from each paper into main and additional (other) experiments. This will help to better understand Table 3.

- Agarwal et al. (2021)
  - Main experiment: Use four datasets
  - Other:
    - Use the MIMIC-II datasets with a doctor to validate intelligibility of the NAM models.
    - Use the FICO, Credit Fraud and California Housing datasets to check intelligibility themselves.
    - Use the COMPAS dataset to demonstrate a multi-task NAM model and compare it against the single task NAM both in terms of predictive performance and intelligibility.
- Arik and Pfister (2020)
  - Main: Use five datasets.
  - Other:
    - Use 6 synthetic datasets for which only a subset of the features is relevant to compare TabNet against competitors and demonstrate feature selection capabilities, including per-data-point feature selection.

    - Use the Mushroom and the Adult dataset in addition to the 6 synthetic datasets to test interpretibility.

- Bahri et al. (2022)
  - Main: OpenML-CC18, but excluding the following image datasets: MNIST, FashionMNIST, and CIFAR10.
  - Other: None.
- Borisov et al. (2023): Uses datasets solely for benchmarking dataset generators.
- Borisov et al. (2022)
  - Main: Use 5 datasets.
  - Other: Use Adult to benchmark interpretibility of the tabular deep learning models.
- Buturović and Miljković (2020)
  - Main: Use 1 dataset.
  - Other: None.
- Cai et al. (2021)
  - Main: Use 5 datasets, mostly recommendation and click prediction datasets, but their exact preprocessing and usage is not clear from the paper. The datasets hosted by the authors does not match the raw representation of the datasets.
  - Other: None.
- Chen et al. (2022)
  - Main: Use 7 datasets.
  - Other: None.
- Dubey et al. (2022)
  - Main: Use 4 datasets in Section 4.1 (Measuring Benchmark Performance) and another 9 datasets that they describe as commonly being used in the interpretability literature (but use them for performance evaluation nevertheless)(same as Radenovic et al. (2022) except that this work also uses the critical COMPAS dataset).
  - Other: Use 3 image classification datasets (Caltech-UCSD Birds, iNaturalists "Birds" and Common Objects Datasets; same as Radenovic et al. (2022)).
- Gorishniy et al. (2021)
  - Main: Use 11 datasets.

  - Other: They dropped four datasets (Bank, Kick, MiniBooNe, Click) from the experiments that they found to be non-informative, i.e., where all models perform similar (and report on these in the appendix). Furthermore, they conduct additional experiments on several synthetic datasets to further study the performance of FT-Transformer and a standard ResNet.

- Gorishniy et al. (2022)
  - Main: Use 11 datasets (different to the ones used by Gorishniy et al. (2021)).
  - Other: They reused synthetic data generated by Gorishniy et al. (2021).

- Grinsztajn et al. (2022)
  - Main:
  - Other:

- Hollmann et al. (2023)
  - Main: Use 30 datasets (from the OpenML-CC18).
  - Other: The paper contains further experiments using the moons and circles toy datasets as well as iris and wine datasets to visualize predictions, a subset of the AutoML benchmark with the same characteristics as the 30 datasets used in the main experiment (5 datasets), 149 validation datasets from OpenML that follow the characteristics from the datasets used in the main paper. Finally, they used 18 larger datasets from the AutoML benchmark to study generalization to larger datasets.

- Huang et al. (2020)
  - Main: The main paper reports 15 binary classification datasets and the appendix reports a superset of 20 binary classification tasks. We could not find an explanation why these five tasks should not be considered in the main experiments and therefore assume that all 20 datasets constitute the main experiment.
  - Other: None.

- Joseph and Raj (2022)
  - Main:
  - Other:o

- Kadra et al. (2021)
  - Main: Use 40 datasets.
  - Other: None.

- Kossen et al. (2021)
  - Main: Use 10 datasets.

  - Other: Use MNIST and CIFAR-10 as image datasets, and create a synthetic task based on the UCI Protein regression dataset for which they know the ground-truth interactions.

