A Closer Look at Deep Learning Methods on Tabular Datasets

Han-Jia Ye YEHJ@LAMDA.NJU.EDU.CN
Si-Yang Liu* LIUSY@LAMDA.NJU.EDU.CN
Hao-Run Cai* CAIHR@LAMDA.NJU.EDU.CN
Qi-Le Zhou ZHOUQL@LAMDA.NJU.EDU.CN
De-Chuan Zhan ZHANDC@LAMDA.NJU.EDU.CN
School of Artificial Intelligence, Nanjing University, China
National Key Laboratory for Novel Software Technology, Nanjing University, 210023, China

Editor: My editor

Abstract

Tabular data is prevalent across diverse domains in machine learning. With the rapid progress of deep tabular prediction methods, especially pretrained (foundation) models, there is a growing need to evaluate these methods systematically and to understand their behavior. We present an extensive study on TALENT, a collection of 300+ datasets spanning broad ranges of size, feature composition (numerical/categorical mixes), domains, and output types (binary, multi-class, regression). Our evaluation shows that ensembling benefits both tree-based and neural approaches. Traditional gradient-boosted trees remain very strong baselines, yet recent pretrained tabular models now match or surpass them on many tasks, narrowing—but not eliminating—the historical advantage of tree ensembles. Despite architectural diversity, top performance concentrates within a small subset of models, providing practical guidance for method selection. To explain these outcomes, we quantify dataset heterogeneity by learning from meta-features and early training dynamics to predict later validation behavior. This dynamics-aware analysis indicates that heterogeneity—such as the interplay of categorical and numerical attributes—largely determines which family of methods is favored. Finally, we introduce a two-level design beyond the 300 common-size datasets: a compact TALENT-tiny core (45 datasets) for rapid, reproducible evaluation, and a TALENT-extension suite targeting high-dimensional, many-class, and very large-scale settings for stress testing. In summary, these results offer actionable insights into the strengths, limitations, and future directions for improving deep tabular learning.

Keywords: machine learning on tabular data, deep tabular learning, tabular benchmarks

1 Introduction

Machine learning systems are now deployed across a wide spectrum of real-world applications. Although raw data may arrive in varied and complex forms, it is commonly cast into vectorized representations through feature engineering or learned encoders. For example, image data can be converted into vectors using feature extractors such as SIFT (Szeliski, 2022), while modern approaches rely on convolutional layers to learn representations automatically (Goodfellow et al., 2016). In supervised learning, the objective is to map these vectors to labels—discrete

*. These authors contributed equally to this work

for classification or continuous for regression—and to generalize to unseen instances drawn from the same distribution.

Among data modalities, *tabular data* plays a central and pervasive role. It represents perhaps the most general and widely used form of supervised learning, organizing information as instances (rows) and attributes (columns), and it naturally arises in applications such as click-through rate prediction (Yan et al., 2014; Juan et al., 2016; Zhang et al., 2016; Guo et al., 2017), healthcare (Hassan et al., 2020), medical analysis (Schwartz et al., 2007; Subasi, 2012), and e-commerce (Nederstigt et al., 2014). Its broad adoption stems from flexibility: scales and domains vary widely, attributes may be numerical or categorical (including binary or ordinal), and features often mix heterogeneous statistical behaviors.

Tabular machine learning methods have evolved significantly over time. Classical approaches, such as Logistic Regression (LR), Support Vector Machines (SVM), Multi-Layer Perceptrons (MLP), and decision trees, have served as the foundation for a wide range of algorithms (Bishop, 2006; Hastie et al., 2009; Mohri et al., 2012). For practical applications, tree-based ensemble methods like Random Forest (Breiman, 2001), XGBoost (Chen and Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2018) have demonstrated consistent advantages across various tasks. Inspired by the success of Deep Neural Networks (DNNs) in domains such as vision and natural language processing (Simonyan and Zisserman, 2015; Vaswani et al., 2017; Devlin et al., 2019), recent efforts have adapted DNNs for tabular classification and regression tasks (Wang et al., 2017; Song et al., 2019; Badirli et al., 2020; Gorishniy et al., 2021; Borisov et al., 2024). Modern practice shows that carefully regularized and tuned MLPs can be highly competitive (Kadra et al., 2021; Holzmüller et al., 2024), while tokenization/attention designs bring additional modeling capacity on mixed-type features (Huang et al., 2020; Chen et al., 2024).

A recent and influential development is the emergence of tabular foundation models. These methods pretrain on large collections of synthetic/real tasks and leverage in-context learning to enable fast adaptation to new datasets with minimal tuning (Hollmann et al., 2023; Ma et al., 2024; Hollmann et al., 2025; Qu et al., 2025; Zhang et al., 2025). In many scenarios, such models close a substantial portion of the historical performance gap between tree ensembles and deep architectures, while retaining attractive deployment properties (*e.g.*, few-shot adaptation, fast inference). Understanding where and why these gains arise—relative to strong classical ensembles and modern deep baselines—requires evaluations that are both *broad* (to avoid benchmark artifacts) and *up-to-date* (to reflect the latest methods).

Unlike vision and language where widely adopted resources such as ImageNet enable consistent comparisons (Deng et al., 2009), the tabular domain lacks a unifying framework for systematic evaluation (Borisov et al., 2024). Existing tabular datasets are scattered across UCI (Hamidieh, 2018), OpenML (Vanschoren et al., 2014), and Kaggle, and they vary substantially in size, feature composition, and application domain. To reflect how tabular learning is used in practice over vectorized representations, evaluations must draw on datasets spanning diverse domains, feature types, and scales—rather than rely on narrow collections that risk benchmark artifacts. This need for broad coverage is further underscored by the "no free lunch" theorem,[1] which implies that empirical superiority only emerges within realistic, bounded task families. Consequently, collecting datasets that cover a wide range

1. Formally, the theorem states that if we average performance uniformly over all possible tasks with training and test sets independent, all algorithms perform equivalently on average (Wolpert, 1996). In practice,

of real-world settings is essential to mimic practical conditions and to obtain meaningful insights into method behavior.

Prior studies also show that limited coverage and outdated baselines can yield biased conclusions (Macià et al., 2013). While average rank remains a common summary across datasets (Delgado et al., 2014; Grinsztajn et al., 2022; McElfresh et al., 2023), complementary criteria have been advocated (Delgado et al., 2014; McElfresh et al., 2023; Holzmüller et al., 2024; Gorishniy et al., 2025), and recent work highlights challenges such as dataset aging (Kohli et al., 2024) and reliance on expert feature engineering (Tschalzev et al., 2024). In this paper, we address these gaps with TALENT, a collection of 300+ datasets covering binary, multi-class, and regression tasks across domains including education, biology, chemistry, and finance. TALENT spans a wide range of sizes, feature types, and imbalance ratios to support fair, up-to-date, and comprehensive comparisons. Based on this resource, we aim to answer three key questions:

Is there a consistent empirical picture across multiple tabular datasets? We compare 40 representative tabular methods under a unified protocol using multiple criteria (average ranks, statistical tests, probability of best performance, aggregated errors). As the number of datasets grows, conclusions stabilize: while no single method dominates universally, top performance consistently concentrates within a small shortlist of models, and ensembling benefits both tree-based and deep tabular approaches. In the presence of recent foundation-style models, our results refine the long-standing "trees vs. DNNs" discussion.

How can we measure dataset heterogeneity, and how does it affect the behavior of deep tabular methods? We quantify heterogeneity using meta-features and study their relation to model behavior by predicting later validation dynamics from meta-features and early training signals. This dynamics-aware view highlights the role of feature-space heterogeneity—especially the interplay between categorical and numerical attributes, sparsity, and entropy variance—in shaping when specific method families succeed or fail.

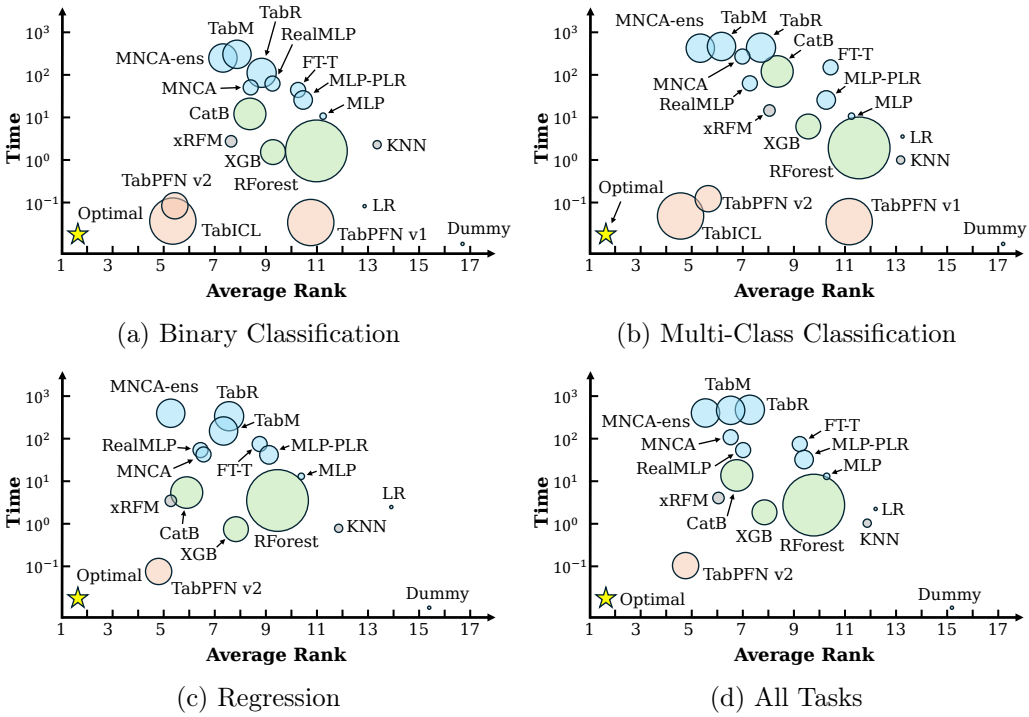
Can we support lightweight yet informative evaluation? We design a two-level evaluation strategy: a compact TALENT-tiny (45 datasets, ∼15% of the full suite) for rapid prototyping that balances tree-friendly and DNN-friendly cases under more stricter quality rules, and a supplemental TALENT-extension that stress-tests methods on high-dimensional, many-class, and very large-scale datasets. Together, they enable efficient experimentation and targeted analysis beyond the common-size regime.

The contributions of this paper are summarized as:[2]

- A large-scale, up-to-date evaluation of 40 tabular methods over 300+ datasets with multiple complementary criteria, showing that top performance concentrates within a small shortlist and that ensembling benefits both tree-based and DNN-based methods.
- A dynamics-aware heterogeneity analysis that maps meta-features and early learning signals to later validation behavior, identifying the most predictive sources of heterogeneity (*e.g.*, categorical–numerical interplay, sparsity, entropy variance).
- A two-level evaluation design: TALENT-tiny (∼15% of TALENT) for fast, balanced comparisons under strict quality controls, and TALENT-extension for stress-testing on high-dimensional, many-class, and large-scale regimes.

however, real-world applications focus on subsets of tasks that exhibit inductive biases and priors, within which certain algorithms may consistently outperform others.

2. The code and the dataset link is available at `https://github.com/LAMDA-Tabular/TALENT`.



(a) Binary Classification

(b) Multi-Class Classification

(c) Regression

(d) All Tasks

Figure 1: Performance–efficiency–size comparison of representative tabular methods on Talent for (a) binary classification, (b) multi-class classification, (c) regression, and (d) all tasks. The performance is measured by the average rank of all methods (lower is better). The efficiency is measured by the average training time in seconds (lower is better). The model size is measured based on the average size of all models (the larger the radius, the larger the model).

2 Related Work

2.1 Learning with Tabular Data

Tabular data is a common format across various applications, such as click-through rate prediction (Richardson et al., 2007; Zhang et al., 2016) and time-series forecasting (Ahmed et al., 2010; Padhi et al., 2021). The most common supervised settings are standard classification and regression, where models learn mappings from vectorized instances to discrete or continuous targets and are evaluated on i.i.d. test data (Bishop, 2006; Hastie et al., 2009; Mohri et al., 2012). Tree-based methods, such as Random Forest (Breiman, 2001), XGBoost (Chen and Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2018), remain highly competitive due to their strong inductive bias for heterogeneous features and interactions. Because both model families and hyperparameters strongly influence generalization (Delgado et al., 2014), automated selection and tuning methods (*e.g.*, AutoML) are widely used (Feurer et al., 2015; Guyon et al., 2019). Beyond supervised prediction, related tabular tasks include clustering (Rauf et al., 2024; Svirsky and Lindenbaum, 2024), anomaly detection (Shenkar and Wolf, 2022; Han et al., 2022; Yin et al., 2024), data generation (Xu et al., 2019; Hansen et al., 2023; Vero et al., 2024), open-environment learning (Ye et al.,

2021; Hou et al., 2023a,b; Xu et al., 2023), symbolic regression (Wilstrup and Kasak, 2021; Cava et al., 2021), and streaming scenarios (Zhou, 2024; Rubachev et al., 2025a).

2.2 Deep Tabular Data Learning

Deep models have been adapted to tabular prediction to learn representations directly from inputs and capture complex nonlinear interactions (Cheng et al., 2016; Guo et al., 2017; Popov et al., 2020; Arik and Pfister, 2021; Katzir et al., 2021; Chen et al., 2022). Architectures commonly explored include residual MLPs and Transformer variants (Gorishniy et al., 2021; Hollmann et al., 2023; Zhou et al., 2023; Chen et al., 2024), complemented by regularization and augmentation tailored for tabular data (Ucar et al., 2021; Bahri et al., 2022; Rubachev et al., 2022). A key observation is that carefully tuned, relatively simple networks can be highly competitive (Kadra et al., 2021; Holzmüller et al., 2024).

Advantages of deep tabular models. DNNs excel at modeling higher-order interactions via nonlinear feature composition (Wang et al., 2017, 2021), support end-to-end multi-task learning and representation sharing (Somepalli et al., 2022; Wu et al., 2024), and are trained by gradient-based optimization that flexibly accommodates new objectives with minimal redesign. They also integrate naturally into multi-modal systems combining tables with images, audio, or text (Gorishniy et al., 2021; Jiang et al., 2024a).

Design trends. Early neural approaches often mimicked tree workflows or emphasized feature-correlation modeling (Cheng et al., 2016; Guo et al., 2017; Popov et al., 2020; Chang et al., 2022). Subsequent work refined MLPs with principled initialization, normalization, and regularization (Gorishniy et al., 2021; Kadra et al., 2021; Holzmüller et al., 2024). Token/attention models adapt Transformer-style processing to heterogeneous columns (Huang et al., 2020; Chen et al., 2024; Zhou et al., 2023). Advanced ensemble strategies are also investigated and incorporated in deep tabular prediction (Gorishniy et al., 2025). Retrieval/neighborhood-based formulations (*e.g.*, context-based prediction or exemplar conditioning) improve robustness and adaptation (Gorishniy et al., 2024; Ye et al., 2025b).

Pretrained/foundation models. Recent work pretrains neural predictors on large collections of (often synthesized) tabular tasks and deploys them to novel datasets via *in-context learning* without explicit gradient updates (Hollmann et al., 2023; Ma et al., 2024; van Breugel and van der Schaar, 2024; Hollmann et al., 2025; Qu et al., 2025; Zhang et al., 2025). Parameter- and data-efficient adaptation strategies further improve performance across regimes (*e.g.*, lightweight fine-tuning and localized adapters) (Feuer et al., 2024; Thomas et al., 2024; Liu and Ye, 2025). Several studies have also evaluated and analyzed the behavior of recent foundation models such as TabPFN v2 (Ye et al., 2025a; Rubachev et al., 2025b). Overall, these foundation-style approaches substantially improve data efficiency and increasingly narrow the historical advantage of tree ensembles. Comprehensive surveys situate the field along a spectrum from task-specific to cross-task to general paradigms (Borisov et al., 2024; Jiang et al., 2025).

2.3 Tabular Prediction with Semantic Information and LLMs

Recent work has begun to exploit the semantic information encoded in feature names, metadata, and textual descriptions to improve tabular prediction. One strategy is to

transform features into embeddings (tokens), thereby mapping tables of varying sizes into a standardized token space. Pretrained models such as Transformers can then encode transferable knowledge that benefits downstream tasks (Yan et al., 2024b; Ye et al., 2024a). Another line of research reformulates tabular inputs as natural language sequences, enabling large language models (LLMs) to directly learn feature–label relationships. For instance, LIFT (Dinh et al., 2022) and TabLLM (Hegselmann et al., 2023) serialize tables into textual prompts for fine-tuning or few-shot prediction. UniPredict (Wang et al., 2023) enhances this paradigm by enriching prompts with metadata and task-specific instructions, while IngesTables (Yak et al., 2023) integrates external reasoning steps for multi-hop tabular inference. More recent efforts (Gardner et al., 2024; Wen et al., 2024) propose specialized instruction-tuning techniques that further adapt LLMs for tabular contexts. These approaches demonstrate the promise of leveraging prior knowledge embedded in LLMs for tabular tasks, particularly in low-data regimes. However, their effectiveness depends heavily on the richness of semantic information available (*e.g.*, meaningful feature names or metadata) and can be limited by scalability issues when serializing high-dimensional tables into text.

2.4 Tabular Methods Evaluations

Comprehensive evaluations are essential for understanding how tabular methods behave before deployment. Several studies have attempted to benchmark tabular models, and differ in (i) the breadth and realism of their dataset coverage, (ii) the families of approaches they include, and (iii) the evaluation protocols they adopt.

Dataset Coverage. Early benchmarks focused on relatively small or narrow collections of datasets. For example, Delgado et al. (2014) evaluated 179 classifiers across 121 datasets, concluding that Random Forest variants were often the best performers, though later work by Wainberg et al. (2016) highlighted flaws in the evaluation protocol. More recent studies have expanded the coverage modestly: Kadra et al. (2021) studied MLPs on 40 classification datasets, while Gorishniy et al. (2021) examined MLPs, ResNets, and Transformer-based models on 11 datasets. Grinsztajn et al. (2022) used 45 datasets to investigate differences between tree-based and deep methods. A broader effort by McElfresh et al. (2023) included 176 classification datasets and 19 methods, but excluded regression tasks and applied strict limits on training data size and time, which may have disadvantaged deep models. Overall, most prior benchmarks underrepresent the diversity of real-world tabular tasks, particularly in regression, high-dimensional, and large-scale settings.

Target Approaches. Benchmark scope also varies by method family. Classical evaluations emphasized tree ensembles and linear models; more recent work incorporates modern deep tabular methods. For instance, McElfresh et al. (2023) compared classical models with modern deep approaches, finding TabPFN (Hollmann et al., 2023) to be a strong performer. Other studies have emphasized specific architectures, such as MLP variants (Kadra et al., 2021), ResNets (Gorishniy et al., 2021), or Transformers (Chen et al., 2024), but often in limited settings that make it difficult to generalize their findings.

Evaluation Protocols. Benchmarking protocols also vary considerably. Some studies adopt uniform hyperparameter settings or limited tuning budgets, which can bias results against deep models that require careful tuning. For example, the strict time and trial limits in McElfresh et al. (2023) may have led to suboptimal evaluations for complex neural

architectures. Recent work has also explored alternative evaluation perspectives, including dataset quality (Erickson et al., 2025), dataset age (Kohli et al., 2024), reliance on expert-crafted features (Tschalzev et al., 2024), temporal characteristics of tabular data (Rubachev et al., 2025a; Tschalzev et al., 2024), and cross-validation as well as post-hoc ensemble strategies to boost performance (Erickson et al., 2025).

Summary. Effective assessment requires datasets that span classification *and* regression, cover diverse domains and feature types, and pair fair, well-tuned protocols with appropriate statistical comparisons. Achieving this breadth entails computational trade-offs. With rapid advances in deep tabular learning (Holzmüller et al., 2024; Ye et al., 2025b; Beaglehole et al., 2025)—especially pretrained foundation models (Hollmann et al., 2025; Qu et al., 2025)—there is a pressing need for evaluations that balance wide coverage with rigor, enabling reliable, nuanced conclusions about the strengths and limitations of modern tabular methods.

3 Preliminary

3.1 Learning with Tabular Data

A tabular dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ is formatted as N examples and d features (attributes), corresponding to N rows and d columns in a table. Each instance $\boldsymbol{x}_i \in \mathbb{R}^d$ is represented by its d feature values. The j-th feature of instance $\boldsymbol{x}_i$, denoted $x_{i,j}$, may be numerical (continuous), $x_{i,j}^{\text{num}} \in \mathbb{R}$, or categorical (discrete), $x_{i,j}^{\text{cat}}$. Categorical features are typically transformed into numerical vectors using encoding strategies such as one-hot or target encoding (Hancock and Khoshgoftaar, 2020).

In a *supervised* prediction task, each instance is associated with a label y_i, where $y_i \in \{1, -1\}$ for binary classification, $y_i \in [C] = \{1, \ldots, C\}$ for multi-class classification, and $y_i \in \mathbb{R}$ for regression. Given $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, the goal is to learn a model f by empirical risk minimization:

$$\min_f \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}} \ell(y_i,\ \hat{y}_i = f(\boldsymbol{x}_i)) + \Omega(f)\ , \tag{1}$$

where $\ell(\cdot, \cdot)$ measures the discrepancy between the predicted label $\hat{y}_i$ and the true label y_i (*e.g.*, cross-entropy for classification), and $\Omega(f)$ is a regularization term. The learned model f is expected to generalize to unseen instances sampled from the same distribution as $\mathcal{D}$.

While this formulation captures the standard setting of supervised tabular learning, we note that it does not encompass all paradigms. Some methods adopt *unsupervised* or *self-supervised pretraining* objectives on tabular data, aiming to learn transferable representations before fine-tuning on supervised tasks (Ucar et al., 2021; Rubachev et al., 2022). Others focus on *generative* modeling of tabular data (Hansen et al., 2023; Vero et al., 2024), or employ *ensemble strategies* that combine multiple predictions from different runs, seeds, or data splits (Erickson et al., 2025). Since such strategies vary widely and often depend on additional design choices, in this work we restrict our evaluation to the *intrinsic supervised performance* of each model, while acknowledging that external ensembles or pretraining can further improve results.

3.2 Representative Tabular Models

We consider several representative families of models for tabular prediction, including classical methods, tree-based methods, and deep neural network (DNN)-based methods.

Classical Methods. As a trivial baseline, we include the "Dummy" approach, which always predicts the majority class for classification or the mean of the target for regression. We further evaluate standard classical methods: Logistic Regression (LR), K-Nearest Neighbors (KNN), and Support Vector Machines (SVM). We also include Recursive Feature Machines (RFM) (Radhakrishnan et al., 2023), which enable kernel machines to learn features by recursively reweighting them via a gradient-inspired mechanism without backpropagation, and their extension xRFM (Beaglehole et al., 2025). For classification tasks, we also include Naive Bayes and the Nearest Class Mean (NCM) (Tibshirani et al., 2002). For regression tasks, Linear Regression replaces LR, and we additionally consider DNNR (Nader et al., 2022).

Tree-based Methods. Tree-based models are widely regarded as strong baselines for tabular learning Delgado et al. (2014). We include Random Forest (Breiman, 2001), as well as gradient-boosting ensembles (Friedman, 2001, 2002) such as XGBoost (Chen and Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2018), all of which are established as highly competitive across tasks (Grinsztajn et al., 2022; McElfresh et al., 2023).

DNN-based Methods. Deep tabular methods vary in their design principles and prediction strategies. We categorize them into the following groups:

- **MLP Variants.** Vanilla MLPs operate directly on raw features and, with careful tuning, can be competitive (Kadra et al., 2021; Yan et al., 2024a). Strong baselines include the implementation in (Gorishniy et al., 2021), MLP-PLR with periodic activations (Gorishniy et al., 2022), and RealMLP with tailored modules/encodings and optimization (Holzmüller et al., 2024). Other variants include SNN (Klambauer et al., 2017) and ResNet-style architectures (Gorishniy et al., 2021).
- **Specially Designed Architectures.** Several works design custom architectures to capture explicit feature interactions. DCNv2 (Wang et al., 2021) combines embeddings, cross layers, and deep networks. TabCaps (Chen et al., 2023a) encapsulates instance features into vectorial representations to enhance representation learning.
- **Token-based Methods.** These methods map feature values into high-dimensional tokens, enabling attention mechanisms to model high-order interactions. Representative approaches include AutoInt (Song et al., 2019), TabTransformer (Huang et al., 2020), FT-Transformer (FT-T) (Gorishniy et al., 2021), and ExcelFormer (Chen et al., 2024), which introduces semi-permeable attention and attentive feed-forward layers.
- **Regularization-based Methods.** These methods enhance generalization through explicit regularization. TANGOS (Jeffares et al., 2023) enforces neuron specialization and orthogonality. SwitchTab (Wu et al., 2024) introduces a self-supervised encoder–decoder framework, while PTaRL (Ye et al., 2024b) calibrates features through prototypes.
- **Tree-mimic Methods.** Inspired by decision trees, these architectures combine neural networks with tree-like structures. NODE (Popov et al., 2020) generalizes oblivious decision trees, GrowNet (Badirli et al., 2020) embeds shallow networks within boosting, and TabNet (Arik and Pfister, 2021) employs sequential attention for feature selection. DANets (Chen et al., 2022) group correlated features to produce higher-level abstractions.

