IN-CONTEXT DATA DISTILLATION WITH TABPFN

Junwei Ma, Valentin Thomas, Guangwei Yu, Anthony Caterini
Layer 6 AI
{jeremy,valentin.t,guang,anthony}@layer6.ai

ABSTRACT

Foundation models have revolutionized tasks in computer vision and natural language processing. However, in the realm of tabular data, tree-based models like XGBoost continue to dominate. TabPFN, a transformer model tailored for tabular data, mirrors recent foundation models in its exceptional in-context learning capability, being competitive with XGBoost's performance without the need for task-specific training or hyperparameter tuning. Despite its promise, TabPFN's applicability is hindered by its data size constraint, limiting its use in real-world scenarios. To address this, we present in-context data distillation (ICD), a novel methodology that effectively eliminates these constraints by optimizing TabPFN's context. ICD efficiently enables TabPFN to handle significantly larger datasets with a fixed memory budget, improving TabPFN's quadratic memory complexity but at the cost of a linear number of tuning steps. Notably, TabPFN, enhanced with ICD, demonstrates very strong performance against established tree-based models and modern deep learning methods on 48 large tabular datasets from OpenML.

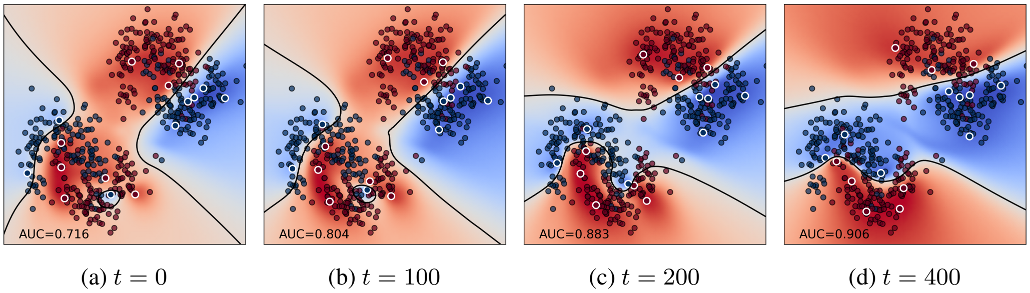
1 INTRODUCTION

Recent advancements in foundation models (Bommasani et al., 2021), including GPT-3 (Brown et al., 2020), CLIP (Radford et al., 2021), and diffusion models (Ho et al., 2020), have demonstrated remarkable proficiency in tasks related to computer vision and natural language processing, marking a significant leap forward in the field of machine learning. However, despite its pivotal role across diverse industries (Tang et al., 2020; Benjelloun et al., 2020; Sattarov et al., 2023), tabular data has long been dominated by tree-based models like XGBoost (Chen & Guestrin, 2016). The predominance of these models is attributed to several factors. Firstly, unlike their deep learning counterparts, tree-based models require minimal hyper-parameter tuning, simplifying their use and boosting their adoption. Secondly, while deep learning models owe their success to pre-training and transfer learning, this advantage is largely absent in tabular data due to its diverse nature (Borisov et al., 2022) and lack of pre-training datasets such as ImageNet (Deng et al., 2009).

Recently, the introduction of TabPFN (Hollmann et al., 2023) and its variants (Ma et al., 2023; Dooley et al., 2023) have marked a significant advance in this domain. Trained on a vast array of synthetic datasets, TabPFN harnesses the transfer learning capabilities of deep learning models while matching the performance of tree-based models on smaller datasets, without the need for extensive hyper-parameter tuning. However, due to its reliance on the Transformer (Vaswani et al., 2017) architecture, it suffers from quadratic memory complexity scaling, which renders it impractical for large real-world datasets.

The concept of prompt tuning, prevalent in the LLM literature (Lester et al., 2021; Li & Liang, 2021), posits that adapting a pre-trained model through prompt modification, rather than adjusting the model weights, can be more effective. This is particularly relevant for TabPFN, whose strength lies in in-context meta-learning across a multitude of datasets. As the context used in TabPFN is the training set, this method can be also be interpreted through the lens of dataset distillation (Wang et al., 2018). As shown in Figure 1, our method is able to optimize the distilled datapoints in order to refine TabPFN's decision boundaries and improve its classification accuracy.

Building on this idea, we introduce *In-Context Distillation* (ICD). This novel approach performs backpropagation on the context tokens, allowing us to extend the original TabPFN to larger datasets. Moreover, TabPFN with ICD demonstrates superior performance over both the original TabPFN



(a) $t = 0$ (b) $t = 100$ (c) $t = 200$ (d) $t = 400$

Figure 1: Evolution of the distilled datapoints (circles with white borders, only 8 per class) and our decision boundary on a simple two double moons 2d dataset. The black line represents the decision boundary $p(y|x, \mathcal{D}_{\text{dist}}) = 0.5$ of TabPFN conditioned on the 16 distilled points. The shaded red (respectively blue) regions represent which class the classifier will assign to datapoints in that region. The circles with a black edge color are the test dataset. The distilled points are initialized on random training points and over time they move around in order to improve the classifier.

and other competitive models across 48 large OpenML benchmark datasets (Bischl et al., 2021; McElfresh et al., 2023).