- Kotelnikov et al. (2023): Uses datasets solely for benchmarking dataset generators.
- Levin et al. (2023):
  - Main: Multi-task dataset based on Meta-MIMIC.
  - Other: 2 multi-label datasets.
- Popov et al. (2020)
  - Main: Use 6 datasets.
  - Other:
- Radenovic et al. (2022)
  - Main: Use 12 datasets (same as Dubey et al. (2022) except that this work does not use the critical COMPAS dataset).
  - Other: Use 3 image classification datasets (Caltech-UCSD Birds, iNaturalists "Birds" and Common Objects Datasets; same as Dubey et al. (2022)).
- Rubachev et al. (2023)
  - Main: Use 11 datasets.
  - Other: Generate synthetic data following Gorishniy et al. (2021). Also, in the appendix, they study four datasets with more categorical features (Diamond, Black Friday, Brazilian houses, Bank).
- Sarkar (2022)
  - Main: Use 8 datasets.
  - Other: None.
- Schäfl et al. (2022)
  - Main: Use 16 datasets.
  - Other: Use four medium-size datasets.
- Shwartz-Ziv and Armon (2022)
  - Main: Use 11 datasets.
  - Other: None.
- Somepalli et al. (2022)
  - Main: Use 30 datasets.
  - Other: Use MNIST to study the proposed attention mechanism.

- Sun et al. (2019)
  - Main: Use 4 datasets.
  - Other: None
- Yoon et al. (2020)
  - Main: Use 11 datasets, out of which three are publicly available (and one of them being MNIST).
  - Other: None
- Zhu et al. (2021)
  - Main: Use 2 datasets.
  - Other: None.

A.5 Dataset Peculiarities

- FICO/HELOC:
  - Borisov et al. (2022) used a cleaned version of the dataset.
  - Agarwal et al. (2021) used it as a regression dataset.
  - Dubey et al. (2022) use the uncleaned version with columns contaning only missing values.
  - Radenovic et al. (2022) used the uncleaned version, but have more features than the original dataset.
- Currently, there are three Higgs datasets in use, but we are unsure how we can decide which one is the correct one.
- Buddy: unclear how to use this as it has time stamps
- The Microsoft dataset was correctly identified as a ranking problem by most studies. However, one paper used it as a classification dataset.
- Cardiovascular Disease dataset: completely unknown origin
- Tours and Travels Customer Churn Prediction: completely unknown origin
- Epsilon is a synthetic dataset, and we are unsure what qualifies it as a tabular dataset. We could not find a publication introducing the competition and were only able to recover a slideset from a video hosting website (Sonnenburg and Franc, 2008) and find a second-hand description by a participant of the competition introducing the dataset (Boullé, 2009).
- Different versions of the same paper (the arXiv version of Somepalli et al. (2022) and the ICLR submission) used different datasets, which makes it hard to decide which one is the experimental setup that is preferred by the authors.
- Three works used the ethically questionable COMPAS dataset (Bao et al., 2021) without further explanation.

A.6 Retracted Datasets

We checked the most-common datasets from Table 6 to see whether the datasets are either retracted or whether someone suggested not to use them any more:

- Adult (Ding et al., 2021)
- German Credit (Grömping, 2019)
- Compas (Bao et al., 2021)
- Iris (Poisot, 2020)
- Diabetes (Radin, 2017)
- Boston Housing (Carlisle, 2019)

Appendix B. Towards a crisp definition of tabular data

As mentioned in the main paper, we first define different categories of how data can arrive in a tabular form:

1. Pixels/Raw: Raw recordings, e.g. images or ;Game: Game database, e.g. chess data
2. Homogeneous Extracted Features (HomE): When a feature extractor applied to raw data (for example, an image) yields similar features, e.g., counts of a pattern, distances to cluster means or a filter applied at multiple locations of an image.
3. Heterogeneous Extracted Features (HetE): Different concepts extracted from one source of raw data without any additional features. This could be e.g. summary statistics of gene expression data or neuroscientific recordings without patient meta-data such as age.
4. Semi-Tabular: Different concepts extracted from one or more sources of raw data with maybe additional information (similar to HetE, but this underlines that the data is somewhat multimodal). Examples of this are often medical data where extracted features are enriched with patient information, such as the arrhythmia dataset (https://openml.org/d/5)
5. Tabular: Highly structured data with features on different scales, e.g., patient records, customer data or sales data extracted from a relational database, etc.