- **Neighborhood-based Methods.** These approaches make predictions by retrieving and weighting similar instances. TabR (Gorishniy et al., 2024) augments a learned predictor with a KNN-style retrieval component, while ModernNCA (Ye et al., 2025b) modernizes classic NCA (Goldberger et al., 2004) for robust, retrieval-based tabular learning.
- **Ensemble-based Methods.** Ensemble-style DNNs share parameters to train multiple predictors efficiently. TabM (Gorishniy et al., 2025) builds on BatchEnsemble (Wen et al., 2020) with an MLP backbone; for analysis we also consider a BatchEnsemble-enhanced variant of ModernNCA (MNCA-ens).
- **Pretrained Foundation Models.** Pretrained tabular transformers enable in-context inference on new datasets with little to no task-specific training or hyperparameter tuning. TabPFN (Hollmann et al., 2023) predicts labels by conditioning directly on the training set. We also include stronger successors that scale to larger datasets: TabPFN v2 (Hollmann et al., 2025) and TabICL (Qu et al., 2025). TabPFN and TabICL currently target *classification*, whereas TabPFN v2 supports both *classification* and *regression*. In addition, we evaluate adaptation methods built on TabPFN, including LocalPFN (Thomas et al., 2024), TuneTables (Feuer et al., 2024), and BETA (Liu and Ye, 2025).

Other Methods. Additional models such as Trompt (Chen et al., 2023b), BiSHop (Xu et al., 2024), ProtoGate (Jiang et al., 2024b), and GRANDE (Marton et al., 2024) fall outside our main categories. We omit them here due to substantially longer training/inference times or because their goals (*e.g.*, extreme efficiency) are orthogonal to our focus on predictive accuracy. Similarly, pretrained variants such as HyperFast (Bonet et al., 2024), TabDPT (Ma et al., 2024), TabFlex (Zeng et al., 2025), and MotherNet (Mueller et al., 2025) are not included, as stronger and more recent foundation models are already covered. Finally, we exclude RealTabPFN (Garg et al., 2025) to avoid potential data overlap: its continual pretraining corpus intersects with our benchmark, which could confound fair evaluation.

4 A Comprehensive Tabular Data Benchmark

This section describes how Talent is constructed, characterizes the datasets it contains, details the quality controls we apply, and highlights why Talent is well-suited for advancing research on tabular learning.

4.1 Design Philosophy

To meaningfully measure the ability of tabular methods across diverse scenarios, the guiding principle of the Talent benchmark is to evaluate models under broad and realistic coverage that mirrors the heterogeneity of real-world applications. The benchmark is built around *two complementary layers of coverage*. First, we construct a large and diverse collection of *common-size* datasets, which form the foundation for standard evaluation. This layer ensures that comparisons are made across a broad range of classical and deep models under typical conditions. Second, we incorporate *specialized* settings—such as high-dimensional feature spaces, many-class classification problems, and very-large-scale datasets—in Talent-extension that reflect more challenging but practically important scenarios. These are evaluated separately (see subsection 7.2), allowing us to analyze scalability and robustness under conditions that go beyond the common-size setting.

In addition to the coverage of dataset sizes, we adopt a *two-level dataset selection strategy* to balance inclusiveness with fairness. The general Talent set is constructed with relatively weak filtering rules, excluding only datasets that present clear quality issues such as label leakage or annotation errors. This design maximizes breadth while minimizing evaluation bias. Complementing this, we curate a core set of datasets, *i.e.*, Talent-tiny, in subsection 7.1 that applies stricter rules to ensure balance across domains, feature types, and scales, and to include both tree-friendly and DNN-friendly tasks. This core set provides a controlled, reliable environment for detailed analysis and rapid iteration.

4.2 Datasets Collection

Talent aggregates datasets from UCI (Hamidieh, 2018), OpenML (Vanschoren et al., 2014), and Kaggle. We first construct the *general* Talent set using a set of filtering and preprocessing rules that emphasize inclusiveness while removing datasets with obvious quality problems. The stricter rules used to define the Talent-*tiny* core set follow the two-level strategy above and are described later.

Initial filtering rules. We begin with the following quality controls:

- **Size.** Exclude datasets with fewer than 500 instances ($N < 500$) or fewer than 5 features ($d < 5$), which tend to yield unstable evaluations due to limited test coverage.
- **Missing values.** Remove datasets with $> 20\%$ missing values to avoid unreliable comparisons.
- **Trivial classification.** Exclude classification datasets that are overly easy (*e.g.*, a simple MLP exceeds 99% accuracy) or dominated by a majority class.
- **Attribute preprocessing.** Drop non-informative attributes (*e.g.*, `id`, `index`, `timestamp`); ordinally encode categorical features (Borisov et al., 2024; McElfresh et al., 2023); and follow Gorishniy et al. (2021) for the remaining preprocessing.
- **Duplicates.** Remove subsets and near-duplicates from UCI/OpenML to ensure uniqueness.
- **Task type correction.** Relabel 22 mis-specified regression datasets with only two unique targets as binary classification.

Multi-version datasets. Some datasets share an origin but differ in collection conditions, feature extraction, or augmentation (*e.g.*, `forex-` at different sampling frequencies, `wine-quality-` under different standards, and `mfeat-` variants of handwritten digits). We retain such versions unless they completely overlap, as they reflect real-world application requirements and resource constraints. These variations provide an opportunity to evaluate how different tabular models handle such practical scenarios: the best method on one version is not always best on others. Removing them has little impact on aggregate rankings when many datasets are included, but retaining them provides valuable insights into consistency across related tasks.

"Easy" datasets revisited. Although the rules above filter out trivial cases, not all models achieve optimal performance on some seemingly easy datasets. For example, `mice_protein_expression` is solved by KNN and RealMLP (Holzmüller et al., 2024), while many strong models underperform. We therefore retain such datasets when performance is not uniformly trivial, as they expose informative discrepancies.

Datasets from other modalities. Talent includes 25 datasets whose features are extracted from images or audio (*e.g.*, `optdigits`, `segment`, `phoneme`). While some recent

analyses argue these are less relevant for tabular evaluation (Kohli et al., 2024; Erickson et al., 2025), in many practical scenarios, only extracted features—not raw modalities—are available due to resource or deployment constraints. We therefore keep them to reflect real-world usage and to test model robustness to such inputs.

Inherent distribution shifts. Some datasets contain implicit distribution shifts, for example, due to temporal splits in the collection where timestamps used during collection create non-stationary distributions (Tschalzev et al., 2024; Rubachev et al., 2025a). Cai and Ye (2025) show that the choice of split strategy between training and validation sets can significantly affect absolute performance values, even though the relative ranking of methods often remains stable. This suggests that while distribution shift complicates reliable generalization, comparative evaluations may still provide useful insights under consistent protocols. In this paper, however, we focus on *standard tabular prediction tasks, where training and test instances are assumed to be drawn from the same underlying distribution*. This assumption allows us to establish a fair and controlled evaluation framework. Nevertheless, we acknowledge that distribution shifts are common in real-world applications, and developing benchmarks that explicitly address such scenarios is an important direction for future research.

Datasets with known leakage. Recent analyses have revealed that a number of widely used tabular datasets suffer from data leakage, most often caused by the inclusion of features that directly or indirectly encode the target variable (Rubachev et al., 2025a; Tschalzev et al., 2025). In our collection, we identify 13 datasets with potential leakage. While such datasets may compromise strict evaluation quality, we deliberately retain them in the general Talent benchmark for two reasons: first, to maintain comparability with prior work, where these datasets have been widely used; and second, to assess the general ability of tabular methods to handle diverse real-world data, including imperfectly curated datasets that are commonly encountered in practice. For rigorous and bias-free evaluation, however, all datasets with identified leakage are excluded from the curated Talent-tiny subset. The detailed list and explanations are provided in Appendix A.

From general to core sets. The rules above define the *general* Talent benchmark, designed for inclusiveness while filtering out datasets with obvious issues. To provide a stricter, more balanced evaluation, we also construct the Talent-*tiny* core set by applying stronger rules: excluding datasets derived from other modalities and with leakages, removing trivial or duplicated variants, and focusing on tasks that balance tree- and DNN-friendly cases. This two-level design allows researchers to use Talent for broad benchmarking while relying on Talent-tiny for controlled, in-depth analysis.

Ultimately, Talent consists of 120 binary-classification, 80 multi-class, and 100 regression datasets, offering both breadth and depth for evaluating tabular learning. The complete list of Talent, the selection for Talent-tiny, and summary statistics for Talent-extension are reported in Table 1.

Table 1: The list of datasets (including names and source URLs) in our proposed benchmark, along with the statistics for each dataset.

Talent													
ID	Name (Source)	Task	Class	Sample	Feature	Tiny	ID	Name (Source)	Task	Class	Sample	Feature	Tiny
1	100-plants-margin	Cls	100	1600	64		151	JapaneseVowels	Cls	9	9961	14	
2	100-plants-shape	Cls	100	1600	64		152	jasmine	Cls	2	2984	144	✓
3	100-plants-texture	Cls	100	1599	64		153	jm1	Cls	2	10885	21	
4	1000-Cameras-Dataset	Reg	–	1038	10		154	Job_Profitability	Reg	–	14480	28	
5	2dplanes	Reg	–	40768	10	✓	155	jungle_chess_endgame	Cls	3	44819	6	✓
6	abalone	Cls	3	4177	8		156	kc1	Cls	2	2109	21	
7	Abalone_reg	Reg	–	4177	8	✓	157	KDD	Cls	2	5032	45	
8	accelerometer	Cls	4	153004	4		158	kdd_ipums_la_97-small	Cls	2	5188	20	
9	ada	Cls	2	4147	48	✓	159	KDDCup09_upselling	Cls	2	5128	49	
10	ada_agnostic	Cls	2	4562	48		160	kin8nm	Reg	–	8192	8	
11	ada_prior	Cls	2	4562	14		161	kr-vs-k	Cls	18	28056	6	
12	adult	Cls	2	48842	14		162	kr-vs-kp	Cls	2	3196	36	
13	Ailerons	Reg	–	13750	40		163	kropt	Cls	18	28056	6	
14	airfoil_self_noise	Reg	–	1503	5		164	Laptop_Prices_Dataset	Reg	–	4441	8	
15	airline_satisfaction	Cls	2	129880	21		165	law-school-admission	Cls	2	20800	11	✓
16	airlines_2000	Cls	2	2000	7	✓	166	led24	Cls	10	3200	24	
17	allbp	Cls	3	3772	29	✓	167	led7	Cls	10	3200	7	
18	allrep	Cls	4	3772	29		168	letter	Cls	26	20000	15	
19	Amazon_employee_access	Cls	2	32769	7		169	Long	Cls	2	4477	19	
20	analcatdata_authorship	Cls	4	841	69		170	longitudinal-survey	Cls	2	4908	16	✓
21	analcatdata_supreme	Reg	–	4052	7	✓	171	madeline	Cls	2	3140	259	
22	archive2	Reg	–	1143	12		172	MagicTelescope	Cls	2	19020	9	
23	archive_r56_Portuguese	Reg	–	651	30		173	mammography	Cls	2	11183	6	
24	artificial-characters	Cls	10	10218	7		174	Marketing_Campaign	Cls	2	2240	27	
25	ASP-POTASSCO	Cls	11	1294	141	✓	175	maternal_health_risk	Cls	3	1014	6	
26	auction_verification	Reg	–	2043	7		176	mauna-loa-atmospheric	Reg	–	2225	6	
27	autoUniv-au4-2500	Cls	3	2500	100		177	mfeat-factors	Cls	10	2000	216	
28	autoUniv-au7-1100	Cls	5	1100	12	✓	178	mfeat-fourier	Cls	10	2000	76	
29	avocado_sales	Reg	–	18249	13		179	mfeat-karhunen	Cls	10	2000	64	
30	bank	Cls	2	45211	16		180	mfeat-morphological	Cls	10	2000	6	
31	bank32nh	Reg	–	8192	32		181	mfeat-pixel	Cls	10	2000	240	
32	bank8FM	Reg	–	8192	8		182	mfeat-zernike	Cls	10	2000	47	
33	Bank_Customer_Churn	Cls	2	10000	10		183	MiamiHousing2016	Reg	–	13932	16	
34	banknote_authentication	Cls	2	1372	4		184	MIC	Cls	2	1649	104	
35	baseball	Cls	3	1340	16		185	mice_protein_expression	Cls	8	1080	75	
36	Basketball_c	Cls	2	1340	11		186	microaggregation2	Cls	5	20000	20	✓
37	Bias_correction_r	Reg	–	7725	21		187	MIP-2016-regression	Reg	–	1090	144	
38	Bias_correction_r_2	Reg	–	7725	21		188	mobile_c36_oversampling	Cls	2	51760	6	
39	bike_sharing_demand	Reg	–	10886	9		189	Mobile_Phone_in_Ghana	Reg	–	3600	14	
40	BLE_RSSI_localization	Cls	3	9984	3		190	Mobile_Price	Cls	4	2000	20	
41	BNG(breast-w)	Cls	2	39366	9		191	Moneyball	Reg	–	1232	14	
42	BNG(cmc)	Cls	3	55296	9		192	mozilla4	Cls	2	15545	4	
43	BNG(echoMonths)	Reg	–	17496	8	✓	193	mv	Reg	–	40768	10	
44	BNG(lowbwt)	Reg	–	31104	9		194	NASA_PHM2008	Reg	–	45918	21	
45	BNG(mv)	Reg	–	78732	10	✓	195	naticusdroid_permissions	Cls	2	29332	86	
46	BNG(stock)	Reg	–	59049	9		196	NHANES_age_prediction	Reg	–	2277	7	
47	BNG(tic-tac-toe)	Cls	2	39366	9		197	Nutrition_Health_Survey	Cls	2	2278	7	
48	boston	Reg	–	506	13		198	okcupid_stem	Cls	3	26677	13	✓
49	Brazilian_houses	Reg	–	10692	8		199	online_shoppers	Cls	2	12330	14	✓
50	California-Housing-Cls	Cls	2	20640	8		200	OnlineNewsPopularity	Reg	–	39644	59	
51	car-evaluation	Cls	4	1728	21		201	optdigits	Cls	10	5620	64	
52	Cardiovascular-Disease	Cls	2	70000	11		202	ozone-level-8hr	Cls	2	2534	72	
53	chscase_foot	Reg	–	526	5		203	ozone_level	Cls	2	2536	36	✓
54	churn	Cls	2	5000	20		204	page-blocks	Cls	5	5473	10	
55	Click_prediction	Cls	2	39948	3		205	Parkinson_Sound_Record	Reg	–	1040	26	
56	cmc	Cls	3	1473	9		206	Parkinson_Telemonitor	Reg	–	5875	19	
57	colleges	Reg	–	7063	44		207	pc1	Cls	2	1109	21	
58	combined_cycle_plant	Reg	–	9568	4		208	pc3	Cls	2	1563	37	
59	communities_and_crime	Reg	–	1994	102		209	pc4	Cls	2	1458	37	✓
60	company_bankruptcy	Cls	2	6819	95	✓	210	pendigits	Cls	10	10992	16	
61	compass	Cls	2	16644	17		211	Performance-Prediction	Cls	2	1340	19	
62	compressive_strength	Reg	–	1030	8	✓	212	philippine	Cls	2	5832	308	
63	connect-4	Cls	3	67557	42		213	PhishingWebsites	Cls	2	11055	30	✓
64	Contaminant-10.0GHz	Cls	2	2400	30		214	phoneme	Cls	2	5404	5	
65	Contaminant-10.5GHz	Cls	2	2400	30		215	Physicochemical_r	Reg	–	45730	9	✓
66	Contaminant-11.0GHz	Cls	2	2400	30		216	PieChart3	Cls	2	1077	37	
67	Contaminant-9.0GHz	Cls	2	2400	30		217	Pima_Indians_Diabetes	Cls	2	768	8	
68	Contaminant-9.5GHz	Cls	2	2400	30		218	PizzaCutter3	Cls	2	1043	37	
69	contraceptive_method	Cls	3	1473	9		219	pol	Cls	2	10082	26	
70	CookbookReviews	Reg	–	18182	7		220	pol_reg	Reg	–	15000	48	
71	CPMP-2015-regression	Reg	–	2108	25		221	pole	Reg	–	14998	26	✓
72	CPS1988	Reg	–	28155	6		222	predict_students_dropout	Cls	3	4424	34	
73	cpu_act	Reg	–	8192	21	✓	223	puma32H	Reg	–	8192	32	
74	cpu_small	Reg	–	8192	12		224	puma8NH	Reg	–	8192	8	
75	credit	Cls	2	16714	10		225	Pumpkin_Seeds	Cls	2	2500	12	
76	credit-g	Cls	2	1000	20		226	qsar_aquatic_toxicity	Reg	–	546	8	✓
77	Credit_c	Cls	3	100000	22		227	QSAR_biodegradation	Cls	2	1054	41	

78	credit_card_defaults	Cls	2	30000	23		228	qsar_fish_toxicity	Reg	–	908	6		
79	Customer_Personality	Cls	2	2240	24		229	Rain_in_Australia	Cls	3	145460	18		
80	dabetes_us_hospitals	Cls	2	101766	20		230	Retinopathy_Debrecen	Cls	2	1151	19		
81	Data_Science_Salaries	Reg	–	3755	5		231	rice_cammeo_and_osmancik	Cls	2	3810	7	✓	
82	dataset_sales	Reg	–	10738	10		232	ringnorm	Cls	2	7400	20		
83	debutanizer	Reg	–	2394	7		233	rl	Cls	2	4970	12		
84	delta_ailerons	Cls	2	7129	5		234	RSSI_Estimation	Reg	–	5760	6	✓	
85	delta_elevators	Reg	–	9517	6		235	RSSI_Estimation_1	Reg	–	14400	12		
86	Diamonds	Reg	–	53940	9		236	SAT11-HAND-regression	Reg	–	4440	116		
87	dis	Cls	2	3772	29		237	satellite_image	Reg	–	6435	36		
88	dna	Cls	3	3186	180		238	satimage	Cls	6	6430	36		
89	drug_consumption	Cls	7	1884	12		239	SDSS17	Cls	3	100000	12		
90	dry_bean_dataset	Cls	7	13611	16		240	segment	Cls	7	2310	17		
91	E-CommereShippingData	Cls	2	10999	10		241	seismic+bumps	Cls	2	2584	18		
92	eeg-eye-state	Cls	2	14980	14		242	semeion	Cls	10	1593	256		
93	electricity	Cls	2	45312	8		243	sensory	Reg	–	576	11		
94	elevators	Reg	–	16599	18		244	shill-bidding	Cls	2	6321	3	✓	
95	Employee	Cls	2	4653	8		245	Shipping	Cls	2	10999	9	✓	
96	estimation_of_obesity	Cls	7	2111	16		246	Shop_Customer_Data	Reg	–	2000	6		
97	eucalyptus	Cls	5	736	19	✓	247	shrutime	Cls	2	10000	10		
98	eye_movements	Cls	3	10936	27		248	shuttle	Cls	7	58000	9		
99	eye_movements_bin	Cls	2	7608	20		249	socmob	Reg	–	1156	5		
100	Facebook_Comment_Volume	Reg	–	40949	53		250	space_ga	Reg	–	3107	6		
101	Fiat	Reg	–	1538	6		251	spambase	Cls	2	4601	57		
102	FICO-HELOC-cleaned	Cls	2	9871	23		252	splice	Cls	3	3190	60		
103	fifa	Reg	–	18063	5	✓	253	sports_articles	Cls	2	1000	59		
104	Firm-Teacher-Direction	Cls	4	10800	16		254	statlog	Cls	2	1000	20	✓	
105	first-order-theorem	Cls	6	6118	51		255	steel_industry_energy	Reg	–	35040	10		
106	Fitness_Club_c	Cls	2	1500	6		256	steel_plates_faults	Cls	7	1941	27		
107	Food_Delivery_Time	Reg	–	45593	8		257	stock	Reg	–	950	9		
108	FOREX_audcad-day-High	Cls	2	1834	10		258	stock_fardamento02	Reg	–	6277	6		
109	FOREX_audcad-hour-High	Cls	2	43825	10		259	sulfur	Reg	–	10081	6		
110	FOREX_audchf-day-High	Cls	2	1833	10		260	Superconductivty	Reg	–	21197	81		
111	FOREX_audjpy-day-High	Cls	2	1832	10		261	svmguide3	Cls	2	1243	22		
112	FOREX_audjpy-hour-High	Cls	2	43825	10		262	sylvine	Cls	2	5124	20		
113	FOREX_audsgd-hour-High	Cls	2	43825	10		263	taiwanese_bankruptcy	Cls	2	6819	95		
114	FOREX_audusd-hour-High	Cls	2	43825	10		264	telco-customer-churn	Cls	2	7043	18		
115	FOREX_cadjpy-day-High	Cls	2	1834	10		265	Telecom_Churn_Dataset	Cls	2	3333	17		
116	FOREX_cadjpy-hour-High	Cls	2	43825	10		266	texture	Cls	11	5500	40		
117	fried	Reg	–	40768	10		267	thyroid	Cls	3	7200	21	✓	
118	GAMETES_Epistasis	Cls	2	1600	20		268	thyroid-ann	Cls	3	3772	21		
119	GAMETES_Heterogeneity	Cls	2	1600	20		269	thyroid-dis	Cls	5	2800	26		
120	garments_productivity	Reg	–	1197	13		270	topo_2_1	Reg	–	8885	266		
121	gas-drift	Cls	6	13910	128		271	treasury	Reg	–	1049	15		
122	gas_turbine_emission	Reg	–	36733	10		272	turiye_student	Cls	5	5820	32		
123	Gender_Gap_in_Spanish	Cls	3	4746	13	✓	273	twonorm	Cls	2	7400	20		
124	Gesture_Phase_Segment	Cls	5	9873	32		274	UJI_Pen_Characters	Cls	35	1364	80		
125	golf_play_extended	Cls	2	1095	9		275	us_crime	Reg	–	1994	126		
126	Goodreads-Computer-Book	Reg	–	1234	5	✓	276	vehicle	Cls	4	846	18		
127	healthcare_expenses	Reg	–	1338	6		277	volume	Reg	–	50993	53		
128	Heart-Disease-Dataset	Cls	2	1190	11		278	VulNoneVul	Cls	2	5692	16		
129	helena	Cls	100	65196	27		279	walking-activity	Cls	22	149332	4		
130	heloc	Cls	2	10000	22		280	wall-robot-navigation	Cls	4	5456	24		
131	hill-valley	Cls	2	1212	100	✓	281	Water_Potability	Cls	2	3276	8		
132	house_16H	Cls	2	13488	16	✓	282	water_quality	Cls	2	7996	20		
133	house_16H_reg	Reg	–	22784	16		283	Waterstress	Cls	2	1188	22		
134	house_8L	Reg	–	22784	8	✓	284	Wave_Energy_Perth_100	Reg	–	7277	201		
135	house_prices_nominal	Reg	–	1460	79		285	Wave_Energy_Sydney_100	Reg	–	2318	201		
136	house_sales_reduced	Reg	–	21613	18		286	Wave_Energy_Sydney_49	Reg	–	17964	99		
137	houses	Reg	–	20640	8		287	waveform-v1	Cls	3	5000	21	✓	
138	housing_price_prediction	Reg	–	545	12		288	waveform-v2	Cls	3	5000	40		
139	HR_Analytics	Cls	2	19158	13		289	weather_izmir	Reg	–	1461	9		
140	htru	Cls	2	17898	8		290	website_phishing	Cls	3	1353	9		
141	ibm-employee-performance	Cls	2	1470	30	✓	291	Wilt	Cls	2	4821	5		
142	IEEE80211aa-GATS	Reg	–	4046	27	✓	292	wind	Reg	–	6574	14		
143	in_vehicle_coupon	Cls	2	12684	21		293	wine	Cls	2	2554	4		
144	Indian_pines	Cls	8	9144	220		294	wine+quality	Reg	–	6497	11		
145	INNHotelsGroup	Cls	2	36275	17	✓	295	wine-quality-red	Cls	6	1599	4	✓	
146	Insurance	Cls	2	23548	10		296	wine-quality-white	Cls	7	4898	11		
147	internet_firewall	Cls	4	65532	7	✓	297	Wine_Quality_red	Reg	–	1599	11		
148	internet_usage	Cls	46	10108	70		298	Wine_Quality_white	Reg	–	4898	11		
149	Intersectional-Bias	Cls	2	11000	19		299	yeast	Cls	10	1484	8		
150	Is-this-a-good-customer	Cls	2	1723	13		300	yprop_4_1	Reg	–	8885	251		