2 Background and Related Work

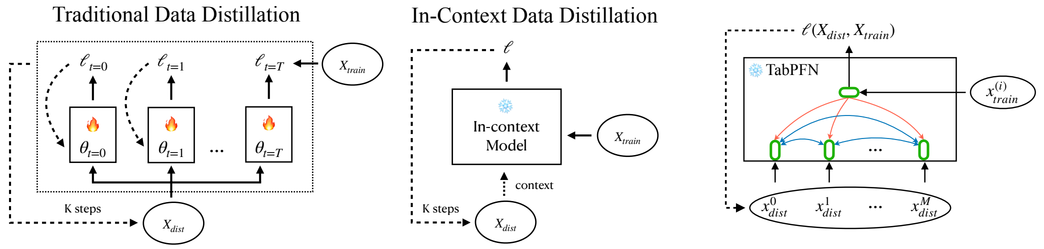
TabPFN (Hollmann et al., 2023) is a novel transformer-based model developed for classifying tabular data, differing from traditional methods (Houthooft et al., 2016; Somepalli et al., 2021; Arik & Pfister, 2021) that require a unique model for each task. Instead, TabPFN adopts a different approach, commonly found in vision and text foundation models, by training on a diverse array of tasks (Müller et al., 2022) with a shared set of weight parameters θ applicable across all tasks. In order to perform classification on a new dataset, one uses the training data as *context* and the new data to classify as a *query*. While this allows making predictions extremely fast, only at the cost of inference, there are inherent limits to the approach. TabPFN suffers from the $\mathcal{O}(n^2)$ memory scaling of transformers, restricting its applicability to only datasets with less than 1000 datapoints and 100 features. Attempts at context selection through sketching methods have yet to yield success (Feuer et al., 2023).

Dataset distillation is research area pioneered by Wang et al. (2018), aiming at learning a small set of training examples – called the *distilled set* $\mathcal{D}_{\text{dist}}$ – such that a classifier trained on only those few examples can achieve a performance comparable to a classifier trained on the original dataset $\mathcal{D}_{\text{train}}$. While there have been advances simplifying the approach (Zhao et al., 2021), the main strategy is to optimize $\mathcal{D}_{\text{dist}}$ directly by backpropagation through the entire neural network training procedure.

Prompt-Tuning (Lester et al., 2021; Li & Liang, 2021) has recently emerged as a new field in machine learning, which specifically addresses efficient fine-tuning of large foundational models. Unlike other parameter-efficient fine-tuning methods (Xu et al., 2023), prompt-tuning significantly reduces the number of trainable parameters. This efficiency becomes particularly apparent in models like TabPFN as we will see in section 3. Instead of using prompts as context inputs, TabPFN uses the real data. Consequently, prompt-tuning in the context of TabPFN can also be seen as data distillation, where the optimization is done on the input data itself, rather than a pure prompt-based approach.

3 In-Context Distillation

As mentioned in section 2, TabPFN is limited to small tabular datasets. This is because the training examples are used as *context* to classify new examples. It can be natural to consider data distillation as a way to resolve that issue by compressing a large dataset into one that can fit in the context of TabPFN. However, naively applying data distillation techniques such as Wang et al. (2018) would require partially retraining the whole TabPFN model several times, which can be very expensive. In this paper, we propose a simpler alternative: we view the distilled dataset as the context, which is learned to maximize the likelihood of the training data.



(a) (b)

Figure 2: (a) Comparison between traditional and in-context data distillation. **Left**: Traditional data distillation methods usually consist of nested optimization loops, where the inner loop optimizes model θ for T steps, then an outer loop optimizes through the inner optimization process to update X_{dist} for K steps. **Right**: In-context data distillation only requires a single loop of optimization on X_{dist} for K steps, eliminating the need to optimize through the inner optimization process. (b) Detailed architecture diagram of applying ICD on TabPFN. Dashed lines indicate gradient flow. Blue and red arrows represent attentions within TabPFN.

More precisely, we consider the following setting: we have access to a classifier denoted by p_θ trained on a variety of datasets. Given a new dataset $\mathcal{D}_{\text{train}}$ and a query point x to classify, its class probability is computed as $p_\theta(y|x, \mathcal{D}_{\text{train}})$, where $\mathcal{D}_{\text{train}}$ can be understood as the "prompt". In-Context Distillation (ICD) – similarly to prompt-tuning (Lester et al., 2021) – optimizes the likelihood of the real data given the distilled one:

$$\mathcal{L}(\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{dist}}) \triangleq -\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{train}}}[\log p_\theta(y|x, \mathcal{D}_{\text{dist}})].$$

By minimizing $\mathcal{L}(\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{dist}})$, we obtain a dataset of size $|\mathcal{D}_{\text{dist}}| \ll |\mathcal{D}_{\text{train}}|$ such that the training data has a high likelihood. Then, given a new point x to classify, we use $\mathcal{D}_{\text{dist}}$ as the context and predict the label $\arg\max_y p_\theta(y|x, \mathcal{D}_{\text{dist}})$. While for small datasets we could directly compute $\arg\max_y p_\theta(y|x, \mathcal{D}_{\text{train}})$, for larger datasets $\mathcal{D}_{\text{train}}$ is too large to be used as context.