These definitions are ordered from most raw to most tabular. We are aware that with this definition, there will be cases in which it will be hard to decide. Nevertheless, we believe this is an improvement over the current status quo, in which the definitions are more vague. However, as also mentioned in the paper, we also need to take the into account whether a representation makes sense for a dataset at hand given the state-of-the-art in modeling for such a data modality. While it might be necessary for some domain such as images to use raw data to obtain peak performance, there might be other domains where extracted features with a model for tabular data might give the best performance.

We believe that such a definition will allow us to define tabular data better than other definitions from the literature, e.g., Borisov et al. (2022) state that tabular data is "heterogeneous data that usually contain a variety of attribute types", Grinsztajn et al. (2022) define it as "tabular data, made of heterogeneous features, small sample sizes, extreme values" and also not being deterministic (game datasets), while Cai et al. (2021) state that "structured data is generally stored in a set of tables (relations) T1, T2, ... of columns and rows, which can be extracted from a relational database with feature extraction queries, e.g., the projection, natural join, and aggregation of these tables in the database".

Appendix C. Experimental Setup

C.1 Implemented Methods

We have recently seen an influx of DL models for tabular data. In this paper, we restrict our study to only a few models so that we can focus on the datasets rather than the methods. We started using code from the recent study by Grinsztajn et al. (2022) and extended the code with further variants of gradient boosting (CatBoost (Prokhorenkova et al., 2018) and LightGBM (Ke et al., 2017)), and finally, we added two more baseline models (LogReg and MLP from scikit-learn). We briefly describe each of the models in the following (implementation details are also provided on our codebase):

- Gradient Boosting Models
  - **XGBoost** (Chen and Guestrin (2016)): XGBoost is a very popular implementation of regularized Gradient boosting trees that uses level-wise tree growth and histogram-based node splitting.
  - **LightGBM** (Ke et al. (2017)): LightGBM employs leaf-wise tree growth and histogram-based node splitting. It also allows to select novel speedup techniques Gradient-based One Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) but does not enable them by default).
  - **CatBoost** (Prokhorenkova et al. (2018)): CatBoost uses oblivious decision trees as base learners, ordered boosting to combat overfitting, and employs encoding similar to mean target encoding for categorical features.
  - **HistGradientBoosting**- Inspired by LightGBM, HistGradientBoosting bins continuous features to accelerate learning. This baseline is implemented in scikit-learn (Pedregosa et al., 2011).
- Deep Neural Network Models
  - **FT-Transformer** (Gorishniy et al. (2021)): Extends the original transformer as defined in Vaswani et al. (2017) by adding a feature tokenizer module to generate embeddings for both numerical and categorical features.
  - **SAINT** (Somepalli et al. (2022)): Applies self-attention (attention between features) and introduces intersample attention (attention between rows, akin to learning the distance metric for nearest neighbour classification).

  - **ResNet** (He et al. (2016)): A popular architecture in image recognition that suggests using skip connections to learn deeper networks. Gorishniy et al. (2021) suggest it as a robust DL baseline for tabular data.

- Baseline Models (all implemented in scikit-learn (Pedregosa et al., 2011)):
  - **LogReg**: estimates the probability of the outcome using the logistic function.
  - **RF** (Breiman, 2001): an ensemble learning method that constructs multiple decision trees and combines their predictions to make accurate predictions.
  - **MLP**$_{sk}$: Simple feed-forward network.
  - **MLP**$_{pt}$ (PyTorch): Simple feed-forward neural network which is also suggested as a robust DL baseline for tabular data by Gorishniy et al. (2021).

C.1.1 Early Stopping

Early stopping is a regularization technique that is used for gradient boosting ensembles as well as for DL models to avoid overfitting. The idea of early stopping is to train an iterative model on a training set and score it on a separate data set after every iteration. Then, one always keeps a copy of the model for the best score on this separate data set. At the end of the training, instead of returning the model after all iterations, one returns the best recorded models as measured on the separate set as well. Following previous studies, we implement early stopping as well. Early stopping also has some degrees of freedom: the dataset on which to decide whether to stop and the so-called patience to stop after t iterations of no further improvement. Generally speaking, one can perform early stopping on the validation set used to tune hyperparameters or split away a separate training set. The first option increases the chances of overfitting because we look at the validation set more often, while the second reduces the effective size of the training set. Another popular option we do not consider because it would increase the compute cost too much is conducting cross-validation and then using the average number of iterations as the number of iterations for a training run on the full dataset. For scikit-learn models, only the second option is possible; therefore, we follow it. For the deep learning and gradient boosting methods that are not implemented in scikit-learn we were able to configure the data set used and found in early experiments that using the validation set results in better generalization performance and therefore used that.