TALENT-Extension (high-dimensional)

ID	Name (Source)	Task	Class	Sample	Feature	ID	Name (Source)	Task	Class	Sample	Feature
1	ALLAML	Cls	2	72	7129	10	lung	Cls	5	203	3312
2	arcene	Cls	2	200	10000	11	orlraws10P	Cls	10	100	10304
3	BASEHOCK	Cls	2	1993	4862	12	PCMAC	Cls	2	1943	3289
4	CLL_SUB_111	Cls	3	111	11340	13	Prostate_GE	Cls	2	102	5966
5	colon	Cls	2	62	2000	14	RELATHE	Cls	2	1427	4322
6	gisette	Cls	2	7000	5000	15	SMK_CAN_187	Cls	2	187	19993
7	GLI_85	Cls	2	85	22283	16	TOX_171	Cls	4	171	5748
8	GLIOMA	Cls	4	50	4434	17	warpAR10P	Cls	10	130	2400
9	leukemia	Cls	2	72	7070	18	warpPIE10P	Cls	10	210	2420

TALENT-Extension (many-class)										
ID	Name (Source)	Task	Class	Sample	Feature	ID	Name (Source)	Task	Class	Sample	Feature
1	aloi	Cls	1000	108000	128	4	dionis	Cls	355	416188	60
2	BachChoralHarmony	Cls	102	5665	15	5	MD_MIX_Mini_Copy	Cls	706	28240	31
3	beer_reviews	Cls	104	1586614	12	6	seattlecrime6	Cls	135	523577	7

TALENT-Extension (very-large-scale)										
ID	Name (Source)	Task	Class	Sample	Feature	ID	Name (Source)	Task	Class	Sample	Feature
1	Airlines_DepDelay_10M	Reg	–	10000000	9	12	jannis	Cls	4	83733	54
2	blogfeedback	Reg	–	60021	276	13	KDDCup99	Cls	23	4898431	41
3	BNG(credit-a)	Cls	2	1000000	15	14	microsoft	Reg	–	1200192	136
4	CDC_Diabetes_Health	Cls	2	253680	21	15	nomao	Cls	2	34465	118
5	covertype	Cls	7	581012	54	16	poker-hand	Cls	10	1025009	10
6	Data_Science_Good_Kiva	Cls	4	671205	11	17	sf-police-incidents	Cls	2	2215023	8
7	dilbert	Cls	5	10000	2000	18	Smoking_and_Drinking	Cls	2	991346	23
8	fabert	Cls	7	8237	800	19	UJIndoorLoc	Reg	–	21048	520
9	Fashion-MNIST	Cls	10	70000	784	20	volkert	Cls	10	58310	180
10	gina_agnostic	Cls	2	3468	970	21	Wave_Energy_Perth_49	Reg	–	36043	99
11	Higgs	Cls	2	1000000	28	22	yahoo	Reg	–	709877	699

4.3 Dataset Splits and Evaluation Criteria

Implementation details. We evaluate all methods described in subsection 3.2. Because tabular models are sensitive to hyperparameters, we adopt a uniform tuning protocol to ensure fairness. Full K-fold cross-validation would be robust but computationally prohibitive at our scale; applying CV only to small datasets would introduce inconsistent selection pressure due to arbitrary size thresholds. We therefore use a single, fixed hold-out protocol for the main benchmark and study CV+ensembling separately on TALENT-tiny (see section 7).

Data splits and tuning. Each dataset is randomly split into train/val/test with proportions 64%/16%/20% following the setup in (Gorishniy et al., 2021, 2024). Hyperparameters are selected on the validation split, and early stopping is triggered by the task metric on validation (accuracy for classification; RMSE for regression). We use Optuna (Akiba et al., 2019) with a fixed budget of 100 trials per method–dataset pair. After selecting the best configuration, we retrain and evaluate each model with 15 random seeds and report the mean across seeds. In section 7 we compare this protocol to CV+ensembles and show that, while CV can improve absolute scores, the relative ordering of methods remains largely unchanged.

Preprocessing. We follow the pipeline of (Gorishniy et al., 2021). Numerical features are imputed by column means and standardized (zero mean, unit variance). Categorical features are ordinally encoded; missing categories are mapped to a dedicated token "−1". For non-deep methods (except CatBoost) and for deep methods without an explicit categorical module, we apply one-hot encoding after the ordinal step.

Method-specific settings. For gradient boosting, we explicitly pass feature types to CatBoost (native categorical handling). For all deep methods, we use AdamW (Loshchilov and Hutter, 2019) and a batch size of 1024 unless noted otherwise. Pretrained tabular models are evaluated from their latest public checkpoints with default inference hyperparameters (i.e., no per-dataset tuning). The complete search spaces and per-method training options are available at `https://github.com/LAMDA-Tabular/TALENT/tree/main/TALENT/configs`.

Evaluation criteria. For classification tasks, we evaluate models using accuracy (higher is better) as the primary metric and use Root Mean Square Error (RMSE, lower is better) for regression tasks to select the best-performing model during training on the validation set. Additionally, for classification tasks, we record F1 and AUC scores, which are especially valuable for imbalanced datasets. For regression tasks, we also compute MAE and R^2 to provide complementary evaluations of test set performance.

To aggregate per-dataset performance and provide a holistic evaluation across all datasets, we adopt several criteria:

- **Average Rank.** Following Delgado et al. (2014); McElfresh et al. (2023), we report the average performance rank across all methods and datasets (lower is better).
- **Statistical Comparison.** To assess significant differences between methods, we plot critical difference diagrams via Wilcoxon-Holm test (Demsar, 2006; McElfresh et al., 2023) and paired t-test heatmaps to illustrate statistical comparisons.
- **Average relative improvement.** Following Gorishniy et al. (2025), we calculate the relative improvement of a tabular method w.r.t. the performance of a tabular baseline, *e.g.*, MLP, and report the average value across all methods and datasets (higher is better).
- **Aggregated Performance.** We aggregate per-dataset results using the Shifted Geometric Mean (SGM) (Holzmüller et al., 2024). For classification tasks, we use classification error (1 – accuracy), and for regression tasks, we use normalized RMSE (nRMSE).
- **PAMA (Probability of Achieving the Best Accuracy).** The fraction of datasets on which a method attains the best performance among all contenders (Delgado et al., 2014). Although originally proposed for classification, we extend it to regression by defining "best" as the lowest error (*e.g.*, RMSE), and retain the name for consistency.

These evaluation metrics ensure both robust performance aggregation and statistically sound comparisons across diverse tabular datasets.

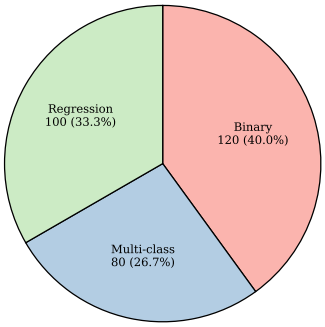
4.4 Advantages of Our Benchmark

Our benchmark is designed to evaluate tabular methods under broad, realistic coverage and to surface behaviors that previous studies may miss. Figure 2 summarizes the key properties of Talent; together they highlight three advantages: substantially broader task/domain/feature coverage, a more balanced size distribution, and explicit attention to real-world difficulty factors (imbalance and class cardinality).

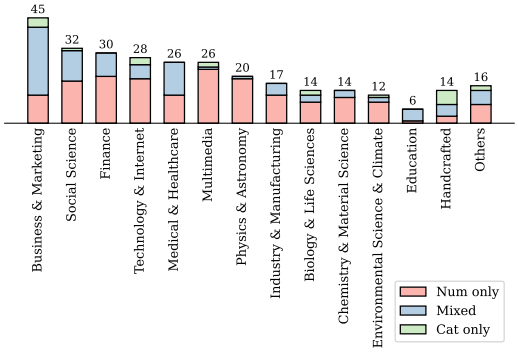
Coverage of tasks. Unlike benchmarks that focus only on classification (McElfresh et al., 2023), Talent spans *all three* standard tabular settings: 120 binary, 80 multi-class, and 100 regression datasets (Figure 2a). This breadth is important because many modern tabular models are intended to handle both classification and regression with a single design.

Coverage of domains & feature types. We curate datasets from 13 application areas—including business & marketing, social science, finance, technology & internet, medical & healthcare, multimedia, physics & astronomy, industry & manufacturing, biology & life sciences, chemistry & materials, environmental science & climate, education, and handcrafted. This diversity enables us to assess whether tabular methods can generalize across applications from varied fields. Within each domain, we retain mixtures of feature types (numeric only, categorical only, and mixed), as shown by the stacked bars in Figure 2b. This combination in our benchmark stresses models to cope with heterogeneous attributes rather than a single, homogeneous regime.

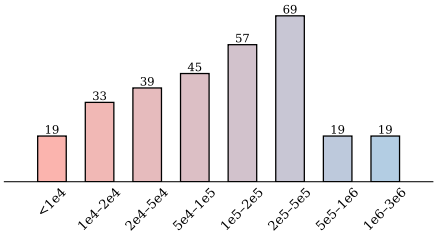
Coverage of data sizes. To avoid size-driven artifacts, we target a *more uniform* spread over dataset complexity, measured by $N \times d$ (instances $\times$ features). Figure 2c shows that Talent allocates substantial mass to small, medium, and moderately large problems alike, yielding fairer aggregate conclusions than skewed size distributions.



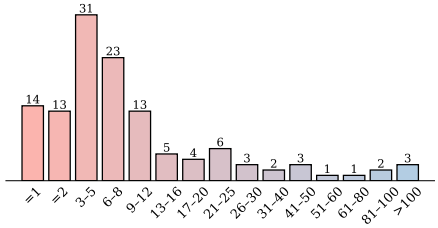
(a) Statistics over task types



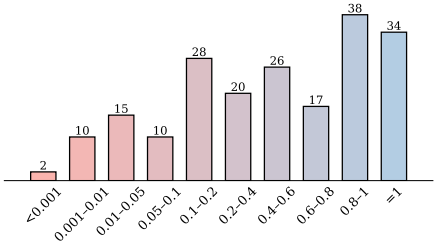
(b) Statistics over domains



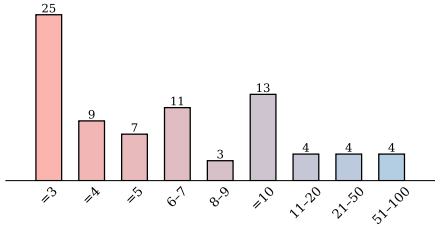
(c) Statistics over data sizes



(d) Statistics of the number of categorical features



(e) Statistics over imbalance rate



(f) Statistics of classes in multi-class tasks

Figure 2: Advantages of the proposed benchmark. (a) shows the number of datasets for three tabular prediction tasks. (b) shows the histogram of datasets across various domains, as well as the types of attributes. (c) shows the number of datasets along with the change of their sizes ($N \times d$). (d) shows the histogram of the number of categorical features in datasets with categorical features. (e) shows the histogram of the imbalance rate for classification datasets. (f) shows the histogram of the number of classes for multi-class classification datasets.

Coverage of categorical structure. Because categorical attributes are central to tabular heterogeneity, we report both the number of categorical features per dataset (Figure 2d) and the number of classes in multi-class tasks (Figure 2f). The resulting distributions are intentionally broad—spanning datasets with very few to many categorical columns, and tasks with small to large class cardinalities. This diversity ensures that evaluations meaningfully stress models' ability to handle categorical encodings, tokenization or embedding strategies, and class-aware objectives across a wide spectrum of practical scenarios.

Coverage of imbalance. Real deployments often face skewed label distributions. Figure 2e shows the imbalance-ratio histogram across our classification sets: while many datasets are near-balanced, a substantial tail is moderately to strongly imbalanced. We evaluate with vanilla training (no task-specific rebalancing), and additionally report F1/AUC to ensure fair comparison on skewed test sets.

Summary. Compared with prior benchmarks, TALENT offers (i) richer task coverage (including regression), (ii) multi-domain, mixed-feature datasets within each domain, (iii) a more even spread over $N \times d$, and (iv) explicit variation in categorical count, class cardinality, and imbalance. This breadth and balance make TALENT a stronger proxy for real-world tabular challenges and a more reliable basis for comparing deep and tree-based methods.

5 Comparison Results among Datasets

We compare tabular methods across 300 datasets using multiple evaluation criteria. To provide a concise and clear analysis, we report only the aggregated performance metrics, such as average rank, across the entire dataset collection. Detailed per-dataset results are available in the online supplementary document at `https://github.com/LAMDA-Tabular/TALENT/tree/main/results`.

In the figures presented in this section, we use distinct colors to represent different categories of methods, ensuring clarity and ease of comparison. Specifically, **Dummy** is represented by gray, while **classical methods** are denoted by coral orange. **Tree-based methods** are visualized using vibrant green, and **MLP variants** are shown in rich red. For methods with **specially designed architectures**, we use soft indigo, while **tree-mimic methods** are represented by emerald teal. **Neighborhood-based methods** are depicted in vivid purple, **token-based methods** in bright cyan, **regularization-based methods** in fresh lime, and **pretrained foundation models** in warm amber.

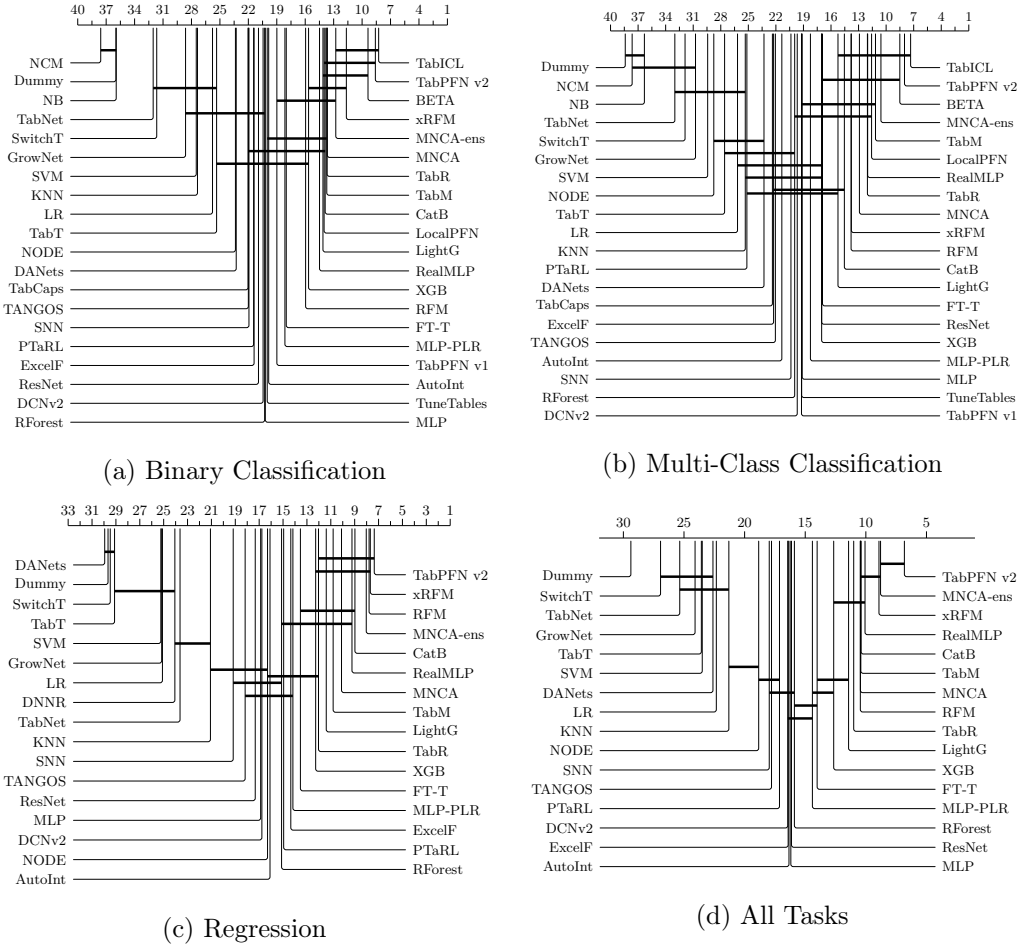
It is notable that some methods are limited to specific types of tasks. For example, TabPFN v1 and TabICL are designed exclusively for classification tasks and cannot handle regression, while DNNR is tailored for regression and cannot be applied to classification. When showing the performance over all datasets, we only present the results of methods capable of addressing both classification and regression tasks.

5.1 On Average Performance

We compare 40 representative tabular methods across 300 datasets, reporting average performance ranks and conducting statistical significance tests with the Wilcoxon–Holm procedure at a 0.05 level (Demsar, 2006). The critical difference diagrams are shown in Figure 3, while detailed rank values and pairwise heatmaps are deferred to the appendix. Figure 1 further contextualizes representative methods in terms of efficiency and model size.[3]

The most striking results come from *pretrained foundation models*. Across different task types, TabPFN v2 and TabICL consistently rank among the best-performing methods. In many cases, they significantly outperform classical ensembles and tuned deep models, highlighting the benefits of pretraining and in-context learning for tabular data. While the

3. The average ranks in Figure 1 are computed on a representative subset and may differ slightly from the full-rank results.



(a) Binary Classification

(b) Multi-Class Classification

(c) Regression

(d) All Tasks

Figure 3: Critical difference of all methods via the Wilcoxon-Holm test with a significance level of 0.05. The lower the rank value, the better the performance.

earlier TabPFN v1 lags behind, its successors—especially TabPFN v2—extend to regression tasks and display clear generalization advantages.

Beyond foundation models, several consistent patterns emerge. Tree-based ensembles remain highly competitive: Random Forest and XGBoost provide reliable baselines, while CatBoost and LightGBM often achieve top-tier ranks, especially in regression tasks. The Wilcoxon–Holm analysis shows no significant differences among these gradient boosting methods, underscoring their maturity and robustness. Recursive Feature Machines (RFM) and its extension xRFM also achieve performance close to the strongest ensembles, occupying similar rank intervals in both binary and regression settings. The results indicate that tree-like structures may form an effective hybrid paradigm.

For deep methods, vanilla MLPs are generally weak, but tuned implementations such as MLP-PLR and RealMLP close the gap substantially. RealMLP, in particular, achieves competitive performance across tasks and is statistically stronger than many ResNet-style or regularization-based variants. Token-based approaches (*e.g.*, FT-T, ExcelFormer, AutoInt)

achieve robust results, especially in classification, but significance tests show they often cluster with ensembles in the same equivalence group, indicating that attention-based tokenization provides stability but not decisive superiority.

Tree-mimic networks such as NODE and TabNet generally underperform relative to ensembles. In contrast, neighborhood-based models such as ModernNCA achieve excellent results and are often statistically comparable to CatBoost and LightGBM, highlighting the promise of retrieval-based learning. Interestingly, *ensembling within deep methods* further improves performance—TabM outperforms base MLPs, and MNCA-ens consistently surpasses ModernNCA—indicating that ensemble effects remain beneficial even for neural models.

Despite these advances, the Wilcoxon–Holm tests show that foundation models, ensembles, and top DNNs (RealMLP, ModernNCA) often remain statistically tied, suggesting that universal superiority has not yet been achieved. Scalability and computational cost also remain open challenges for foundation models. Results from BETA further indicate that fine-tuning strategies (*e.g.*, task-specific adaptation of pretrained TabPFN) can yield improvements, suggesting a promising direction for enhancing current foundation models.

Overall, these results yield several key observations:

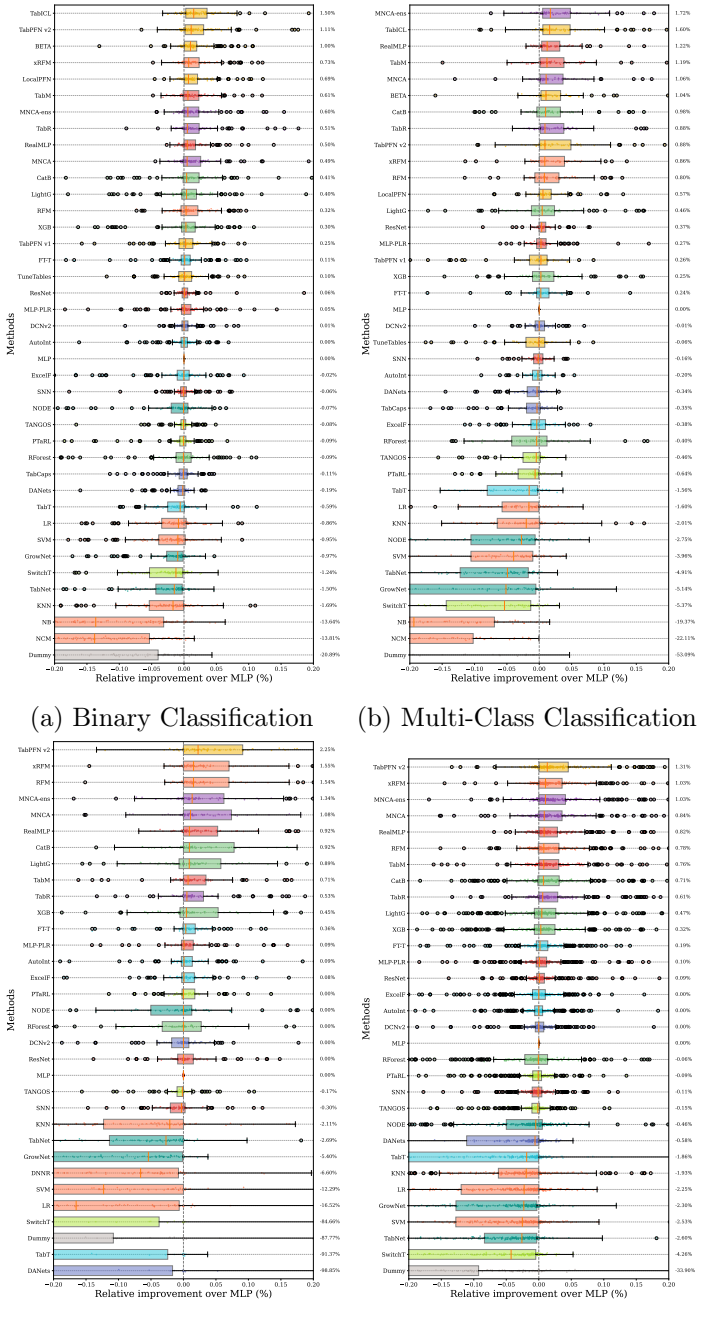
- Pretrained foundation models (TabPFN v2, TabICL) deliver state-of-the-art performance across many datasets and task types, substantially advancing over earlier versions. The results of foundation models significantly narrow—but not entirely close—the gap between tree-based and DNN-based paradigms.
- Tree-based ensembles (CatBoost, LightGBM, XGBoost) remain strong, reliable, and statistically robust baselines.
- Carefully optimized DNNs, especially RealMLP and ModernNCA, can rival or surpass ensembles, showing robustness across both classification and regression tasks.
- Token-based transformers (FT-T, ExcelFormer, AutoInt) provide stable and competitive results, but their advantages are not statistically decisive over ensembles.
- Ensemble-style strategies (*e.g.*, TabM, MNCA-ens) demonstrate consistent gains over their base variants, suggesting that ensembling remains an effective principle even in modern deep tabular learning.
- The Wilcoxon–Holm tests highlight large equivalence groups: many methods, while different in design, are statistically indistinguishable. This indicates that progress often comes from incremental but robust improvements, rather than single universally dominant architectures, reflecting the growing maturity of the tabular learning ecosystem.

5.2 Relative Improvements over Tabular Baselines

Well-tuned MLP is widely regarded as a strong baseline for tabular prediction tasks. To assess robustness, we evaluate each method by its relative improvement over MLP following Gorishniy et al. (2025). Formally, for a method m on dataset d, the relative improvement is defined as

$$\Delta R_{m,d} = \frac{R_{m,d} - R_{\mathrm{MLP},d}}{R_{\mathrm{MLP},d}},$$

where R represents accuracy for classification and the min–max scaled negative RMSE for regression. Box plots in Figure 4 summarize improvements across tasks, with medians reflecting typical gains and interquartile ranges (IQRs) indicating stability.