C.1.2 Data preprocessing

We decided to go for very lean preprocessing pipelines. By this, we mean that we let the models handle the data preprocessing themselves when possible. We chose this strategy to not influence the results by data preprocessing and in order to make full benefit of the developments of advanced methods such as CatBoost, which has effective methods for handling categorical data. The only preprocessing we applied for all models was removing constant columns. For all deep learning models, we used the following preprocessing:

- Categorical attributes: Categorical embedding where we treat missing values and unknown categories at test time as the same.

- Numerical attributes: Median imputation of missing values and Quantile transformation to transform all attributes to follow a Gaussian distribution.

The Gradient Boosting models do not require any preprocessing and can all handle categorical data, missing values and unscaled numerical features by construction. Solely for CatBoost we had to transform the categorical data to strings because this is how CatBoost handles categorical features. Last but not least, we used a standard scikit-learn preprocessing pipeline for the baseline models:

- We used OneHotEconding to transform categorical features into numerical features, added an additional column for missing values and dropped unknown values at test time.
- For numerical features, we use a mean imputation for missing values.

C.2 Experimental Protocol

We use 100 iterations of Random Search (Bergstra and Bengio, 2012) to optimize the hyperparameters of each model. We provide the search spaces in Appendix D. For the datasets we use OpenML tasks with four folds. However, we executed the models only on the first fold, which leads to a test set with 25% of the data. We use 20% of the training data as a validation set. Following previous work (Grinsztajn et al., 2022) we treat the data in a train-validation-test protocol and do not refit the models. For CPU models (models implemented in scikit-learn and other gradient boosting models), we allocate 12 GB and 1 CPU and give a time limit of 3 hours. We used NVIDIA GeForce RTX 2080 Ti for DL models and Intel(R) Xeon(R) Gold 6242 CPU @ 2.80GHz for all other models.

Appendix D. Search Spaces

Table 5: Model configuration space for CatBoost.

Name	Type	Default	Range
learning_rate	UniformFloat	0.03	[0.01, 0.3]
max_depth	UniformInt	6	[2, 12]
reg_lambda	UniformFloat	3	[0.5, 30.0]

Table 6: Model configuration space for FT-Transformer.

Name	Type	Default	Range
batch_size	Categorical	256	(64, 256, 512, 1024)
d_token	UniformInt	192	[64, 512]
lr	UniformFloat	0.0001	[1e-05, 0.001]
lr_scheduler	Categorical	False	(True, False)
module__activation	Categorical	reglu	reglu
module__attention_dropout	UniformFloat	0.2	[0.0, 0.5]
module__d_ffn_factor	UniformFloat	1.3333333333	[0.6666666666666666, 2.6666666666666665
module__ffn_dropout	UniformFloat	0.1	[0.0, 0.5]
module__initialization	Categorical	kaiming	kaiming
module__kv_compression	Categorical	True	(True, False)
module__kv_compression_sharing	Categorical	headwise	('headwise', 'key-value')
module__n_heads	Categorical	8	8
module__n_layers	UniformInt	3	[1, 6]
module__prenormalization	Categorical	True	True
module__residual_dropout	UniformFloat	0.0	[0.0, 0.5]
module__token_bias	Categorical	True	True
optimizer	Categorical	adamw	adamw
optimizer__weight_decay	UniformFloat	1e-05	[1e-08, 0.001]

Table 7: Model configuration space for HistGradientBoosting.

Name	Type	Default	Range
learning_rate	NormalFloat	0.010000000000000005	[2.2250738585072014e-308, 65535.0]
max_depth	Categorical	None	('None', 2, 3, 4)
max_leaf_nodes	NormalInt	31	[2, 65535]
min_samples_leaf	NormalInt	20	[1, 65535]

Table 8: Model configuration space for LightGBM.