(a) Binary Classification (b) Multi-Class Classification

(c) Regression (d) All Tasks

Figure 4: The Box-Plot of relative performance improvements of tabular methods over the MLP baseline across binary classification, multi-class classification, and regression tasks. The relative improvement is calculated for each dataset, where larger values indicate stronger performance relative to the MLP baseline. The box plots show the median, interquartile range (IQR), and outliers for each method. Methods with narrower IQRs demonstrate greater stability, while wider distributions suggest variability in performance.

Overall, the results confirm earlier statistical findings while highlighting additional nuances. Tree ensembles (CatBoost, LightGBM, XGBoost) not only achieve strong median gains over MLP but also exhibit narrow IQRs, underscoring their stability. Among deep methods, RealMLP, TabR, and ModernNCA consistently improve upon MLP, with ModernNCA showing particularly high and robust gains in regression. In contrast, models such as SwitchTab, GrowNet, and TabNet show wide distributions with negative medians in some settings, reflecting instability and lack of robustness.

Pretrained foundation models (TabPFN v2, TabICL) outperform MLP across nearly all datasets, though their relative margins are often modest, suggesting broad consistency rather than large per-dataset gains. Ensemble-enhanced approaches (*e.g.*, MNCA-ens, TabM) also reliably surpass MLP, confirming that ensembling deep methods improves stability. Variants such as MLP-PLR provide small but systematic improvements over the vanilla MLP, validating the importance of encoding refinements.

Across tasks, binary classification shows the most consistent gains for ensembles and tuned DNNs, while multi-class classification is more challenging, with greater variance and some ensembles underperforming MLP on subsets of datasets. Regression highlights the relative strength of retrieval-based methods (ModernNCA, TabR) and pretrained models, which achieve both higher medians and broader coverage of positive gains.

In summary, outperforming a strong MLP baseline remains non-trivial. Only a subset of methods—gradient boosting ensembles, carefully tuned MLP variants, retrieval-based methods, and pretrained foundation models—achieve consistent and stable improvements, validating them as reliable baselines for future research.

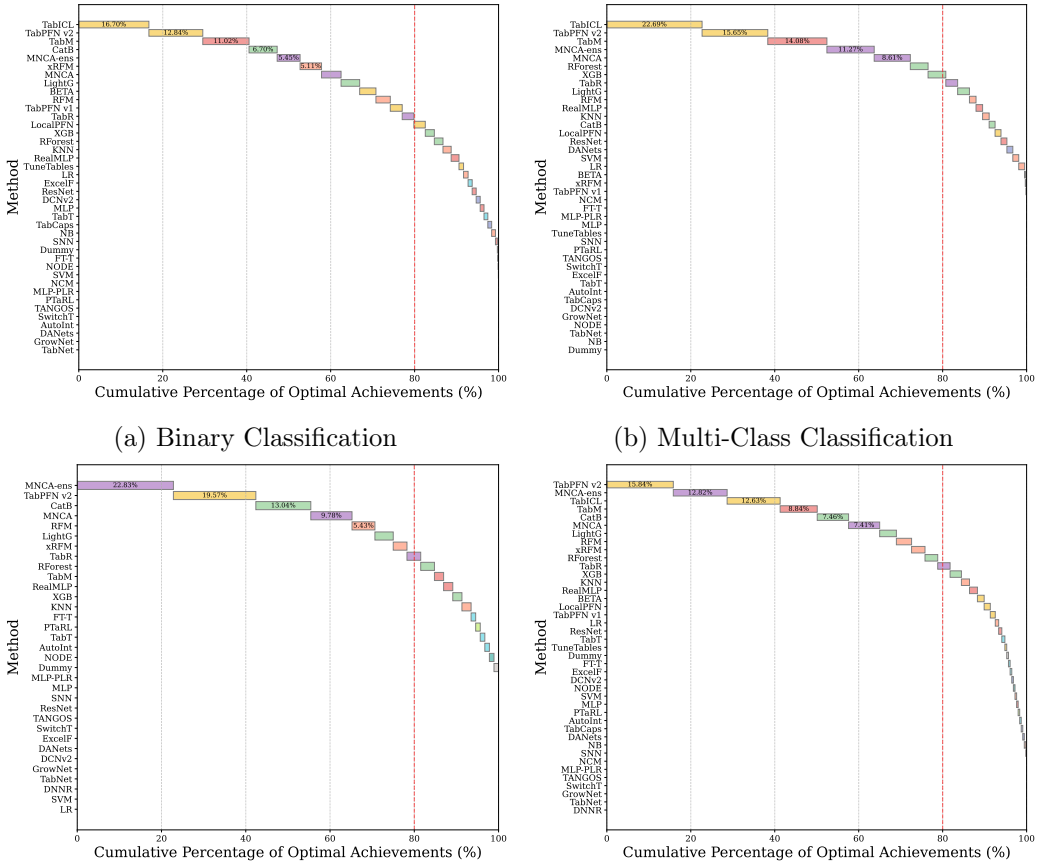
5.3 Probability of Achieving the Best Accuracy

Since the performance of tabular methods varies across datasets, average ranks and statistical tests may obscure methods that excel in specific scenarios. To complement these aggregate metrics, we evaluate the Probability of Achieving the Best Accuracy (PAMA) (Delgado et al., 2014), which measures the proportion of datasets on which a method achieves the best performance. This perspective highlights dataset-specific adaptability and identifies methods that frequently dominate.

The results in Figure 5 reveal several key findings. First, pretrained foundation models show a decisive advantage. TabICL and TabPFN v2 achieve the highest PAMA scores across tasks, winning on a substantial portion of datasets (up to 22.7% in multi-class classification). Their strong performance underscores the value of pretraining and in-context learning for diverse tabular problems. Importantly, their success is not limited to classification: TabPFN v2 ranks second overall in regression tasks, further demonstrating its generality.

Second, classical ensembles remain highly competitive. CatBoost, LightGBM, and XGBoost frequently appear among the top methods, particularly in regression, where CatBoost and LightGBM achieve some of the highest PAMA scores. Extensions such as RFM/xRFM also perform strongly, often statistically indistinguishable from the top ensembles and pretrained models. These results reinforce earlier findings that tree-based ensembles remain robust baselines, with strong adaptability across heterogeneous datasets.

Third, deep methods vary in their adaptability. ModernNCA and its ensemble variant (MNCA-ens) frequently rank near the top across all tasks, with MNCA-ens achieving the



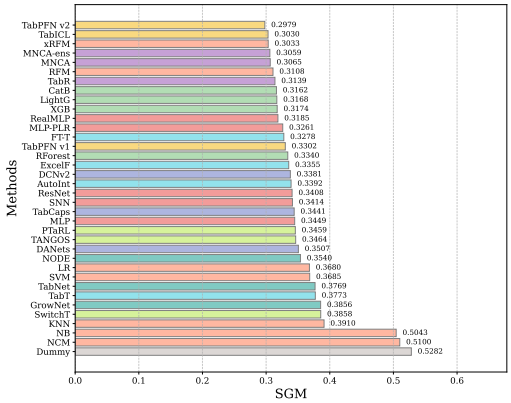
(a) Binary Classification (b) Multi-Class Classification

(c) Regression (d) All Tasks

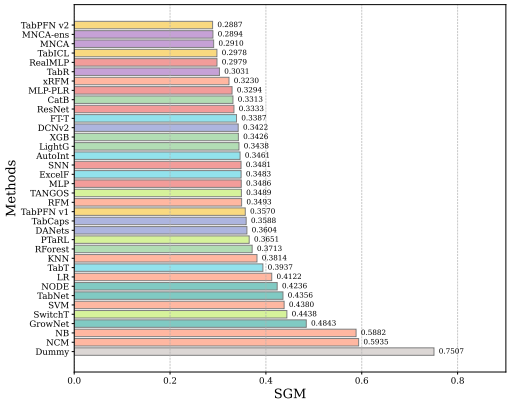
Figure 5: PAMA (Probability of Achieving the Best Accuracy) of various methods in binary classification (a), multi-class classification (b), regression (c), and all tasks (d). Each bar segment denotes a tabular method, whose width is the percentage that the method achieves the best performance over a kind of tabular prediction task. The wider the cell, the more often that a method performs well on the tabular prediction task.

highest PAMA score (22.8%) in regression. This confirms the strength of neighborhood-based retrieval strategies, particularly for numerical prediction. In contrast, early tree-mimic architectures such as NODE and TabNet rarely achieve top ranks, echoing the earlier statistical test results. Among MLP-based methods, RealMLP consistently achieves non-trivial PAMA scores, outperforming most other DNN variants and demonstrating that careful tuning can elevate simple architectures.

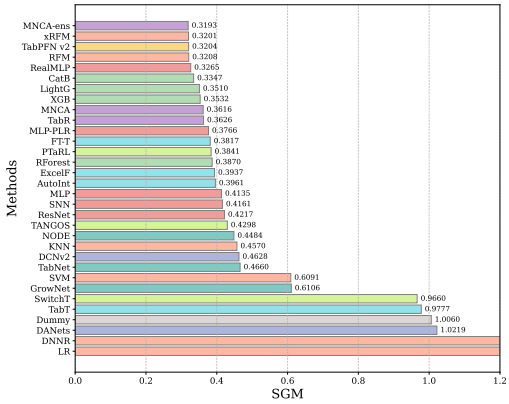
Finally, the PAMA distributions highlight the concentration of top-performing methods. Across binary, multi-class, and regression tasks, fewer than 10 methods account for over 80% of all best-performing cases (as marked by the dashed red line). This identifies a practical "shortlist" of strong candidates—primarily TabICL, TabPFN v2, MNCA-ens, ModernNCA, CatBoost, LightGBM, and TabM—that dominate across most scenarios. Simpler models



(a) Binary Classification



(b) Multi-Class Classification



(c) Regression

Figure 6: Aggregated performance across datasets using Shifted Geometric Mean Error (SGM). Per-dataset metrics are classification error (1–accuracy) for classification tasks and normalized RMSE (nRMSE) for regression tasks. Lower values indicate better performance and higher robustness across datasets and random seeds.

(*e.g.*, Logistic Regression, KNN) occasionally achieve best results in niche datasets, but their contributions are relatively small and task-specific.

In summary, PAMA provides a complementary view to average rank and statistical tests. While many methods perform competitively on average, only a small subset consistently wins across diverse datasets. Pretrained foundation models clearly lead, followed by strong ensembles and retrieval-based methods, suggesting that future research should focus on enhancing adaptability while preserving efficiency.

5.4 Averaged Performance

To evaluate robustness across datasets and random seeds, we report aggregated metrics: Shifted Geometric Mean Error (SGM) for classification and normalized RMSE (nRMSE) for regression, following Holzmüller et al. (2024). The results are shown in Figure 6.

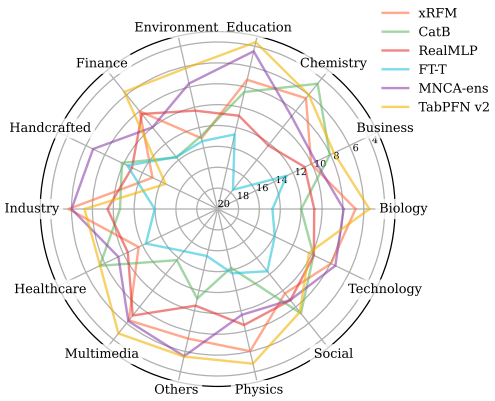


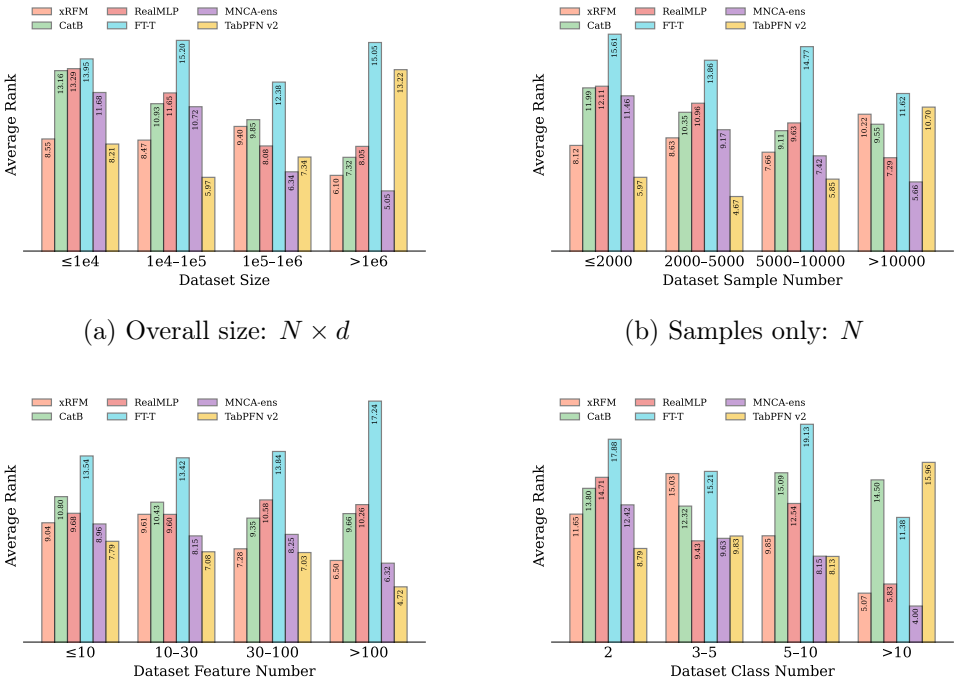
Figure 7: Comparison of representative tabular prediction methods across 14 application domains. The radar plot shows reversed rank scores, where larger values indicate better average performance in a domain.

In binary classification, foundation models (*e.g.*, TabPFN v2, TabICL) achieve the lowest SGM values, closely followed by context-based methods (MNCA-ens, TabR) and strong ensembles (CatBoost, LightGBM). RealMLP and MLP-PLR also perform competitively, further confirming that enhanced MLPs can close much of the historical tree–DNN gap. These results echo earlier average-rank analyses but emphasize that foundation and neighborhood-based models deliver more stable, low-error behavior across seeds.

For multi-class classification, the ranking remains similar, with TabPFN v2 and MNCA-ens again achieving the best SGM, while RealMLP and TabR remain close. Ensembles such as CatBoost and LightGBM still perform strongly but no longer dominate, highlighting the advantage of retrieval-based and pretrained approaches in more complex label structures. Classical methods like LR, KNN, and NB perform poorly under SGM, reinforcing their lack of robustness in high-class scenarios.

Regression shows a slightly different pattern: ensemble-enhanced models (MNCA-ens, xRFM, CatBoost, LightGBM) and RealMLP achieve the lowest nRMSE. TabPFN v2 also ranks among the top performers, suggesting that pretraining contributes to consistent generalization even in continuous targets. In contrast, models such as TabNet, GrowNet, and SwitchTab perform poorly, with high variability across datasets. Linear models (LR) and simple baselines are the weakest, consistent with earlier analyses.

Overall, SGM and nRMSE highlight three key insights. First, pretrained foundation models (TabPFN v2, TabICL) and neighborhood-based ensembles (MNCA-ens, TabR) provide the most stable performance across tasks. Second, gradient boosting ensembles remain reliable, particularly in regression. Third, carefully tuned MLPs (RealMLP, MLP-PLR) consistently rank among the top tier, showing that architectural refinements plus optimization strategies can rival classical ensembles. These findings reinforce conclusions from average-rank and PAMA analyses while underscoring the added stability of foundation and context-based methods under seed-sensitive metrics.



(a) Overall size: $N \times d$

(b) Samples only: N

(c) Features only: d

(d) Number of classes: C

Figure 8: Average ranks of representative tabular methods as dataset characteristics vary. (a) summarizes trends with joint scale $N \times d$; (b) and (c) isolate the marginal effects of sample size N and feature dimensionality d; (d) varies the number of classes C, reflecting the label granularity.

5.5 Results across Domains and Dataset Sizes

To better understand the behavior of tabular methods, we analyze their performance across 14 application domains and four dataset size groups. We select six representative models from diverse categories: the tree-based model CatBoost, the token-based model FT-T, the neighborhood-based model MNCA-ens, the pretrained foundation model TabPFN v2, the enhanced MLP variant RealMLP, and the recursive feature model xRFM. Figure 7 presents the reversed average ranks across domains using a spider chart (larger values indicate better performance). Figure 8 presents the average ranks across dataset sizes (and other statistics) with bar charts (lower values indicate better performance).

Domain-level analysis shows clear specialization among methods. Pretrained and neighborhood-based approaches stand out with broad adaptability. TabPFN v2 performs consistently well across many domains, particularly excelling in `education`, `multimedia`, and `social sciences`, reflecting the benefits of pretraining for generalization on heterogeneous tasks. MNCA-ens also achieves strong and steady performance, ranking among the best in `handcrafted`, `environmental`, and `healthcare` datasets, which highlights the robustness of retrieval-based ensemble strategies. Tree-based ensembles continue to serve as highly competitive baselines. CatBoost is especially effective in `chemistry` and `finance`, where

categorical structures and feature interactions dominate, while xRFM shows stable results across a wide spectrum of domains, often close to the strongest pretrained and context-based models. Other deep neural methods demonstrate complementary strengths. RealMLP achieves notable gains in `finance`, `physics`, and `industry`, confirming the effectiveness of enhanced MLP designs in structured domains. FT-T delivers solid adaptability in `technology` and `social` datasets, benefiting from tokenization and attention mechanisms, although its advantage over ensembles is less pronounced.

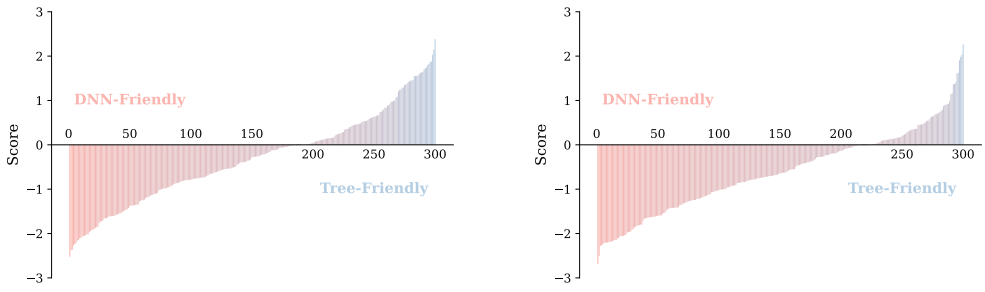
The cross-domain analysis also underscores the variability of model behavior. For instance, in `biology` and `healthcare`, ensemble methods maintain strong performance, while in `multimedia` and `social` science tasks, foundation and retrieval-based models dominate. This variation suggests that domain alignment is a critical factor in achieving optimal results, and no single approach universally leads across all areas.

Dataset-size and composition analysis reveals fine-grained scalability patterns. As shown in Figure 8, model performance varies notably when examined along four complementary dimensions—overall dataset size ($N \times d$), number of samples (N), feature dimensionality (d), and number of classes (C). *All rank values are computed over the full set of evaluated methods in this study, while the figure visualizes only representative models for clarity.*

Across overall dataset scales (Figure 8a), CatBoost continues to exhibit strong scalability, improving steadily as the dataset size increases. In contrast, RealMLP performs best on small-to-medium datasets but declines slightly as scale grows, highlighting optimization and regularization challenges common to MLP-style models. TabPFN v2 ranks near the top on medium-to-large datasets, demonstrating that pretraining confers robust generalization in typical-size regimes, though its effectiveness tapers off when data size becomes very large—an observation consistent with its pretraining limits and context-size constraints. MNCA-ens remains consistently strong, benefiting from ensembling over neighborhood-based embeddings, while xRFM shows competitive performance on small-to-medium scales but struggles a bit when dataset size increases.

When isolating the effect of sample count (Figure 8b)), MNCA-ens, CatBoost, and RealMLP all gain from larger N, confirming their strong scalability under data abundance. TabPFN v2, however, shows its best results around 5k–10k samples before flattening out, suggesting that its pretrained inference window constrains further improvements without architectural extension. FT-Transformer remains stable but shows limited scalability advantage. Conversely, xRFM again performs well in the small-data regime ($N \leq 2000$), consistent with its design as a lightweight, backpropagation-free architecture that benefits from smaller sample sizes.

For feature dimensionality (Figure 8c)), both CatBoost and RealMLP retain strong rankings as d grows, showcasing their robustness to redundant or weakly informative features. Interestingly, TabPFN v2 maintains good performance even when $d > 100$, suggesting that its pretraining includes sufficient diversity to generalize beyond low-dimensional regimes. In contrast, FT-Transformer and xRFM exhibit noticeable degradation as d increases, possibly due to insufficient regularization and the growing difficulty of effective feature selection under very high dimensionality. We will later show in subsection 7.2 that such high-dimensional regimes further amplify these differences, where foundation models in particular face performance degradation in extremely wide feature spaces.



(a) Tree-DNN score w/o TabPFN v2 (b) Tree-DNN score w/ TabPFN v2

Figure 9: Distribution of Tree–DNN scores across 300 datasets. The score is defined as the difference between the best representative tree-based method and the best representative DNN-based method. (a) excludes pretrained foundation models, while (b) includes TabPFN v2 among the DNN-based group. Positive values indicate tree-friendly datasets, negative values indicate DNN-friendly datasets.

When comparing across class cardinalities (Figure 8d)), ensemble-style deep models such as MNCA-ens and RealMLP improve significantly as class numbers increase, demonstrating their flexibility in capturing complex, fine-grained label boundaries. CatBoost remains robust across all C, reinforcing its role as a dependable, well-regularized baseline. FT-Transformer shows declining performance beyond $C > 10$, possibly due to its token-based design being less effective for large label vocabularies. xRFM maintains moderate performance in low-class scenarios but exhibits limited gains for higher class counts.

Summary across domains and scales. Taken together, the domain- and size-wise analyses reveal that model behavior is shaped jointly by the *structural characteristics* and the *scale* of tabular data. These findings collectively suggest that fine-grained analyses along domain and data-size axes offer more informative insights than aggregate rankings alone. They reveal how different modeling principles respond to variations in feature structure, label granularity, and data scale. Such observations motivate a more nuanced understanding of how classical and deep paradigms complement each other in tabular prediction, laying the groundwork for our subsequent discussion.

5.6 Revisiting the Tree-DNN Debate

A longstanding question in tabular learning is whether tree-based ensembles or deep neural networks (DNNs) are inherently stronger. Earlier benchmarks generally favored ensembles such as Random Forest, XGBoost, and CatBoost, while deep models struggled to consistently outperform them. This "tree–DNN divide" motivated much of the early work on specialized architectures for tabular data (Grinsztajn et al., 2022; McElfresh et al., 2023).

To quantify this divide, we adopt the Tree–DNN score (Equation 2), defined as the difference between the best-performing tree-based model and the best-performing DNN-based model after normalization. Higher values indicate datasets where ensembles dominate

(tree-friendly), while lower values indicate datasets where DNNs dominate (DNN-friendly).

$$s = \max(\hat{s}_{\text{XGBoost}}, \hat{s}_{\text{CatBoost}}, \hat{s}_{\text{RForest}}, \hat{s}_{\text{LightG}}) - \max(\hat{s}_{\text{RealMLP}}, \hat{s}_{\text{FT-T}}, \hat{s}_{\text{MNCA}}, \hat{s}_{\text{TabM}}) \ . \quad (2)$$

$\hat{s}$ represents the normalized metric, such as accuracy for classification tasks or negative RMSE for regression tasks.

Figure 9(a) shows the sorted score distribution without including pretrained models (TabPFN v2). A large portion of datasets remains tree-friendly, confirming that ensembles retain a structural advantage on many tasks. However, a comparable fraction of datasets favors DNNs, reflecting the progress of modern deep tabular methods such as RealMLP and ModernNCA.

The picture changes once pretrained tabular foundation models are included (Figure 9(b)). With TabPFN v2 added to the DNN group, the balance shifts substantially toward DNN-friendly datasets. This highlights the transformative role of foundation models: they narrow, and in many cases invert, the traditional advantage of ensembles by leveraging pretraining and in-context inference. Still, the right tail of the distribution shows that there remain numerous datasets where ensembles achieve clear wins, particularly in regression-heavy or highly categorical settings.

Overall, the debate has evolved rather than disappeared. Tree-based ensembles remain reliable, statistically competitive baselines across diverse tasks, especially when data structure aligns with their strengths. Yet, pretrained foundation models represent a paradigm shift: they elevate DNNs to state-of-the-art levels across many benchmarks, reducing the universality of ensemble dominance. The results suggest a more nuanced view—trees still matter, but pretrained models are redefining the frontier of tabular learning.

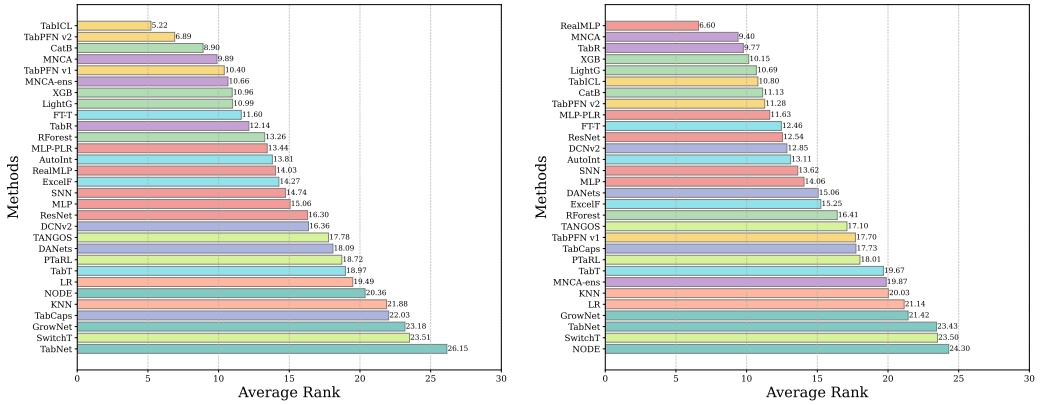
5.7 Comparisons with Imbalance-Sensitive Criteria

Real-world tabular datasets often exhibit class imbalance, making accuracy insufficient as a sole evaluation metric. To complement previous analyses, we assess methods using AUC and F1-score on 67 classification datasets with imbalance rates below 0.25, without applying any additional imbalance-handling strategies. The results are shown in Figure 10.

The rankings reveal both consistency and divergence compared to accuracy-based results. AUC favors ensemble-style approaches and pretrained models: TabICL, TabPFN v2, and CatBoost dominate, with TabICL achieving the lowest average rank. These methods excel at separating minority and majority classes, reflecting their robust decision boundaries under imbalance. ModernNCA and its ensemble variant also remain competitive, consistent with their strong overall performance in earlier benchmarks.

In contrast, F1-score shifts the advantage toward certain DNN-based approaches. RealMLP emerges as the top performer, with ModernNCA, TabR, and CatBoost following closely. These models balance precision and recall more effectively, which is critical when evaluating rare classes. Interestingly, TabICL and TabPFN v2—dominant under AUC—perform slightly less consistently under F1, suggesting that while they are excellent at ranking predictions, they may not optimize precision–recall trade-offs equally well.

Classical ensembles such as Random Forest and XGBoost maintain mid-level performance across both metrics, outperforming most classical baselines but trailing behind modern DNNs and foundation models. Token-based methods like FT-T remain stable, appearing competitive



(a) AUC-based Average Rank (b) F1-score-based Average Rank

Figure 10: Evaluations of tabular methods on 67 imbalanced classification datasets (31 binary and 36 multi-class tasks with imbalance rates below 0.25). The average ranks are computed using AUC (a) and F1-score (b). Lower values indicate better performance.

under both AUC and F1, though rarely leading, which aligns with earlier observations from statistical comparisons.

Overall, imbalance-sensitive evaluations confirm the robustness of strong ensembles (CatBoost, LightGBM), advanced DNNs (RealMLP, ModernNCA), and foundation models (TabICL, TabPFN v2). Yet the divergence between AUC and F1 underscores that model choice should align with task-specific goals: AUC-oriented scenarios benefit from ensembles and pretraining, while F1-sensitive contexts may prefer RealMLP or neighborhood-based designs.

6 Measuring the Heterogeneity of Tabular Data

One of the central challenges in tabular learning arises from the inherent heterogeneity of tabular datasets (Shwartz-Ziv and Armon, 2022). Unlike other modalities such as images or text, where inputs share relatively uniform structures, tabular datasets often combine diverse attribute types–including continuous values, binary indicators, ordinal features, and high-cardinality categorical variables (Borisov et al., 2024). This diversity poses substantial challenges for deep models, which must simultaneously accommodate heterogeneous statistical properties and learn meaningful interactions across them.

In this section, we build on meta-features that capture dataset properties and systematically examine how their variations reflect the heterogeneity of tabular data. To this end, we introduce a performance-curve prediction task, which learns to forecast the training dynamics of a tabular method from both meta-features and early learning signals. A meta-feature is considered effective if it contributes to accurately predicting these dynamics, thereby indicating its role in shaping model behavior.

By analyzing the predictive relationships between meta-features and training dynamics, we identify which dataset characteristics most strongly influence the effectiveness of deep tabular models. Our results provide insights into how heterogeneity can be measured, which

factors are most critical, and how they affect the success or failure of different representative methods. This analysis not only clarifies the limitations of current approaches but also offers guidance for designing more robust and adaptive tabular learning models.

6.1 Reformulating Meta-Feature Selection as a Dynamics Forecasting Task

Meta-features capture intrinsic properties of a tabular dataset. To understand which properties matter most for deep tabular learning, we link them to the epoch-wise training dynamics of neural models. Instead of treating meta-feature selection as an isolated procedure, we reformulate it as a forecasting task: predicting validation curves from dataset properties. This formulation allows us to identify which dataset characteristics most strongly influence model behavior.

Formally, given a training set $\mathcal{D}$, we optimize a deep tabular model f with stochastic gradient descent over Equation 1. Each epoch is a complete pass through $\mathcal{D}$, with mini-batches drawn after random permutation of the examples. Assuming the best hyperparameters of f are predetermined, we record validation statistics (*i.e.*, classification accuracy or normalized RMSE for regression) as a sequence $\boldsymbol{a} = [a_1, a_2, \ldots, a_T] \in \mathbb{R}_+^T$ over T epochs until early stopping.

We propose to forecast $\boldsymbol{a}$ using two signals: (i) dataset meta-features $\boldsymbol{m}_\mathcal{D}$ that encode structural properties (*e.g.*, number of attributes, joint entropy with the target), and (ii) the initial segment of the validation curve. Specifically, we define a support set $\mathcal{S} \in \mathbb{R}_+^K$ containing the first K values of $\boldsymbol{a}$, and a query set $\mathcal{Q} \in \mathbb{R}_+^{T-K}$ with the remaining values. The task is to learn a mapping

$$g : \{\boldsymbol{m}_\mathcal{D}, \mathcal{S}\} \mapsto \mathcal{Q},$$

leveraging both dataset statistics and early validation behavior. By analyzing which meta-features improve forecasting accuracy, we can reveal the key factors shaping the training dynamics of deep tabular methods.

6.2 Selecting Effective Meta-Features for Heterogeneity Analysis

To forecast training dynamics from $\mathcal{S}$, we model the shape of validation curves directly. Let t denote the epoch index and y the performance measure. Because $\mathcal{S}$ is short, we adopt a learnable approach: predicting the parameters of a curve family from $\{\mathcal{S}, \boldsymbol{m}_\mathcal{D}\}$, where meta-features provide auxiliary signals that adapt predictions to dataset-specific properties.

All data for this task are drawn from validation curves of MLPs trained with default hyperparameters on our benchmark datasets. We split these curves into 80% training and 20% testing, ensuring no dataset overlap between the two.

Dynamics Curve Approximation. We model validation dynamics with the following curve family:

$$\boldsymbol{a}_\theta(t) = A \log t + B\sqrt{t} + C + D/t, \tag{3}$$

where t is the epoch number, $\boldsymbol{a}(t)$ is the validation performance, and $\theta = \{A, B, C, D\}$ defines the curve. This functional form is empirically selected to capture the characteristic sub-linear growth and asymptotic convergence observed in tabular deep learning validation curves. For accuracy curves, A and B are typically positive, reflecting monotonic improvement. To capture dataset effects, we learn a meta-mapping $h : \{\boldsymbol{m}_\mathcal{D}, \mathcal{S}\} \mapsto \theta$ (Vinyals et al., 2016;

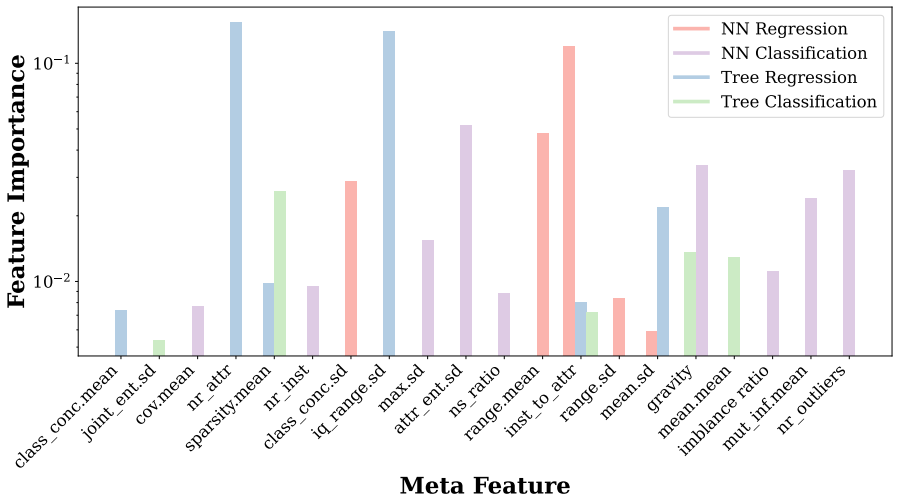


Figure 11: Importance of various meta-features for predicting training dynamics across different types of tabular datasets. Legends such as "NN Regression" denote selected important meta-features for regression tasks on DNN-friendly datasets.

Chao et al., 2020). Comparisons with other formulations of the curve family are discussed in the Appendix.

Learning Objective. We optimize h by minimizing the mean absolute error (MAE) between predicted and observed validation curves:

$$\min_{h} \sum_{\{\boldsymbol{m}_{\mathcal{D}},\mathcal{S}\}} \sum_{a_t \in \mathcal{Q}} \ell\big(\boldsymbol{a}_{\theta=h(\boldsymbol{m}_{\mathcal{D}},\mathcal{S})}(t),\, a_t\big)\,. \tag{4}$$

For each dataset, we collect the first five epochs ($\mathcal{S}$) and meta-features $\boldsymbol{m}_{\mathcal{D}}$ (see Table 5), feed them into h (an MLP), and predict θ. The predicted curve extrapolates $\mathcal{Q}$, and accuracy is assessed by the discrepancy between predictions and ground-truth dynamics. After training, h generalizes to unseen datasets, highlighting the most effective meta-features for forecasting training behavior.

6.3 The Selected Meta-Features

We implement h as a four-layer MLP. The input dimension is 24: five dimensions from the first five epochs of $\mathcal{S}$ and 19 from dataset meta-features (and derived statistics such as `range.mean`, `range.std`, etc.). The output is the four parameters of Equation 3. We evaluate h on classification and regression curves and find that including meta-features substantially improves prediction accuracy. Additional results are reported in the appendix.

We further analyze which meta-features most strongly influence predictions by examining their weights in h. Training datasets are divided into four categories (classification/regression $\times$ tree-/DNN-friendly), where the tree-/DNN-friendly split is determined according to the Tree-DNN score in Figure 9(a). For each category, we train a separate predictor and extract the most important meta-features (see Figure 11). Results reveal that: (1) For classification tasks, the `gravity` meta-feature—measuring the distance between minority and majority

class centers—is critical for both tree-based and DNN-based methods. (2) For regression tasks, `range.mean` (average attribute range) and `mean.mean` (average attribute mean) are highly predictive. (3) Tree-based methods rely heavily on `sparsity.mean`, while DNN-based methods are more sensitive to distributional statistics such as `max.sd`.

These observations suggest that dataset heterogeneity is well captured by a small set of meta-features encoding complexity, feature variability, and data quality. Concretely:

- `gravity`: Measures the Minkowski distance between the centers of mass of the majority and minority classes. A smaller distance indicates higher overlap between class distributions and, thus, greater classification difficulty.
- `inst_to_attr`: The ratio of the number of instances to the number of attributes, reflecting the balance between samples and features in the dataset.

There are four meta-features encoding the heterogeneity of features.

- `sparsity_mean`, `sparsity_std` : Quantify the variability of unique values in numeric features, where sparsity for a feature vector v is defined as:

$$S(v) = \frac{1.0}{n - 1.0} \cdot \left(\frac{n}{\phi(v)} - 1.0 \right),$$

  with n representing the instance number and $\phi(v)$ the number of distinct values in v.
- `entropy_mean`, `entropy_std`: Reflect the diversity in feature value distributions, where entropy for a feature v is calculated using:

$$H(v) = -\sum_{k} p_k \cdot \log_{\text{base}}(p_k),$$

  where p_k is the proportion of instances with the k-th unique value, and the logarithm base equals the number of unique values in v.
- `iq_range_std`: The standard deviation of interquartile ranges ($Q_3 - Q_1$) across all attributes, capturing variability in feature spread.
- `range_mean`: The mean range ($\max - \min$) across all attributes.

There is a meta-feature encoding the data quality of a tabular dataset.

- `nr_outliers` encodes the quality of a tabular dataset. In detail, it is the number of features containing at least one outlier value, where an outlier is defined as a value lying outside 1.5 times the interquartile range (IQR).

These meta-features collectively provide a robust way for assessing dataset characteristics, capturing factors such as class separability, feature variability, and the presence of anomalies. By formalizing these properties, we enable a systematic evaluation of dataset properties and their impact on model performance.

6.4 Correlation between Selected Meta-Features and Types of Tabular Methods

We analyze the relationship between dataset meta-features and the performance of nine representative methods introduced in section 4. By pairing each dataset's selected meta-features with the corresponding average rank of each method, we identify the meta-feature exhibiting the strongest absolute correlation with that method's performance. A higher absolute correlation indicates that the property revealed by the meta-feature has a stronger influence on the performance of the corresponding tabular method.

Table 2: Meta-features with the strongest correlation to the performance (average rank) of each representative method. The correlation values indicate the strength of the relationship, with negative values showing better performance as the meta-feature decreases.

Method	Correlation Value	Meta-Feature
XGBoost	0.3809	`entropy_mean`
LightGBM	-0.3487	`entropy_std`
Catboost	0.3101	`entropy_mean`
Random Forest	-0.2571	`sparsity_std`
RealMLP	-0.2819	`inst_to_attr`
TabM	-0.2059	`sparsity_mean`
FT-T	-0.2671	`inst_to_attr`
ModernNCA	0.1805	`entropy_std`
TabPFN v2	0.2765	`inst_to_attr`

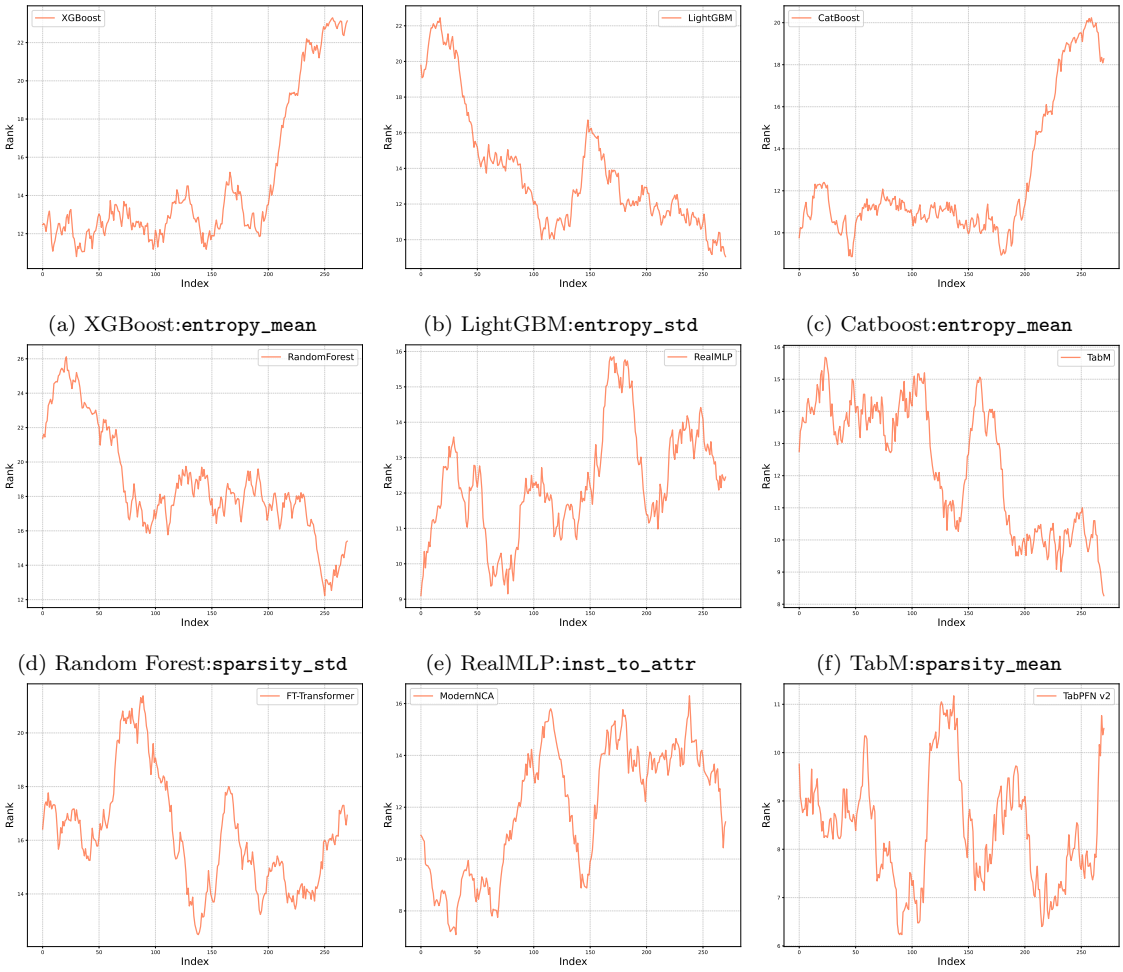
Table 2 summarizes the meta-feature most strongly correlated with the performance of each representative method. A negative correlation indicates that higher values of the meta-feature are associated with better performance, while a positive correlation suggests the opposite.

Specifically, most methods show the strongest correlation with *Feature Heterogeneity* metrics, highlighting the impact of feature distribution variability on performance. For example, XGBoost and CatBoost are most correlated with `entropy_mean` , while LightGBM and ModernNCA show their strongest correlations with `entropy_std`. Random Forest is most sensitive to `sparsity_std`, whereas TabM relies more on `sparsity_mean`. Several deep learning methods—including RealMLP, FT-Transformer, and TabPFN v2—exhibit strong correlations with `inst_to_attr` , a meta-feature representing the ratio of instances to attributes. This diversity underscores that different tabular methods are influenced by distinct dataset properties, even though feature heterogeneity metrics emerge as the most consistent driver across models.

To further illustrate these relationships, we visualize how the rank of different methods changes with respect to their most correlated meta-feature in Figure 12. The horizontal axis shows the sorted values of the most correlated meta-feature, while the vertical axis indicates the average rank for each representative method.

In some cases, the rank changes monotonically. For example, in Figure 12 (b), LightGBM shows a decreasing trend in rank as `entropy_std` increases. Since the standard deviation of entropy captures the variation in feature types within a dataset, this observation suggests that tree-based methods perform better (achieve lower ranks) when the dataset contains a higher diversity of feature types.

Conversely, several deep tabular methods, including RealMLP, FT-Transformer, and TabPFN v2, show a monotonic increase in rank as `inst_to_attr` increases (see Figure 12 (e), (g), and (i)). This indicates that these methods tend to perform worse when the number of instances relative to the number of features is high, highlighting their sensitivity to dataset size and feature dimensionality.



(a) XGBoost:`entropy_mean` (b) LightGBM:`entropy_std` (c) Catboost:`entropy_mean`

(d) Random Forest:`sparsity_std` (e) RealMLP:`inst_to_attr` (f) TabM:`sparsity_mean`

(g) FT-Transformer:`inst_to_attr` (h) ModernNCA:`entropy_std` (i) TabPFN v2:`inst_to_attr`

Figure 12: Dynamics of the average rank for each representative method as shown in Table 2, plotted against datasets ordered by their most correlated meta-feature. Each subfigure corresponds to a selected method and its associated meta-feature. The horizontal axis represents datasets ranked by ascending values of the meta-feature, while the vertical axis shows the rank. To enhance readability, the curves have been smoothed.

TabM and Random Forest exhibit clear correlations with sparsity-related meta-features. Specifically, TabM's rank decreases as `sparsity_mean` decreases (better performance on denser datasets), while Random Forest shows a decreasing trend in rank with lower `sparsity_std` (see Figure 12 (d) and (f)).

Some methods, such as ModernNCA, display moderate sensitivity to feature heterogeneity (`entropy_std`), with rank fluctuating in response to increasing variability (see Figure 12 (h)). These observations collectively highlight the diverse ways in which dataset properties influence the performance of different tabular methods.

Table 3: Meta-features with the strongest correlation to the performance (average rank) of tree-based methods, DNN-based methods, and their differences. The correlation values indicate the strength of the relationship, with negative values showing better performance as the meta-feature increases.

Category	Correlation Value	Meta-Feature
Tree	-0.3854	`entropy_std`
DNN	-0.3502	`inst_to_attr`
Tree-DNN	-0.3333	`entropy_std`

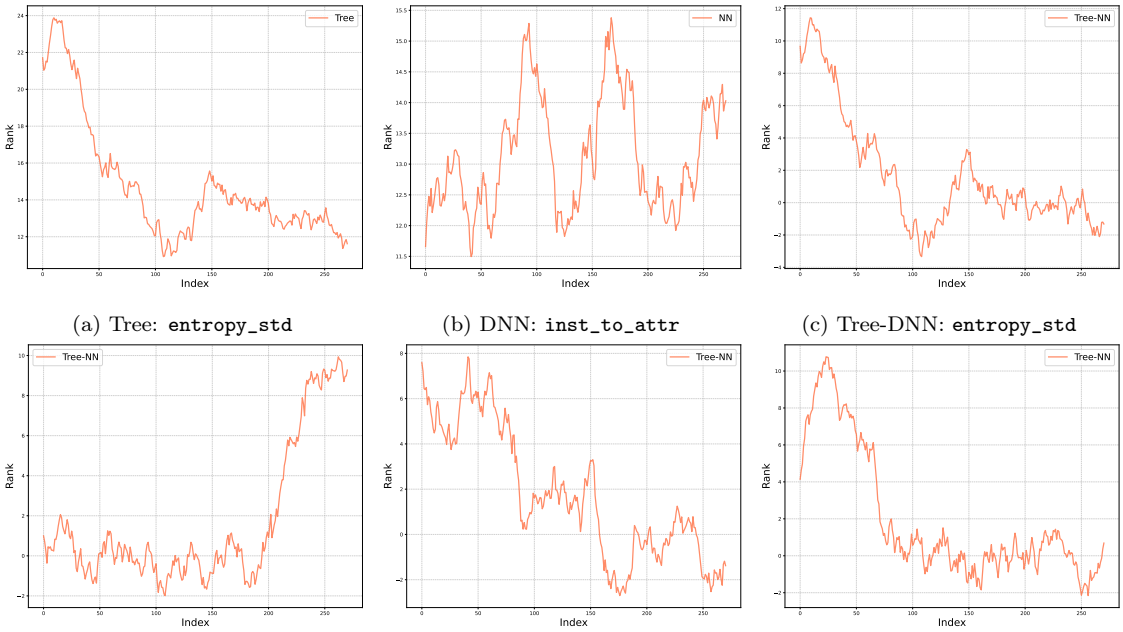
6.5 Analysis of the Tree-DNN Performance Gap via Selected Meta-Features

We further analyze which meta-features most significantly influence the performance gap between tree-based and DNN-based methods. Tree-based methods include Random Forest, XGBoost, LightGBM, and CatBoost. We choose representative DNN-based methods with low average ranks, *i.e.*, RealMLP, TabM, ModernNCA, and FT-Transformer.

For each dataset, we compute the performance gap as the **difference in average rank values** between tree-based and DNN-based methods. This tree-DNN performance gap quantifies the relative advantage or disadvantage of one category over the other. The meta-features most strongly correlated with the performance of tree-based methods, DNN-based methods, and the tree-DNN gap are listed in Table 3.

Results presented in Table 3 establish a strong negative correlation between the Tree-DNN performance gap and Feature Heterogeneity metrics, exemplified by `entropy_std` (Correlation: -0.3333). This observation contrasts with the meta-feature `inst_to_attr`—which has been emphasized in previous studies (McElfresh et al., 2023)—as it exhibits a markedly weaker correlation with the performance differential. These findings suggest that the degree of heterogeneity among dataset features is a more critical determinant driving the relative model superiority. Specifically, datasets characterized by greater feature heterogeneity (*e.g.*, higher variability in feature `entropy` or `sparsity`) tend to confer an advantage to tree-based methods, likely attributable to their intrinsic ability to effectively handle diverse and non-uniform feature distributions through successive partitioning.