Name	Type	Default	Range
lambda_l1	UniformFloat	1e-08	[1e-08, 10.0]
lambda_l2	UniformFloat	1e-08	[1e-08, 10.0]
learning_rate	UniformFloat	0.1	[0.01, 0.3]
num_leaves	UniformInt	31	[2, 4096]

Table 9: Model configuration space for MLP_{pt}.

Name	Type	Default	Range
batch_size	Categorical	256	(64, 256, 512, 1024)
lr	UniformFloat	0.001	[1e-05, 0.01]
lr_scheduler	Categorical	True	(True, False)
module__d_embedding	UniformInt	128	[64, 512]
module__d_layers	UniformInt	256	[16, 1024]
module__dropout	Categorical	0.0	0.0
module__n_layers	UniformInt	4	[1, 8]
optimizer	Categorical	adamw	adamw

Table 10: Model configuration space for ResNet.

Name	Type	Default	Range
batch_size	Categorical	256	(64, 256, 512, 1024)
lr	UniformFloat	0.001	[1e-05, 0.01]
lr_scheduler	Categorical	True	(True, False)
module__activation	Categorical	reglu	reglu
module__d	UniformInt	256	[64, 1024]
module__d_embedding	UniformInt	128	[64, 512]
module__d_hidden_factor	UniformFloat	2.0	[1.0, 4.0]
module__hidden_dropout	UniformFloat	0.2	[0.0, 0.5]
module__n_layers	UniformInt	8	[1, 16]
module__normalization	Categorical	batchnorm	('batchnorm', 'layernorm')
module__residual_dropout	UniformFloat	0.2	[0.0, 0.5]
optimizer	Categorical	adamw	adamw
optimizer__weight_decay	UniformFloat	1e-07	[1e-08, 0.001]

Table 11: Model configuration space for SAINT.

Name	Type	Default	Range
args__batch_size	Categorical	64	(32, 64, 256)
args__lr	Categorical	0.0001	0.0001
args__val_batch_size	Categorical	32	32
params__depth	Categorical	6	(1, 2, 3, 6, 12)
params__dim	Categorical	32	(32, 64, 128, 256)
params__dropout	Categorical	0.1	(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8)
params__heads	Categorical	8	(2, 4, 8)

Table 12: Model configuration space for MLP_{sk}.

Name	Type	Default	Range
activation	Categorical	relu	('tanh', 'relu')
alpha	UniformFloat	0.0001	[1e-07, 0.1]
early_stopping	Categorical	True	(True, False)
hidden_layer_depth	UniformInt	1	[1, 3]
learning_rate	Categorical	constant	('constant', 'invscaling', 'adaptive')
learning_rate_init	UniformFloat	0.001	[0.0001, 0.5]
num_nodes_per_layer	UniformInt	32	[16, 264]

Table 13: Model configuration space for LogReg.

Name	Type	Default	Range
C	UniformFloat	1.0	[1e-12, 1.6094379124341003]
fit_intercept	Categorical	True	(True, False)
penalty	Categorical	l2	('l2', 'None')

Table 14: Model configuration space for RF.

Name	Type	Default	Range
bootstrap	Categorical	True	(True, False)
criterion	Categorical	gini	('gini', 'entropy')
max_depth	Categorical	None	('None', 2, 3, 4)
max_features	Categorical	sqrt	('sqrt', 'log2', 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 'None'
min_impurity_decrease	Categorical	0.0	(0.0, 0.01, 0.02, 0.05)
min_samples_leaf	UniformInt	1	[1, 50]
min_samples_split	Categorical	2	(2, 3)

Table 15: Model configuration space for XGBoost.

Name	Type	Default	Range
colsample_bylevel	UniformFloat	1	[0.5, 1.0]
colsample_bytree	UniformFloat	1	[0.5, 1.0]
gamma	UniformFloat	1e-08	[1e-08, 7.0]
learning_rate	UniformFloat	0.3	[1e-05, 0.7]
max_depth	UniformInt	6	[1, 11]
min_child_weight	UniformFloat	1	[1.0, 100.0]
reg_alpha	UniformFloat	1e-08	[1e-08, 100.0]
reg_lambda	UniformFloat	1e-08	[1e-08, 100.0]
subsample	UniformFloat	1	[0.5, 1.0]