The visualizations in Figure 13 provide empirical substantiation for these correlations. Figure 13(a) illustrates that as `entropy_std` increases (along the index dimension), the average rank for tree-based methods consistently **decreases**, indicative of performance improvement. Crucially, Figure 13(c), which depicts the Tree-DNN gap against `entropy_std`, exhibits a clear negative slope (downward trend). Since the gap is defined as $\text{Rank}_{\text{Tree}} - \text{Rank}_{\text{DNN}}$, a consistently negative value confirms that $\text{Rank}_{\text{Tree}} < \text{Rank}_{\text{DNN}}$, thereby establishing that tree-based models achieve superior performance relative to neural networks as feature heterogeneity intensifies. Conversely, Figure 13(b) presents the change in DNN rank with respect to `inst_to_attr`; its overall **ascending trend** signifies that performance degrades as `inst_to_attr` increases, an outcome consistent with the negative correlation of -0.3502 reported in Table 3.



(a) Tree: `entropy_std` (b) DNN: `inst_to_attr` (c) Tree-DNN: `entropy_std`

(d) Tree-DNN: `entropy_mean` (e) Tree-DNN: `iq_range_std` (f) Tree-DNN: `sparsity_std`

Figure 13: (a)-(b): The change of average rank from tree-based methods and DNN-based methods w.r.t. the most correlated meta-feature `entropy_std`, respectively. We further consider the differences in relative performance between tree-based models and DNN-based models, represented by the difference in average rank between the two types of methods. Plots in (c)-(f) highlight the changes of the tree-DNN gap against other meta-feature, such as `entropy_std`, `entropy_mean`, `sparsity_std`, `iq_range_std`, and `inst_to_attr`. To enhance readability, the curves have been smoothed.

6.6 Performance Comparison across Different Dataset Feature Types

We validate the previous observations using real-world tabular datasets by comparing the performance of various methods across datasets with different feature types: purely numerical features (*No Cat Data*), purely categorical features (*No Num Data*), and mixed feature types (*Mixed Data*). This analysis explores how dataset feature types influence performance, with particular emphasis on heterogeneity metrics such as `sparsity_attr` or `entropy_attr`. The results are visualized in a radar chart in Figure 14.

We have several key observations from Figure 14. First, raw-feature-based DNN methods perform poorly on mixed feature datasets. DNN-based methods such as MLP and RealMLP exhibit their worst performance on datasets with mixed feature types (*Mixed Data*). This observations validates the challenges faced by raw-feature-based neural networks when dealing with datasets characterized by significant heterogeneity in feature distributions. These methods perform better on purely numerical (*No Cat Data*) or purely categorical (*No Num Data*) datasets, where the homogeneity of feature types reduces the learning complexity.

Tree-based methods like XGBoost, LightGBM, CatBoost, and Random Forest excel on heterogeneous datasets. The results underscore their inherent advantage in handling datasets with heterogeneous feature distributions, including a mix of numerical and categorical features.

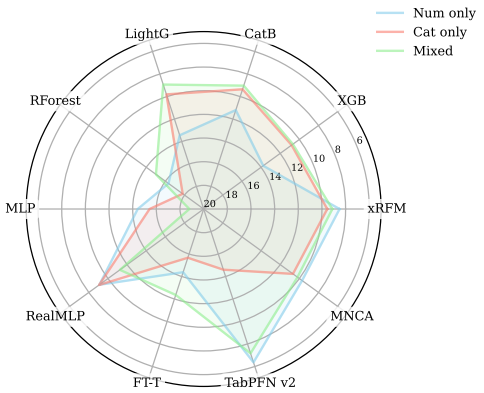


Figure 14: Performance comparison of methods across datasets with purely numerical features, purely categorical features, and mixed feature types. Tree-based methods include XGBoost, LightGBM, CatBoost, and Random Forest. DNN-based methods include MLP, ResNet, and TabR. Token-based methods include FT-T, AutoInt, and ExcelFormer.

These methods effectively leverage feature splitting and hierarchical decision-making, making them robust to varying feature types.

Token-based architectures, exemplified by the FT-Transformer (FT-T), demonstrate performance metrics that align more closely with tree-based models than traditional raw-feature-based DNN methods. This observation suggests that the learned embeddings for categorical and numerical features in FT-T enable a superior capability to manage the challenges inherent in mixed feature datasets. By effectively encoding and projecting features into a unified embedding space, the model likely mitigates the effects of feature variability and heterogeneity, thus facilitating better generalization across heterogeneous datasets.

Additionally, analysis of the specialized model **TabPFN v2** reveals distinct performance dependencies on feature types. TabPFN v2 exhibits its strongest performance on datasets composed purely of **numerical features**, followed by mixed feature datasets, while demonstrating the poorest results on purely **categorical feature datasets**. This pattern is intrinsically linked to TabPFN v2's pretraining data generation methodology and its inherent strategy for handling categorical features, which typically involves treating them as numerical data after simple preprocessing (such as one-hot or ordinal encoding). Consequently, the development of robust and generalizable strategies for handling **categorical features** remains a critical challenge for future research in the design of high-performing, universal tabular models.

7 Lightweight and Stress-Test via Talent-Tiny and Talent-Extension

Although the proposed large benchmark facilitates the analysis of deep tabular models, running a single method on all the datasets incurs a high computational burden. In this section, we extract a subset of the benchmark containing 15% of the full benchmark, *i.e.*, 45 datasets, to enable more efficient tabular analysis. We also collect an extension set

with challenging tabular datasets for stress testing. The statistics of all datasets are listed in Table 1.

7.1 TALENT-Tiny: A Compact Benchmark for Efficient and Detailed Evaluations

Selection strategy. As mentioned in section 4, TALENT is designed with a large collection of tabular datasets covering diverse characteristics. To curate TALENT-tiny, we apply stricter rules to remove datasets from other modalities, those with inherent distribution shifts or known leakage, and duplicated variants. To ensure representativeness, we also consider the "evolved" tree–DNN debate (see subsection 5.6), selecting datasets where both tree-based and DNN-based methods exhibit diverse behaviors.

We base this selection on the Tree–DNN score (Equation 2), which quantifies a dataset's preference for representative tree-based versus DNN-based methods. We categorize datasets into three groups: tree-friendly, DNN-friendly, and tie. For each task type (binary classification, multi-class classification, and regression), we partition datasets into groups by size ($N \times d$) to ensure small, medium, and large problems are all represented. From each size group, we select one dataset from each Tree–DNN category (tree, DNN, and tie). When a group has multiple candidates, we prefer datasets with clearer signal-to-noise ratios, higher-quality metadata, and balanced categorical vs. numerical feature compositions.

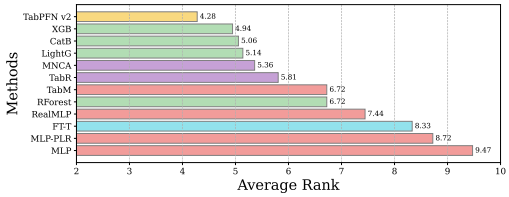
To further promote diversity, we refine the pool by enforcing representation across 14 application domains (*e.g.*, `biology`, `finance`, and `healthcare`). In cases where two datasets are similar, we substitute with an alternative to avoid redundancy. This strategy results in 45 datasets: including 15 binary classification, 12 multi-class classification, and 18 regression tasks. The final subset balances dataset size, feature type, domain, and method preference, ensuring TALENT-tiny is compact yet representative for controlled, efficient evaluations.

Analysis on the necessity for cross-validation. Recent studies argue that the hold-out strategy may cause hyperparameters to overfit the validation set (Tschalzev et al., 2024; Erickson et al., 2025), while cross-validation (CV) with ensembling provides more stable evaluation. However, CV greatly increases tuning costs, making it infeasible across 300 datasets in TALENT. Here, we compare both strategies on TALENT-tiny.

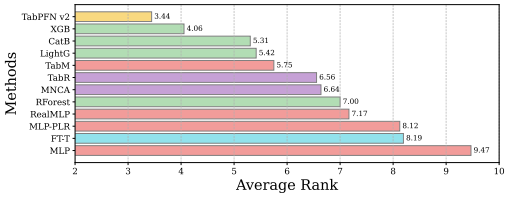
Figure 15 (left) reports results with the hold-out strategy. The ranking patterns reflect those of the full benchmark, with tree-based ensembles (*e.g.*, CatBoost, LightGBM, XGBoost) showing particularly strong performance in binary classification, and RealMLP and MNCA performing well in multi-class and regression. Importantly, the relative order of methods aligns with large-scale results, indicating that the subset remains representative.

Figure 15 (right) shows results with cross-validation plus ensembling. Across most methods, average ranks improve slightly compared to hold-out, confirming that ensembling boosts stability. However, the relative order among methods remains largely unchanged. For example, tree-based methods still dominate in binary classification, while RealMLP and MNCA maintain strong performance in regression and multi-class classification. This validates the use of hold-out in TALENT, given the impractical cost of CV across all datasets.

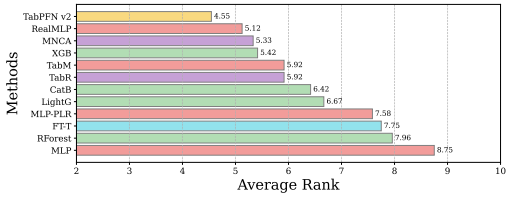
Interestingly, MNCA does not benefit much from vanilla ensembling, in contrast to the clear gains observed with its dedicated ensemble variant (MNCA-ens) in earlier results. This suggests that some methods require specialized ensemble designs—for MNCA, strategies that increase neighborhood diversity may be particularly important. In contrast, gradient



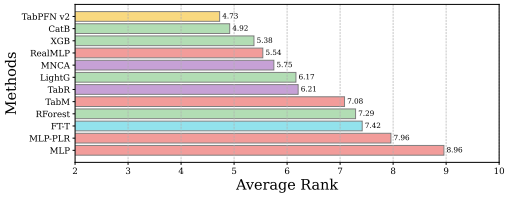
(a1) Binary Classification (Hold-out)



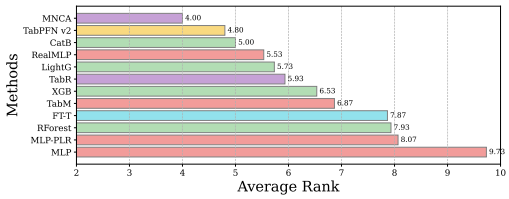
(a2) Binary Classification (CV + Ensemble)



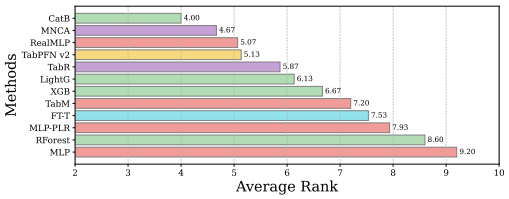
(b1) Multi-Class Classification (Hold-out)



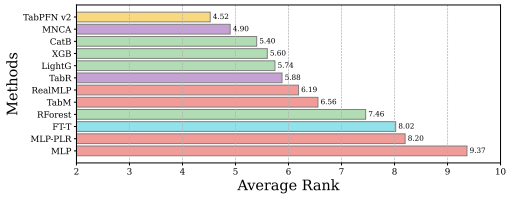
(b2) Multi-Class Classification (CV + Ensemble)



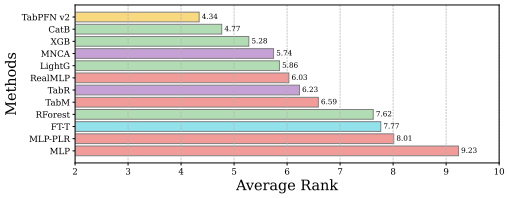
(c1) Regression (Hold-out)



(c2) Regression (CV + Ensemble)



(d1) All Tasks (Hold-out)



(d2) All Tasks (CV + Ensemble)

Figure 15: Average rank of representative tabular methods on Talent-tiny under two evaluation protocols: (left) the original hold-out strategy and (right) cross-validation with ensemble. Ranks are computed based on accuracy for classification tasks and RMSE for regression tasks. Lower rank values indicate stronger performance. While CV+ensemble generally improves absolute performance values, the relative ordering among methods remains stable, validating the practicality of the hold-out strategy for large-scale benchmarks.

boosting methods and RealMLP benefit naturally from CV+ensemble, showing reduced variance without needing customized ensemble mechanisms.

Overall, Talent-tiny proves useful for efficient yet representative analysis. The comparison of hold-out versus CV+ensemble indicates that while ensembling stabilizes results, the hold-out strategy provides reliable relative rankings across methods, justifying its adoption in Talent. At the same time, the divergent behaviors of methods like MNCA highlight that ensemble strategies must sometimes be tailored to model design rather than applied uniformly.

Table 4: Performance of MLP models under different hyperparameter tuning trials, measured by average rank across all methods. The six groups correspond to subsets divided by dataset sample size, while the last column reports the overall performance. Parentheses indicate p-values. Results show that compared with our standard setting of 100 tuning trials, 50 trials are insufficient for effective tuning, whereas increasing the number of trials beyond 100 does not yield further improvement.

	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Overall
50 trials	+0.63 (0.144)	+0.30 (0.147)	+0.48 (0.547)	+0.64 (0.214)	+0.11 (0.919)	+1.53* (0.003)	+0.61* (0.002)
100 trials	17.54	17.10	18.57	18.32	17.36	15.94	17.47
150 trials	+0.07 (0.828)	−0.03 (0.879)	+0.07 (0.958)	−0.48 (0.522)	−0.54 (0.211)	−0.40 (0.303)	−0.22 (0.262)
200 trials	+0.20 (0.750)	−0.29 (0.965)	+0.45 (0.547)	−0.65 (0.298)	−0.54 (0.324)	0.00 (0.845)	−0.14 (0.602)

Analysis on the number of hyperparameter search. We further investigate the influence of the number of hyperparameter search trials, which was set to 100 in our previous experiments following Gorishniy et al. (2021). To assess the sensitivity of performance to search effort, we evaluate the MLP method with 50, 100, 150, and 200 trials, and compare their average ranks across subsets of datasets of different sizes.

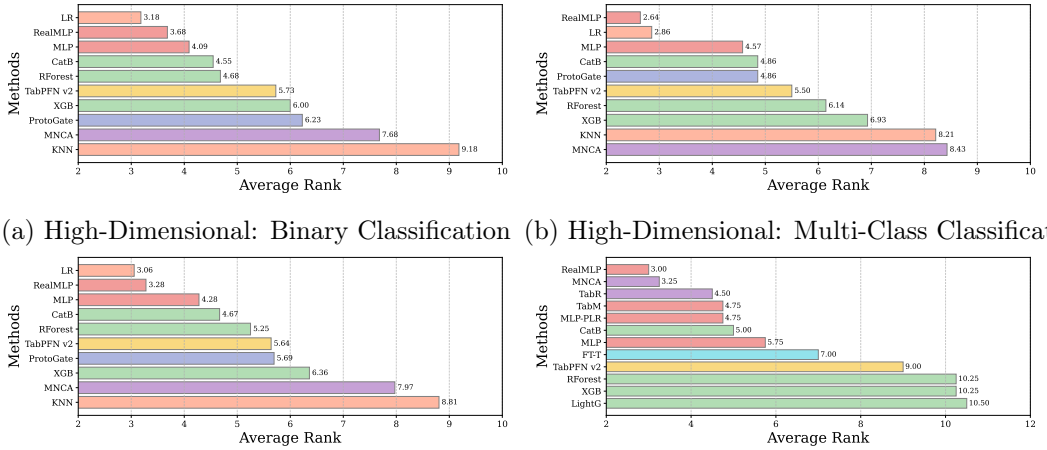
The results, shown in Table 4, reveal several important trends. On smaller datasets, increasing the number of trials beyond 50 yields diminishing returns: the gap between 50 and 100 trials is noticeable, but further increases to 150 or 200 trials do not lead to consistent improvements. On larger datasets, additional search efforts beyond 100 trials bring marginal but still limited gains. This suggests that 50 trials are insufficient for stable optimization, but 100 trials already provide a near-saturation point for tuning effectiveness.

Across all dataset groups, the overall differences among 100, 150, and 200 trials are relatively minor, as confirmed by the comparative rankings in the rightmost panel. The performance curves of these three settings are nearly overlapping, indicating that the benefit of exhaustive hyperparameter search is minimal once a certain search budget is reached. Interestingly, while 50 trials consistently rank lower, the rank order of 100, 150, and 200 trials fluctuates slightly across dataset subsets without forming a clear hierarchy.

These results support the use of 100 trials as a balanced and practical setting in large-scale benchmarks like Talent, since it offers a strong trade-off between computational efficiency and model competitiveness. Moreover, they highlight that blindly scaling up search budgets does not guarantee better results, especially for tabular tasks where model robustness may dominate over hyperparameter fine-tuning.

7.2 Talent-Extension: Stress Testing in Challenging Scenarios

Talent provides two complementary layers of coverage. While the main Talent benchmark provides broad coverage of typical tabular tasks, real-world data often exhibit more extreme conditions that challenge scalability and model robustness. To explore these regimes, we introduce Talent-extension, a complementary suite designed to stress-test tabular methods under three specialized yet practically important settings: *high-dimensional feature spaces*, *many-class classification problems*, and *very-large-scale datasets*. These settings expose



(a) High-Dimensional: Binary Classification (b) High-Dimensional: Multi-Class Classification

(c) High-Dimensional: All Tasks (d) Many-Class: All Tasks

Figure 16: Average rank of representative methods on Talent-extension under high-dimensional and many-class conditions. Subfigures (a)–(c) correspond to high-dimensional datasets, while (d) summarizes the results for many-class classification. Ranks are computed using accuracy for classification and RMSE for regression (lower is better).

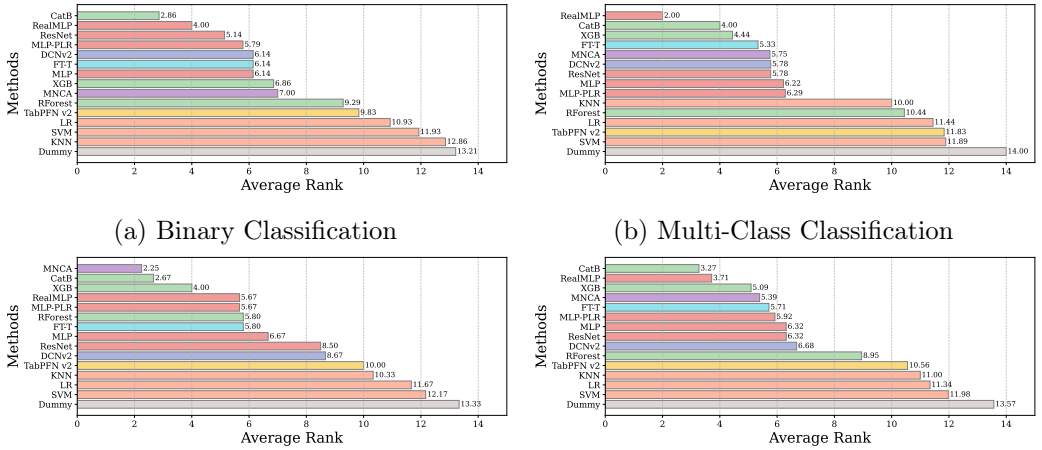
performance bottlenecks that are not always visible in standard-sized datasets and thus provide a deeper understanding of each model's inductive biases.

Dataset groups. Talent-extension contains three groups of specialized tabular tasks.

- **High-dimensional datasets.** This group contains 18 datasets with feature dimensionality ranging from 2,000 to over 20,000 (Table 1). Representative examples include biomedical datasets such as `colon`, `glioma`, and `TOX_171`, as well as text-derived datasets like `BASEHOCK` and `RELATHE`.
- **Many-class datasets.** This group includes 12 datasets with more than ten classes, such as `orlraws10P` (10 classes, 10,304 features) and `Fashion-MNIST` (10 classes, 784 features). These datasets emphasize the difficulty of learning fine-grained label structures where class-aware objectives and embeddings are critical.
- **Very large-scale datasets.** This group comprises 18 datasets containing hundreds of thousands to millions of instances, such as `Airlines_DepDelay` (10M samples), `Higgs` (1M samples), and `sf-police-incidents` (2.2M samples). They test the computational scalability of tabular methods under massive data volumes.

The evaluation protocol follows the same setup as in Talent, except for high-dimensional datasets, where limited sample sizes motivate aggregated cross-validation results with default hyperparameters. We evaluate representative methods from classical baselines (Logistic Regression, kNN), tree ensembles (Random Forest, XGBoost, LightGBM, CatBoost), and deep tabular architectures (MLP, RealMLP, FT-Transformer, MNCA, TabM). Some other methods, such as MNCA-ens and TabM, require significantly longer training times in these specialized scenarios, so we omit them from the extended evaluation.

Results and analysis. The Talent-extension results reveal several notable shifts relative to the main Talent benchmark:



(a) Binary Classification (b) Multi-Class Classification

(c) Regression (d) All Tasks

Figure 17: Average rank of representative methods on Talent-extension (very large-scale datasets). Ranks are computed based on accuracy and RMSE for classification and regression tasks, respectively. Lower rank values indicate better overall performance.

- **High-dimensional datasets.** On high-dimensional datasets, the results (Figure 16a–c) reveal several notable deviations from the main Talent findings. Surprisingly, Logistic Regression emerges as one of the strongest performers, consistently ranking near the top across binary, multi-class, and aggregated tasks. This indicates that in ultra-sparse, high-dimensional spaces—common in biomedical and text-derived data—simpler linear models with regularization can generalize better than complex nonlinear architectures. In contrast, methods like ModernNCA and KNN, which rely on neighborhood retrieval, suffer significant degradation. Their performance drop reflects the *curse of dimensionality*, where distance-based similarity becomes less meaningful as dimensionality grows. Tree-based ensembles (*e.g.*, CatBoost, LightGBM, XGBoost) remain stable but are not dominant; their greedy feature-splitting mechanisms struggle when the number of irrelevant or redundant features is large. Interestingly, RealMLP and standard MLPs show resilience—likely due to their ability to perform distributed feature selection through gradient-based learning—but still lag slightly behind linear baselines in some cases. Pretrained foundation models such as TabPFN v2 perform poorly in this setting, suggesting that pretrained priors, while powerful for small to mid-size datasets, transfer poorly when the feature distributions of target data differ drastically from those seen during pretraining. This performance gap points to a distribution mismatch between synthetic pretraining corpora and real high-dimensional domains, where feature semantics differ and sample sparsity limits adaptation.
- **Many-class datasets.** In the many-class regime (Figure 16d), the trends diverge from the high-dimensional case. Deep models with strong *representation learning* capacity regain their lead: RealMLP, ModernNCA, and TabR rank at the top, followed closely by TabM and MLP-PLR. Their ability to learn shared embeddings across fine-grained label spaces proves crucial for distinguishing numerous closely related classes. Tree-based ensembles such as CatBoost and XGBoost remain competitive but no longer dominant, suggesting that their partition-based decision structures may not scale efficiently with

large label cardinalities. Interestingly, while foundation models (*e.g.*, TabPFN v2) excelled in the main Talent benchmark, they perform inconsistently here, further underscoring the limitation of in-context pretrained inference when label structures diverge from the few-class distributions prevalent during pretraining. Ensemble-style deep models such as TabM consistently outperform their base DNNs, reaffirming that ensembling remains an effective strategy even in modern deep tabular learning.

- **Large-scale datasets.** As shown in Figure 17, very large datasets (up to millions of rows) produce a performance landscape that differs from the main Talent results. Classical tree ensembles, especially CatBoost, rank at or near the top across tasks. Beyond their inductive bias for categorical structure, a practical factor likely contributes: highly optimized implementations make it feasible to train and *ensemble many trees* at scale, which compounds accuracy. By contrast, ensembling deep tabular models is far more time-consuming (multiple large models must be trained), so neural methods seldom benefit from the same degree of ensemble amplification under strict compute budgets. Modern deep methods still perform well: RealMLP is consistently strong, indicating that well–regularized MLPs scale gracefully. Retrieval/attention models (*e.g.*, MNCA, FT-T) remain competitive but do not close the gap to the best tree ensembles. Pretrained foundation models (*e.g.*, TabPFN v2) lag in this regime, likely due to a distribution/scale mismatch with their pretraining setup and the lack of fine-tuning for massive datasets.

Broader observations. Taken together, the Talent-extension results show both continuity and clear departures from the main Talent findings. Several methods, such as RealMLP, retain their strengths across regimes, yet the ordering of most methods changes once we move to high dimensionality, many classes, or very large scale. In high-dimensional problems, simple linear models (*e.g.*, logistic regression) are unexpectedly competitive, and neighborhood/retrieval methods degrade, indicating that feature redundancy and the curse of dimensionality, rather than model expressivity, become the primary bottlenecks. For very large datasets, tree ensembles—most notably CatBoost—regain a clear edge.

Pretrained foundation models remain reasonably robust but do not dominate in these stress settings, indicating limits of pretraining when the target distribution departs from the pretraining regime. It is notable that our use follows the default deployment, and simple divide-and-conquer adaptations of TabPFN v2 have been shown to boost its effectiveness efficiently (Ye et al., 2025a; Rubachev et al., 2025b). A promising direction is to either endow tabular foundation models with an intrinsic ability to handle stress cases like large-scale datasets, or to develop corresponding lightweight adaptation strategies.

Overall, these stress tests refine the tree–DNN discussion. Foundation models and modern neural architectures have closed much of the gap in typical settings, yet tree ensembles remain hard to beat in very large-scale scenarios, and linear baselines re-emerge in ultra high-dimensional spaces. The evidence points toward hybrid, adaptive pipelines—combining strong trees, scalable MLPs, and pretrained components—as a principled path to robust tabular learning across diverse real-world conditions.

8 Conclusion

We presented a large-scale, systematic evaluation and analysis of deep tabular learning using Talent, a 300+ dataset collection spanning varied sizes, domains, feature compositions,

and task types. Across this breadth, method rankings do vary by dataset, but performance consistently concentrates within a small shortlist of models—offering a practical starting point for model selection. We also find that ensembling benefits both tree-based and DNN-based approaches: strong classical ensembles remain competitive, while recent pretrained (foundation) models frequently narrow—though do not fully eliminate—the historical advantage of trees. This refines the "trees vs. neural networks" narrative in today's landscape. To explain when different families win, we quantified dataset heterogeneity by learning from meta-features and early training dynamics to predict later validation behavior. The analysis highlights the roles of categorical–numerical interplay, sparsity, and entropy variation as key drivers of model advantage. Finally, our two-level design complements the main collection with Talent-tiny (45 carefully balanced datasets for rapid, reproducible evaluation) and Talent-extension (high-dimensional, many-class, and very large-scale settings for stress testing). Results on these subsets surface additional distinctions among model families and provide actionable guidance for heterogeneity-aware, ensemble-strengthened tabular learning.

Appendix A. Datasets Selection Details

This appendix provides detailed information on datasets with certain quality issues and the corresponding adjustments applied in our benchmark construction.

Datasets with mis-labeled task types. We identified 22 datasets whose task types were incorrectly labeled, including `Contaminant-9.0GHz`, `Contaminant-9.5GHz`, `Contaminant-10.0GHz`, `Contaminant-10.5GHz`, `Contaminant-11.0GHz`, `Heart-Disease-Dataset`, `Insurance`, `Intersectional-Bias`, `Is-this-a-good-customer`, `KDD`, `Long`, `Performance-Prediction`, `Shipping`, `VulNoneVul`, `Waterstress`, `compass_reg`, `credit_reg`, `law-school-admission`, `ozone_level`, `shill-bidding`, `shrutime`, and `svmguide3`. After reviewing their metadata and label structures, we corrected these datasets to binary classification tasks.

Tabular datasets derived from other modalities. Our benchmark includes 25 datasets where tabular features are extracted from non-tabular sources such as images or audio. These include `Indian_pines`, `JapaneseVowels`, `Parkinsons_Telemonitor`, `artificial-characters`, `dry_bean_dataset`, `hill-valley`, `kropt`, `letter`, `mfeat-factors`, `mfeat-fourier`, `mfeat-karhunen`, `mfeat-morphological`, `mfeat-pixel`, `100-plants-margin`, `100-plants-shape`, `100-plants-texture`, `optdigits`, `page-blocks`, `pendigits`, `phoneme`, `satellite_image`, `satimage`, `segment`, `semeion`, and `texture`. Although some works (Kohli et al., 2024; Erickson et al., 2025) exclude such datasets, we retain them because they reflect practical cases where only pre-extracted features are available due to resource or efficiency constraints.

Datasets with known or potential leakage. Prior analyses (Rubachev et al., 2025a; Tschalzev et al., 2025) have shown that several public tabular datasets contain data leakage, which can distort model comparison outcomes. Specifically, leakage has been reported in `Kaggle_bike_sharing_demand_challange`, `Facebook_Comment_Volume`, `GesturePhaseSegmentationProcessed`, `artificial-characters`, `compass`, `electricity`, `eye_movements`, `eye_movements_bin`, `sulfur`, and `Brazilian_houses_reproduced`. These issues often arise from erroneous preprocessing or from features that directly encode the target variable. For instance, the `sulfur` dataset includes a feature that is a near-duplicate of the target variable, creating a direct leak (Rubachev et al., 2025a).

We also identify potential leakage in three additional datasets. In `Job_Profitability`, the target variable `Jobs_Gross_Margin_Percentage` can likely be inferred from the feature `Jobs_Gross_Margin`. In `CPMP-2015-regression`, the feature `run status` reveals information about the target runtime. In `estimation_of_obesity_levels`, the inclusion of raw height and weight features makes obesity prediction almost trivial.

Potential leakage when evaluating general tabular models. General tabular models are pretrained on multiple real-world or synthetic datasets and are often early-stopped using a separate validation set of real-world datasets. While these models can be efficiently applied to new datasets, the evaluation may lead to potential dataset leakage if the target dataset is part of the pretraining or validation datasets. In such cases, the general model might exhibit inflated performance due to prior exposure to the target dataset.

Specifically, we evaluate the general tabular model TabPFN (Hollmann et al., 2023), which is pretrained on synthetic datasets and early-stopped via its performance on 180 real-world datasets. The checkpoint selection rule of TabPFN potentially creates biases for these datasets. Among the datasets in our benchmark, we found only two of them, `PizzaCutter3` and `PieChart3`, that overlap with TabPFN's validation set. For TabPFN v2 (Hollmann et al., 2025), we further examined its validation set and identified **27 datasets** that overlap with those in our 300-dataset benchmark: `ada_prior`, `allbp`, `baseball`, `delta_ailerons`, `eye_movements`, `eye_movements_bin`, `GAMETES_Epistasis_-2-Way_20atts_0.1H_EDM-1_1`, `hill-valley`, `JapaneseVowels`, `jungle_chess_2pcs_raw_-endgame_complete`, `led24`, `longitudinal-survey`, `page-blocks`, `ringnorm`, `rl`, `thyroid-ann`, `waveform-5000`, `debutanizer`, `delta_elevators`, `mauna-loa-atmospheric`, `puma32H`, `stock_fardamento02`, `treasury`, `weather_izmir`, `wind`.

To maintain comparability with prior studies, these datasets are retained in the general Talent benchmark. However, they are excluded from the stricter Talent-tiny subset, which focuses on high-quality, leakage-free evaluation. This design enables fair historical comparison while supporting rigorous analysis in controlled settings.

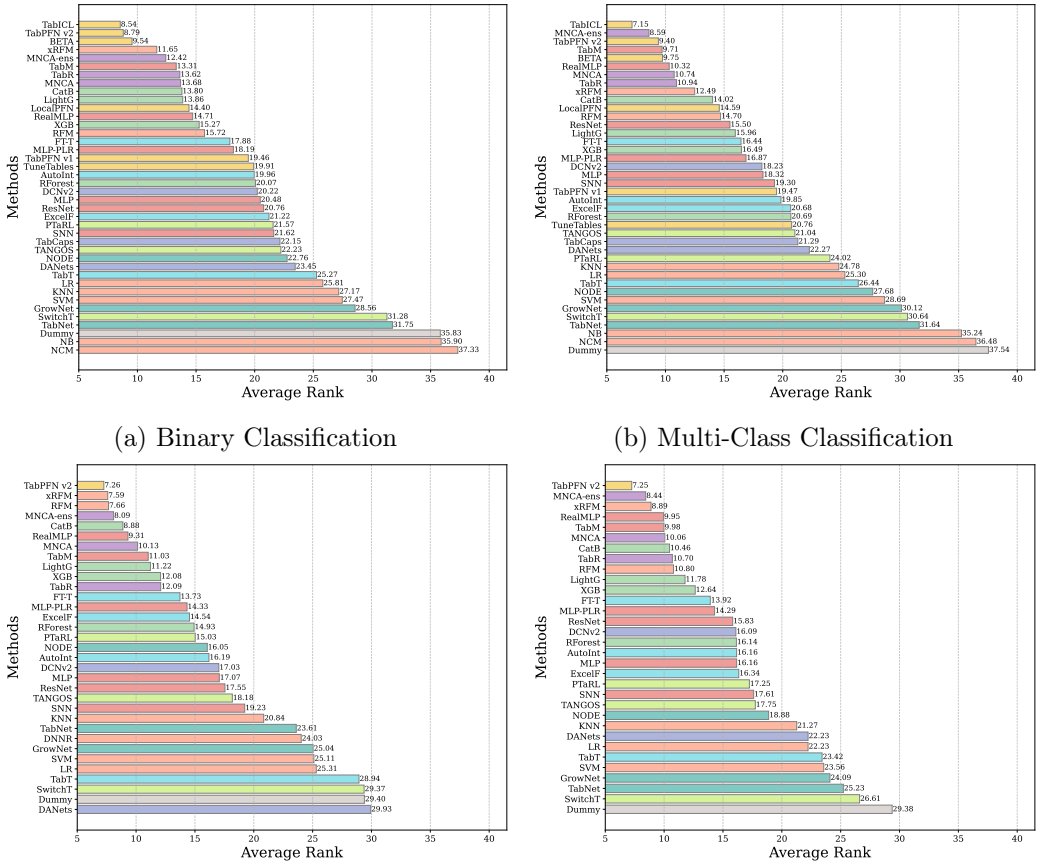
Appendix B. Additional Comparison Results

B.1 Average Performance and Rankings

To complement the main statistical comparisons, we report detailed average ranks and pairwise t-test results. The average ranks of 40 representative methods across 300 datasets are shown in Figure 18, while pairwise Win/Tie/Lose outcomes are illustrated in Figure 19. These provide a finer-grained view of the relative positioning of methods beyond the critical difference diagrams in the main text.

From Figure 18, several clear patterns emerge. First, pretrained tabular foundation models dominate across settings. TabPFN v2 and TabICL consistently achieve the lowest ranks in binary and multi-class classification, confirming the strength of pretraining and in-context learning strategies. In regression, TabPFN v2 remains the top performer, followed closely by xRFM, MNCA-ens, and RealMLP. Notably, foundation models are the only family that simultaneously excels across all three task types, underlining their broad generalization.

Tree-based ensembles continue to provide strong baselines. CatBoost and LightGBM achieve top-tier ranks across both classification and regression tasks, while XGBoost trails slightly but still outperforms most DNNs. Recursive Feature Machines (RFM) and its



(a) Binary Classification

(b) Multi-Class Classification

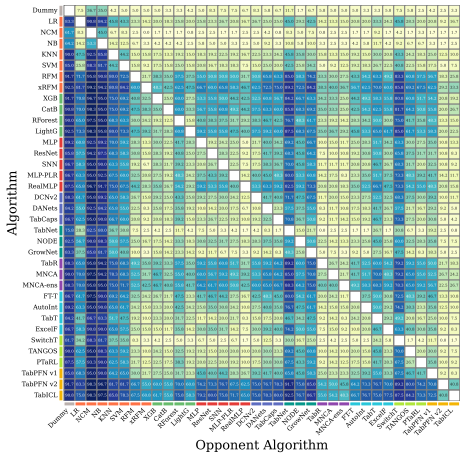
(c) Regression

(d) All Tasks

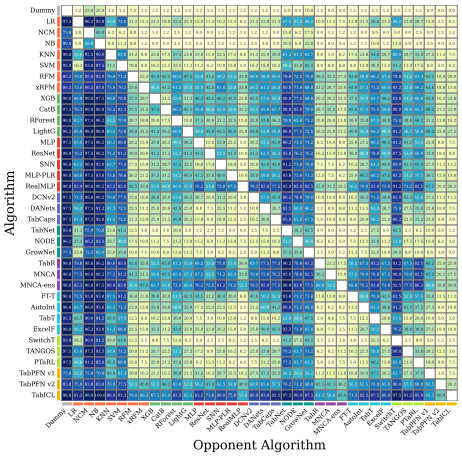
Figure 18: Average rank of tabular methods. We show the average rank of all methods over binary (120 datasets), multi-class (80 datasets), regression (100 datasets), and all 300 datasets. The ranks are calculated based on accuracy and RMSE over classification and regression tasks, respectively. The lower the rank value, the better the average performance.

extension xRFM perform particularly well on regression, often rivaling the strongest ensembles and DNNs.

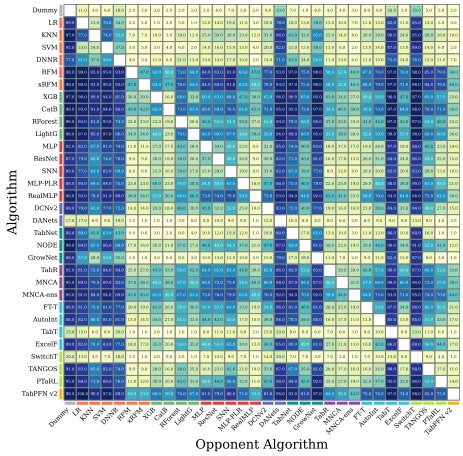
Among deep learning approaches, neighborhood-based methods stand out. ModernNCA achieves consistently high ranks across all task types and remains competitive with CatBoost and LightGBM. TabR also performs strongly in classification tasks. MLP variants show a clear divide: vanilla MLP is weak, but tuned designs such as MLP-PLR and RealMLP achieve much lower ranks, with RealMLP frequently joining the top-performing group. Token-based models (*e.g.*, FT-T, ExcelFormer, AutoInt) are stable performers, especially in classification, but their ranks indicate they are not decisively stronger than the best ensembles or neighborhood-based models. Tree-mimic models (NODE, TabNet, GrowNet) generally remain in the lower half of the rankings.



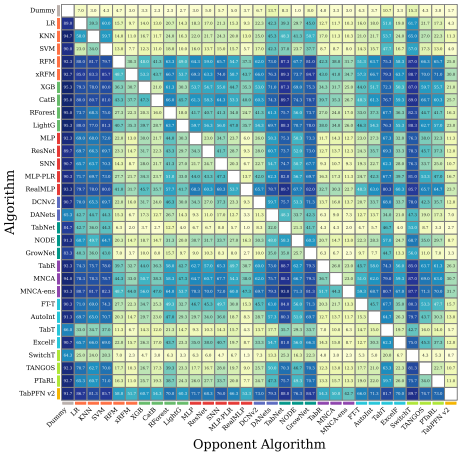
(a) Binary Classification



(b) Multi-Class Classification



(c) Regression



(d) All Tasks

Figure 19: Heatmaps illustrating the statistical comparisons between all pairs of methods based on t-tests with a 95% confidence interval. The Win/Tie/Lose counts between each pair of methods are also denoted. Darker colors indicate higher counts.

B.2 Pairwise Statistical Comparisons

To complement the average rank analysis, we further conduct pairwise statistical comparisons between all pairs of methods using t-tests with a 95% confidence interval. The results are visualized in Figure 19, which reports the Win/Tie/Lose counts between each pair of methods across binary, multi-class, regression, and all tasks. Darker colors indicate higher counts, reflecting more consistent superiority in the corresponding comparisons.

The heatmaps confirm many earlier findings: tree-based ensembles (CatBoost, LightGBM, XGBoost) dominate weaker baselines and cluster together as robust, statistically indistinguishable methods. RealMLP and MLP-PLR clearly outperform vanilla MLPs and ResNets, though they tie with strong ensembles in many cases. Token-based approaches

(*e.g.*, FT-T, ExcelFormer) show stable but not dominant behavior, often tying with both ensembles and tuned DNNs.

Neighborhood-based methods, especially ModernNCA and MNCA-ens, stand out as frequent winners in classification, rivaling ensembles and confirming the strength of retrieval-based learning. Finally, pretrained foundation models (TabPFN v2, TabICL) achieve the most consistent wins in binary and multi-class classification, while in regression, they remain statistically tied with ensembles and ModernNCA, indicating task-dependent benefits.

Overall, the pairwise comparisons highlight that while pretrained models push state-of-the-art performance, top ensembles and retrieval-based methods remain highly competitive, forming overlapping statistical equivalence groups across many tasks.

Appendix C. Details of the Heterogeneity Analysis

This section provides additional details complementing our analysis of dataset heterogeneity in section 6. We describe the meta-features employed, the recording of training dynamics, the curve families used for modeling validation trajectories, and supplementary results. Finally, we highlight a by-product of this framework: predicting the training dynamics of deep tabular models, which may enable more efficient training in practice.

C.1 Details of Meta-Features

The meta-features encode structural and statistical properties of a dataset (McElfresh et al., 2023). They form the foundation for analyzing how heterogeneity influences model behavior. By incorporating these meta-features, we not only characterize tabular datasets but also predict training dynamics, thereby identifying which dataset factors most strongly shape model performance. We provide the full list of all meta-features in Table 5.

C.2 Recording Training Dynamics

Beyond end-point accuracy or RMSE, we record detailed training dynamics for each dataset–method pair. These include:

- **Training logs**: learning rate schedules, batch-wise losses, and intermediate statistics.
- **Performance metrics**: validation/test loss, accuracy, RMSE, as well as secondary metrics such as F1/AUC (classification) and MAE/R^2 (regression).
- **Running time**: measured across 15 random seeds, accounting for early stopping at variable epochs.
- **Model size**: for both default and tuned hyperparameters.

These logs provide a rich foundation for connecting dataset properties with optimization behavior, thereby supporting heterogeneity analysis.

C.3 Alternative Curve Families for Validation Dynamics

To forecast validation trajectories from early training segments, we experimented with several curve families inspired by prior work in vision and language (Hestness et al., 2017; Rosenfeld et al., 2020; Bahri et al., 2021). Let t denote epoch and y the performance measure. We consider four functional forms:

- **M1**: $y = at^b$ (basic power-law form) (Alabdulmohsin et al., 2022).

Table 5: Meta-features used in the training dynamics prediction task. The first column indicates the selected key meta-features.

Selected	Meta-Feature	Explanation
	attr_conc	The concentration coef. of each pair of distinct attributes.
✓	class_conc	The concentration coefficient between each attribute and class.
	class_ent	The target attribute Shannon's entropy.
✓	inst_to_attr	The ratio between the number of instances and attributes.
✓	mean	The mean value of each attribute.
	sd	The standard deviation of each attribute.
	var	The variance of each attribute.
✓	range	The range (max - min) of each attribute.
✓	iq_range	The interquartile range (IQR) of each attribute.
✓	nr_attr	The total number of attributes.
✓	sparsity	The (possibly normalized) sparsity metric for each attribute.
	t_mean	The trimmed mean of each attribute.
	nr_bin	The number of binary attributes.
	nr_cat	The number of categorical attributes.
	nr_num	The number of numeric features.
	nr_norm	The number of attributes normally distributed based in a given method.
	nr_cor_attr	The number of distinct highly correlated pair of attributes.
✓	gravity	The distance between minority and majority classes' center of mass.
	nr_class	The number of distinct classes.
✓	joint_ent	The joint entropy between each attribute and class.
✓	attr_ent	Shannon's entropy for each predictive attribute.
✓	cov	The absolute value of the covariance of distinct dataset attribute pairs.
	eigenvalues	The eigenvalues of covariance matrix from dataset.
	eq_num_attr	The number of attributes equivalent for a predictive task.
✓	max	The maximum value from each attribute.
	min	The minimum value from each attribute.
	median	The median value from each attribute.
	freq_class	The relative frequency of each distinct class.
	mad	The Median Absolute Deviation (MAD) adjusted by a factor.
✓	mut_inf	The mutual information between each attribute and target.
✓	nr_inst	The number of instances (rows) in the dataset.
✓	nr_outliers	The number of attributes with at least one outlier value.
✓	ns_ratio	The noisiness of attributes.
✓	imblance_ratio	The ratio of the number of instances in the minority to the majority class.
	attr_to_inst	The ratio between the number of attributes.

- **M2**: $y = at^b + c$ (shifted power-law) (Cortes et al., 1993; Hestness et al., 2017; Rosenfeld et al., 2020; Abnar et al., 2022).
- **M3**: $y = a(t + d)^b + c$ with offset d controlling the onset of improvement.
- **M4**: $(y - \epsilon_\infty)/((\epsilon_0 - y)^a) = bt^c$, where ϵ_∞ is irreducible error and ϵ_0 random-guess performance.

Parameters are estimated from the initial support set $\mathcal{S}$ (first epochs). Once fitted, these forms extrapolate the query set $\mathcal{Q}$, enabling curve reconstruction.

Table 6: Average MAE and OVD for various curves across test datasets. Both metrics are "lower is better." "Ours" refers to directly fitting the curves using Equation 3. "Ours with MLP" indicates the method using the learned h.

Comparison Methods	MAE	OVD
M1	0.1845	0.2936
M2	0.8458	6.0805
M3	0.1331	0.1148
M4	0.1818	0.1794
Direct Fit (ours)	1.1927	1.8860
Meta-learned h (ours)	**0.0748**	**0.0701**

C.4 Main Results and Analysis

We implement the meta-mapping h as a four-layer MLP. The input includes both the first 5 validation points and 19 meta-features, and the output is the parameter set θ defining our curve family.

We evaluate with Mean Absolute Error (MAE) and Optimal Value Difference (OVD). OVD measures the discrepancy between the optimal values of predicted and true curves (maximum for classification, minimum for regression). Results are reported in Table 6.

We also attempt to fit parameters for other curve families (M1–M4) using the optimization objective in Equation 1. However, these formulations often face convergence issues, and direct fitting with our curve form produces suboptimal results due to differences in fitting strategy. As shown in Table 6, the gap between "Direct Fit" and "Meta-learned h" underscores the necessity of incorporating meta-features: leveraging dataset properties in addition to early dynamics substantially improves curve prediction accuracy.

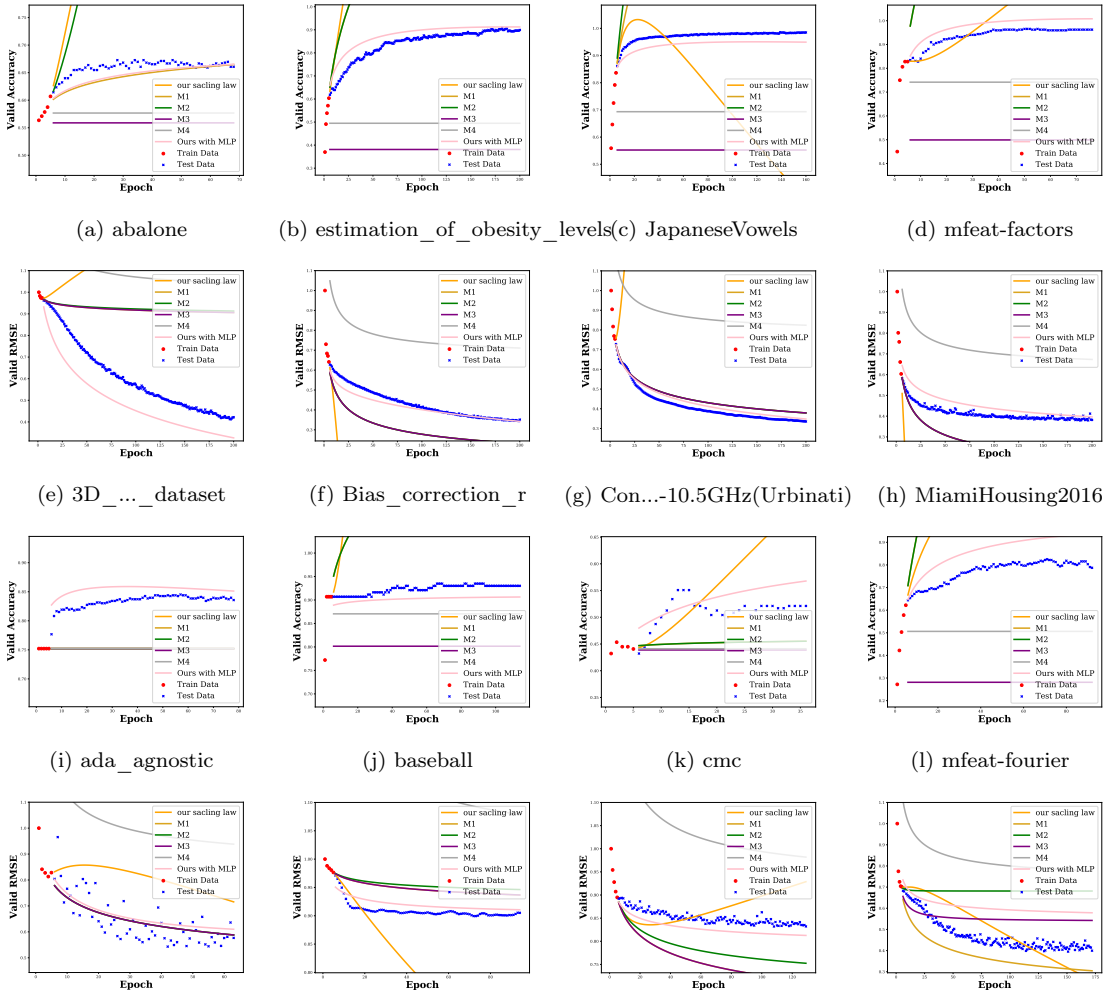
Importantly, the learned predictor h can accurately extrapolate the remaining validation performance curves from only the first few epochs.

Figure 20 illustrates qualitative fits for 16 unseen datasets. The predictor h successfully reconstructs both DNN-friendly and tree-friendly learning curves, spanning classification and regression tasks. Compared to baseline curve families, our method delivers more faithful extrapolations, especially when meta-features are included.

In summary, these results confirm that linking dataset meta-features to training dynamics is effective for characterizing heterogeneity. By capturing how dataset properties shape optimization trajectories, our approach not only predicts validation curves more accurately but also deepens understanding of why models succeed or fail on specific tabular datasets.

C.5 By-Product: Forecasting Training Dynamics for Efficiency

While our primary goal is to use dynamic forecasting as a tool for heterogeneity analysis, it also yields a practical by-product: efficient early stopping. Since deep tabular training is often expensive and hyperparameter-sensitive, forecasting later performance from early epochs allows us to prune poor runs. For example, if accuracy plateaus or oscillates early, training can be terminated and resources reallocated (Cortes et al., 1993).



(a) abalone (b) estimation_of_obesity_levels (c) JapaneseVowels (d) mfeat-factors

(e) 3D_..._dataset (f) Bias_correction_r (g) Con...-10.5GHz(Urbinati) (h) MiamiHousing2016

(i) ada_agnostic (j) baseball (k) cmc (l) mfeat-fourier

(m) Facebook_..._Volume (n) Performance-Prediction (o) wine+quality (p) Diamonds

Figure 20: Visualization of the training dynamics fitting (validation curves of an MLP trained with default hyperparameters) on 16 unseen datasets. The datasets in the first two rows and the last two rows are DNN-friendly and Tree-friendly, respectively. The first and third rows represent classification tasks, while the second and fourth rows represent regression tasks.

Thus, although not our main focus, this framework can also guide adaptive training strategies while primarily serving as an analytical tool for understanding heterogeneity in tabular datasets.

Appendix D. More Details on Talent-Tiny

Figure 21 illustrates the distribution of Tree–DNN scores across the 45 datasets in Talent-tiny, a curated subset of Talent designed for fine-grained analysis. Each score measures the relative advantage of tree-based versus neural models on the same dataset. Datasets on

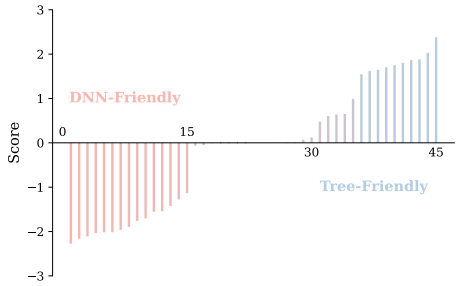


Figure 21: Distribution of Tree–DNN scores across the 45 datasets in Talent-tiny, which illustrates the balanced nature of this curated subset.

the left are DNN-friendly, where deep models such as RealMLP and MNCA perform better, while those on the right are tree-friendly, favoring ensembles like CatBoost. The overall distribution is approximately symmetric, reflecting that Talent-tiny maintains a balanced mixture of both categories. This design facilitates controlled evaluations of model behaviors and helps disentangle algorithmic factors from dataset bias, complementing the large-scale results reported in the main benchmark.

References

Samira Abnar, Mostafa Dehghani, Behnam Neyshabur, and Hanie Sedghi. Exploring the limits of large scale pre-training. In *ICLR*, 2022.

Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric reviews*, 29(5-6):594–621, 2010.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *KDD*, pages 2623–2631, 2019.

Ibrahim M. Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. Revisiting neural scaling laws in language and vision. In *NeurIPS*, pages 22300–22312, 2022.

Sercan Ö. Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *AAAI*, pages 6679–6687, 2021.

Sarkhan Badirli, Xuanqing Liu, Zhengming Xing, Avradeep Bhowmik, and Sathiya S. Keerthi. Gradient boosting neural networks: Grownet. *CoRR*, abs/2002.07971, 2020.

Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. Scarf: Self-supervised contrastive learning using random feature corruption. In *ICLR*, 2022.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *CoRR*, abs/2102.06701, 2021.

Daniel Beaglehole, David Holzmüller, Adityanarayanan Radhakrishnan, and Mikhail Belkin. xRFM: Accurate, scalable, and interpretable feature learning models for tabular data. *CoRR*, abs/2508.10053, 2025.

Christopher Bishop. *Pattern recognition and machine learning*. Springer, 2006.

David Bonet, Daniel Mas Montserrat, Xavier Giró i Nieto, and Alexander G. Ioannidis. Hyperfast: Instant classification for tabular data. In *AAAI*, pages 11114–11123, 2024.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions Neural Networks and Learning Systems*, 35(6):7499–7519, 2024.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Hao-Run Cai and Han-Jia Ye. Understanding the limits of deep tabular methods with temporal shift. In *ICML*, 2025.

William G. La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabrício Olivetti de França, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason H. Moore. Contemporary symbolic regression methods and their relative performance. In *NeurIPS Datasets and Benchmarks*, 2021.

Chun-Hao Chang, Rich Caruana, and Anna Goldenberg. NODE-GAM: neural generalized additive model for interpretable deep learning. In *ICLR*, 2022.

Wei-Lun Chao, Han-Jia Ye, De-Chuan Zhan, Mark E. Campbell, and Kilian Q. Weinberger. Revisiting meta-learning as supervised learning. *CoRR*, abs/2002.00573, 2020.

Jintai Chen, Kuanlun Liao, Yao Wan, Danny Z. Chen, and Jian Wu. Danets: Deep abstract networks for tabular data classification and regression. In *AAAI*, pages 3930–3938, 2022.

Jintai Chen, KuanLun Liao, Yanwen Fang, Danny Chen, and Jian Wu. Tabcaps: A capsule neural network for tabular data classification with bow routing. In *ICLR*, 2023a.

Jintai Chen, Jiahuan Yan, Qiyuan Chen, Danny Ziyi Chen, Jian Wu, and Jimeng Sun. Can a deep learning model be a sure bet for tabular prediction? In *KDD*, pages 288–296, 2024.

Kuan-Yu Chen, Ping-Han Chiang, Hsin-Rung Chou, Ting-Wei Chen, and Tien-Hao Chang. Trompt: Towards a better deep neural network for tabular data. In *ICML*, pages 4392–4434, 2023b.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *KDD*, pages 785–794, 2016.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide & deep learning for recommender systems. In *DLRS*, pages 7–10, 2016.

Corinna Cortes, Lawrence D. Jackel, Sara A. Solla, Vladimir Vapnik, and John S. Denker. Learning curves: Asymptotic values and rate of convergence. In *NIPS*, pages 327–334, 1993.

Manuel Fernández Delgado, Eva Cernadas, Senén Barro, and Dinani Gomes Amorim. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15(1):3133–3181, 2014.

Janez Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019.

Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy yong Sohn, Dimitris S. Papailiopoulos, and Kangwook Lee. LIFT: language-interfaced fine-tuning for non-language machine learning tasks. In *NeurIPS*, pages 11763–11784, 2022.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data. *CoRR*, abs/2506.16791, 2025.

Benjamin Feuer, Robin Tibor Schirrmeister, Valeriia Cherepanova, Chinmay Hegde, Frank Hutter, Micah Goldblum, Niv Cohen, and Colin White. Tunetables: Context optimization for scalable prior-data fitted networks. In *NeurIPS*, pages 83430–83464, 2024.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In *NIPS*, pages 2962–2970, 2015.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

Josh Gardner, Juan C. Perdomo, and Ludwig Schmidt. Large scale transfer learning for tabular data via language modeling. In *NeurIPS*, pages 45155–45205, 2024.

Anurag Garg, Muhammad Ali, Noah Hollmann, Lennart Purucker, Samuel Müller, and Frank Hutter. Real-tabpfn: Improving tabular foundation models via continued pre-training with real-world data. *CoRR*, abs/2507.03971, 2025.

Jacob Goldberger, Sam T. Roweis, Geoffrey E. Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *NIPS*, pages 513–520, 2004.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *NeurIPS*, pages 18932–18943, 2021.

Yury Gorishniy, Ivan Rubachev, and Artem Babenko. On embeddings for numerical features in tabular deep learning. In *NeurIPS*, pages 24991–25004, 2022.

Yury Gorishniy, Ivan Rubachev, Nikolay Kartashev, Daniil Shlenskii, Akim Kotelnikov, and Artem Babenko. Tabr: Tabular deep learning meets nearest neighbors. In *ICLR*, 2024.

Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. TabM: Advancing tabular deep learning with parameter-efficient ensembling. In *ICLR*, 2025.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *NeurIPS*, pages 507–520, 2022.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: A factorization-machine based neural network for CTR prediction. In *IJCAI*, pages 1725–1731, 2017.

Isabelle Guyon, Lisheng Sun-Hosoya, Marc Boullé, Hugo Jair Escalante, Sergio Escalera, Zhengying Liu, Damir Jajetic, Bisakha Ray, Mehreen Saeed, Michèle Sebag, et al. Analysis of the automl challenge series. *Automated Machine Learning*, 177:177–219, 2019.

Kam Hamidieh. Superconductivty Data. UCI Machine Learning Repository, 2018. DOI: https://doi.org/10.24432/C53P47.

Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark. In *NeurIPS*, pages 32142–32159, 2022.

John T. Hancock and Taghi M. Khoshgoftaar. Survey on categorical data for neural networks. *Journal of Big Data*, 7(1):28, 2020.

Lasse Hansen, Nabeel Seedat, Mihaela van der Schaar, and Andrija Petrovic. Reimagining synthetic tabular data generation through data-centric AI: A comprehensive benchmark. In *NeurIPS*, pages 33781–33823, 2023.

Md. Rafiul Hassan, Sadiq Al-Insaif, Muhammad Imtiaz Hossain, and Joarder Kamruzzaman. A machine learning approach for prediction of pregnancy outcome following IVF treatment. *Neural Computing and Applications*, 32(7):2283–2297, 2020.

Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer, 2009.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: few-shot classification of tabular data with large language models. In *AISTATS*, pages 5549–5581, 2023.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory F. Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *CoRR*, abs/1712.00409, 2017.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. In *ICLR*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

David Holzmüller, Léo Grinsztajn, and Ingo Steinwart. Better by default: Strong pre-tuned mlps and boosted trees on tabular data. In *NeurIPS*, pages 26577–26658, 2024.

Chenping Hou, Ruidong Fan, Ling-Li Zeng, and Dewen Hu. Adaptive feature selection with augmented attributes. *IEEE Transactions on pattern analysis and machine intelligence*, 45(8):9306–9324, 2023a.

Chenping Hou, Shilin Gu, Chao Xu, and Yuhua Qian. Incremental learning for simultaneous augmentation of feature and class. *IEEE Transactions on pattern analysis and machine intelligence*, 45(12):14789–14806, 2023b.

Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar S. Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *CoRR*, abs/2012.06678, 2020.

Alan Jeffares, Tennison Liu, Jonathan Crabbé, Fergus Imrie, and Mihaela van der Schaar. Tangos: Regularizing tabular neural networks through gradient orthogonalization and specialization. In *ICLR*, 2023.

Jun-Peng Jiang, Han-Jia Ye, Leye Wang, Yang Yang, Yuan Jiang, and De-Chuan Zhan. Tabular insights, visual impacts: Transferring expertise from tables to images. In *ICML*, pages 21988–22009, 2024a.

Jun-Peng Jiang, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, and Han-Jia Ye. Representation learning for tabular data: A comprehensive survey. *CoRR*, abs/2504.16109, 2025.

Xiangjian Jiang, Andrei Margeloiu, Nikola Simidjievski, and Mateja Jamnik. Protogate: Prototype-based neural networks with global-to-local feature selection for tabular biomedical data. In *ICML*, pages 21844–21878, 2024b.

Yu-Chin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. Field-aware factorization machines for CTR prediction. In *RecSys*, pages 43–50, 2016.

Arlind Kadra, Marius Lindauer, Frank Hutter, and Josif Grabocka. Well-tuned simple nets excel on tabular datasets. In *NeurIPS*, pages 23928–23941, 2021.

Liran Katzir, Gal Elidan, and Ran El-Yaniv. Net-dnf: Effective deep modeling of tabular data. In *ICLR*, 2021.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *NIPS*, pages 3146–3154, 2017.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *NIPS*, pages 971–980, 2017.

Ravin Kohli, Matthias Feurer, Katharina Eggensperger, Bernd Bischl, and Frank Hutter. Towards quantifying the effect of datasets for benchmarking: A look at tabular machine learning. In *ICLR Workshop*, 2024.

Si-Yang Liu and Han-Jia Ye. Tabpfn unleashed: A scalable and effective solution to tabular classification problems. In *ICML*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Hamidreza Kamkari, Alex Labach, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony L. Caterini. Tabdpt: Scaling tabular foundation models. *CoRR*, abs/2410.18164, 2024.

Núria Macià, Ester Bernadó-Mansilla, Albert Orriols-Puig, and Tin 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.

Sascha Marton, Stefan Lüdtke, Christian Bartelt, and Heiner Stuckenschmidt. GRANDE: gradient-based decision tree ensembles for tabular data. In *ICLR*, 2024.

Duncan C. McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C., Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? In *NeurIPS*, pages 76336–76369, 2023.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, 2012.

Andreas C. Mueller, Carlo Curino, and Raghu Ramakrishnan. Mothernet: Fast training and inference via hyper-network transformers. In *ICLR*, 2025.

Youssef Nader, Leon Sixt, and Tim Landgraf. DNNR: differential nearest neighbors regression. In *ICML*, pages 16296–16317, 2022.

Lennart J Nederstigt, Steven S Aanen, Damir Vandic, and Flavius Frasincar. Floppies: a framework for large-scale ontology population of product information from tabular data in e-commerce stores. *Decision Support Systems*, 59:296–311, 2014.

Inkit Padhi, Yair Schiff, Igor Melnyk, Mattia Rigotti, Youssef Mroueh, Pierre L. Dognin, Jerret Ross, Ravi Nair, and Erik Altman. Tabular transformers for modeling multivariate time series. In *ICASSP*, 2021.

Sergei Popov, Stanislav Morozov, and Artem Babenko. Neural oblivious decision ensembles for deep learning on tabular data. In *ICLR*, 2020.

Liudmila Ostroumova Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *NeurIPS*, pages 6639–6649, 2018.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabicl: A tabular foundation model for in-context learning on large data. In *ICML*, 2025.

Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Mechanism of feature learning in deep fully connected networks and kernel machines that recursively learn features. *CoRR*, abs/2212.13881v3, 2023.

Hafiz Tayyab Rauf, André Freitas, and Norman W. Paton. Tabledc: Deep clustering for tabular data. *CoRR*, abs/2405.17723, 2024.

Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: estimating the click-through rate for new ads. In *WWW*, pages 521–530, 2007.

Jonathan S. Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. In *ICLR*, 2020.

Ivan Rubachev, Artem Alekberov, Yury Gorishniy, and Artem Babenko. Revisiting pretraining objectives for tabular deep learning. *CoRR*, abs/2207.03208, 2022.

Ivan Rubachev, Nikolay Kartashev, Yury Gorishniy, and Artem Babenko. Tabred: A benchmark of tabular machine learning in-the-wild. In *ICLR*, 2025a.

Ivan Rubachev, Akim Kotelnikov, Nikolay Kartashev, and Artem Babenko. On finetuning tabular foundation models. *CoRR*, abs/2506.08982, 2025b.

Lisa M Schwartz, Steven Woloshin, and H Gilbert Welch. The drug facts box: providing consumers with simple tabular data on drug benefit and harm. *Medical Decision Making*, 27(5):655–662, 2007.

Tom Shenkar and Lior Wolf. Anomaly detection for tabular data with internal contrastive learning. In *ICLR*, 2022.

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

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Gowthami Somepalli, Avi Schwarzschild, Micah Goldblum, C. Bayan Bruss, and Tom Goldstein. SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training. In *NeurIPS Workshop*, 2022.

Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *CIKM*, pages 1161–1170, 2019.

Abdulhamit Subasi. Medical decision support system for diagnosis of neuromuscular disorders using dwt and fuzzy support vector machines. *Computers in Biology and Medicine*, 42(8): 806–815, 2012.

Jonathan Svirsky and Ofir Lindenbaum. Interpretable deep clustering for tabular data. In *ICML*, pages 47314–47330, 2024.

Richard Szeliski. *Computer Vision - Algorithms and Applications, Second Edition*. Springer, 2022.

Valentin Thomas, Junwei Ma, Rasa Hosseinzadeh, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony L. Caterini. Retrieval & fine-tuning for in-context tabular models. In *NeurIPS*, pages 108439–108467, 2024.

Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99(10):6567–6572, 2002.

Andrej Tschalzev, Sascha Marton, Stefan Lüdtke, Christian Bartelt, and Heiner Stuckenschmidt. A data-centric perspective on evaluating machine learning models for tabular data. In *NeurIPS*, pages 95896–95930, 2024.

Andrej Tschalzev, Lennart Purucker, Stefan Lüdtke, Frank Hutter, Christian Bartelt, and Heiner Stuckenschmidt. Unreflected use of tabular data repositories can undermine research quality. In *ICLR*, 2025.

Talip Ucar, Ehsan Hajiramezanali, and Lindsay Edwards. Subtab: Subsetting features of tabular data for self-supervised representation learning. In *NeurIPS*, pages 18853–18865, 2021.

Boris van Breugel and Mihaela van der Schaar. Position: Why tabular foundation models should be a research priority. In *ICML*, 2024.

Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

Mark Vero, Mislav Balunovic, and Martin T. Vechev. Cuts: Customizable tabular synthetic data generation. In *ICML*, pages 49408–49433, 2024.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016.

Michael Wainberg, Babak Alipanahi, and Brendan J. Frey. Are random forests truly the best classifiers? *Journal of Machine Learning Research*, 17:110:1–110:5, 2016.

Ruiyu Wang, Zifeng Wang, and Jimeng Sun. Unipredict: Large language models are universal tabular predictors. *CoRR*, abs/2310.03266, 2023.

Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *ADKDD*, pages 12:1–12:7, 2017.

Ruoxi Wang, Rakesh Shivanna, Derek Zhiyuan Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed H. Chi. DCN V2: improved deep & cross network and practical lessons for web-scale learning to rank systems. In *WWW*, pages 1785–1797, 2021.

Xumeng Wen, Han Zhang, Shun Zheng, Wei Xu, and Jiang Bian. From supervised to generative: A novel paradigm for tabular deep learning with large language models. In *SIGKDD*, pages 3323–3333, 2024.

Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *ICLR*, 2020.

Casper Wilstrup and Jaan Kasak. Symbolic regression outperforms other models for small data sets. *CoRR*, abs/2103.15147, 2021.

David H. Wolpert. The lack of A priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.

Jing Wu, Suiyao Chen, Qi Zhao, Renat Sergazinov, Chen Li, Shengjie Liu, Chongchao Zhao, Tianpei Xie, Hanqing Guo, Cheng Ji, Daniel Cociorva, and Hakan Brunzell. Switchtab: Switched autoencoders are effective tabular learners. In *AAAI*, pages 15924–15933, 2024.

Chao Xu, Hong Tao, Jing Zhang, Dewen Hu, and Chenping Hou. Label distribution changing learning with sample space expanding. *Journal of Machine Learning Research*, 24:36:1–36:48, 2023.

Chenwei Xu, Yu-Chao Huang, Jerry Yao-Chieh Hu, Weijian Li, Ammar Gilani, Hsi-Sheng Goan, and Han Liu. Bishop: Bi-directional cellular learning for tabular data with generalized sparse modern hopfield model. In *ICML*, pages 55048–55075, 2024.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional GAN. In *NeurIPS*, pages 7333–7343, 2019.

Scott Yak, Yihe Dong, Javier Gonzalvo, and Sercan Ö. Arık. Ingestables: Scalable and efficient training of llm-enabled tabular foundation models. In *Table Representation Learning Workshop at NeurIPS 2023*, 2023.

Jiahuan Yan, Jintai Chen, Qianxing Wang, Danny Ziyi Chen, and Jian Wu. Team up gbdts and dnns: Advancing efficient and effective tabular prediction with tree-hybrid mlps. In *KDD*, pages 3679–3689, 2024a.

Jiahuan Yan, Bo Zheng, Hongxia Xu, Yiheng Zhu, Danny Z. Chen, Jimeng Sun, Jian Wu, and Jintai Chen. Making pre-trained language models great on tabular prediction. In *ICLR*, 2024b.

Ling Yan, Wu-Jun Li, Gui-Rong Xue, and Dingyi Han. Coupled group lasso for web-scale CTR prediction in display advertising. In *ICML*, pages 802–810, 2014.

Chao Ye, Guoshan Lu, Haobo Wang, Liyao Li, Sai Wu, Gang Chen, and Junbo Zhao. Towards cross-table masked pretraining for web data mining. In *WWW*, pages 4449–4459, 2024a.

Han-Jia Ye, De-Chuan Zhan, Yuan Jiang, and Zhi-Hua Zhou. Heterogeneous few-shot model rectification with semantic mapping. *IEEE Transactions on pattern analysis and machine intelligence*, 43(11):3878–3891, 2021.

Han-Jia Ye, Si-Yang Liu, and Wei-Lun Chao. A closer look at tabpfn v2: Understanding its strengths and extending its capabilities. *CoRR*, abs/2502.17361, 2025a.

Han-Jia Ye, Huai-Hong Yin, De-Chuan Zhan, and Wei-Lun Chao. Revisiting nearest neighbor for tabular data: A deep tabular baseline two decades later. In *ICLR*, 2025b.

Hangting Ye, Wei Fan, Xiaozhuang Song, Shun Zheng, He Zhao, Dan dan Guo, and Yi Chang. Ptarl: Prototype-based tabular representation learning via space calibration. In *ICLR*, 2024b.

Jiaxin Yin, Yuanyuan Qiao, Zitang Zhou, Xiangchao Wang, and Jie Yang. MCM: masked cell modeling for anomaly detection in tabular data. In *ICLR*, 2024.

Yuchen Zeng, Tuan Dinh, Wonjun Kang, and Andreas C. Mueller. Tabflex: Scaling tabular learning to millions with linear attention. In *ICML*, 2025.

Weinan Zhang, Tianming Du, and Jun Wang. Deep learning over multi-field categorical data - - A case study on user response prediction. In *ECIR*, pages 45–57, 2016.

Xingxuan Zhang, Gang Ren, Han Yu, Hao Yuan, Hui Wang, Jiansheng Li, Jiayun Wu, Lang Mo, Li Mao, Mingchao Hao, Ningbo Dai, Renzhe Xu, Shuyang Li, Tianyang Zhang, Yue He, Yuanrui Wang, Yunjia Zhang, Zijing Xu, Dongzhe Li, Fang Gao, Hao Zou, Jiandong Liu, Jiashuo Liu, Jiawei Xu, Kaijie Cheng, Kehan Li, Linjun Zhou, Qing Li, Shaohua Fan, Xiaoyu Lin, Xinyan Han, Xuanyue Li, Yan Lu, Yuan Xue, Yuanyuan Jiang, Zimu Wang, Zhenlei Wang, and Peng Cui. Limix: Unleashing structured-data modeling capability for generalist intelligence. *CoRR*, abs/2509.03505, 2025.

Qi-Le Zhou, Han-Jia Ye, Leye Wang, and De-Chuan Zhan. Unlocking the transferability of tokens in deep models for tabular data. *CoRR*, abs/2310.15149, 2023.

Zhi-Hua Zhou. Learnability with time-sharing computational resource concerns. *National Science Review*, 11(10):nwae204, 2024.