While this objective function $\mathcal{L}(\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{dist}})$ is similar to previous works in dataset distillation, we can directly compute the gradient $\nabla_{\mathcal{D}_{\text{dist}}}\mathcal{L}(\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{dist}})$ without having to retrain θ because TabPFN's inference is conceptually similar to the whole training process of traditional methods.

Indeed, TabPFN is a foundation model for tabular data and it has been trained on a large number of tasks. As such, TabPFN is able to use the distilled data efficiently, even if it does not the follow the original data distribution. Moreover, traditional models that do not perform in-context learning have to be (at least partially) trained on the distilled data in order to generate useful gradients with respect to $\mathcal{D}_{\text{dist}}$.

Comparison with dataset distillation (Wang et al., 2018) In comparison, traditional models do not use a context. These do not generalize to new distributions or distribution shifts well and as such need to be trained for a number of steps on each data distribution. For these reasons, the distillation objective is of the form

$$\begin{aligned} &\arg\min_{\mathcal{D}_{\text{dist}}} \mathcal{L}(\mathcal{D}_{\text{dist}}) \triangleq -\mathbb{E}_{(x,y)\sim\mathcal{D}}[\log(y|x, \theta_T(\mathcal{D}_{\text{dist}}))], \\ &\text{s.t} \quad \theta_{t+1}(\mathcal{D}_{\text{dist}}) = \theta_t(\mathcal{D}_{\text{dist}}) - \eta\nabla_\theta\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{dist}}}\ell(y|x, \theta_t(\mathcal{D}_{\text{dist}})). \end{aligned}$$

As we can see, the dependence on $\mathcal{D}_{\text{dist}}$ is now implicit through θ and as such updating the distilled dataset for just one step requires backpropagation through T steps of training as shown in Figure 2a. Note that several lines of work have proposed more efficient but approximate methods such as matching gradient statistics (Zhao et al., 2021).

However, **distillation methods usually lower the performance of downstream models**. In contrast, because our motivation is to obtain the most information about $\mathcal{D}_{\text{train}}$ and fit it into the context, **our method actually increases the performance of the base model** compared to when it is trained on $|\mathcal{D}_{\text{dist}}|$ random points as is usually done (McElfresh et al., 2023; Feuer et al., 2023).

	XGBoost (Tuned)	TabPFN-ICD	XGBoost	TabPFN	TabNet	RandomForest	SAINT	MLP	VIME
Median AUC	**0.969**	0.967	0.953	0.951	0.939	0.928	0.878	0.864	0.824
Median f1	**0.921**	0.899	0.893	0.847	0.887	0.862	0.853	0.840	0.760
Median accuracy	**0.923**	0.902	0.894	0.844	0.887	0.866	0.855	0.841	0.762

Table 1: Median accuracy, AUC, and F1 score for **TabPFN-ICD** and a variety of baselines calculated over the 48 datasets. Highest performance is bolded and the second-highest is underlined in each row.

4 Experiments

Experimental Setup Our experiments and analyses are based on the Tabzilla benchmark (McElfresh et al., 2023), which comprises 176 datasets sourced from OpenML (Bischl et al., 2021), a widely recognized repository for tabular data. We use 48 out of all the available datasets that have between 2,000 to 200,000 instances and do not contain NaNs, the specifics of which are detailed in appendix A.1. Each dataset is partitioned into train, validation, and test sets with a 80:10:10 split. For all methods, we used for early stopping on validation AUC to prevent overfitting. The performance of the model was evaluated based on the best-validated model's score on the test set, providing a reliable measure of its generalization capability. To ensure a fair and comprehensive comparison of the models, we report on three performance metrics: accuracy, F1 score, and Area Under the ROC Curve (AUC). In particular, AUC and F1 are known to give a more complete view of the performance of the model, especially in the case of imbalanced datasets (which most are, see Table 2).

For **TabPFN-ICD** we use a constant learning rate of 10^{-2} with the Adam optimizer (Kingma & Ba, 2015). We always use $|\mathcal{D}_{\text{dist}}| = 1{,}000$. In this preliminary work, we only learn X_{dist} and not y_{dist}. We choose to partition $\mathcal{D}_{\text{dist}}$ into sets of equal size for each class. For TabPFN, we sample at random $1{,}000$ points in each dataset and use them for classification on new points.

Results We report the median of each metric aggregated over the 48 datasets in Table 1. While currently our method does not beat a tuned XGBoost, its rank is always second in each metric and it significantly outperforms its base version TabPFN, and XGBoost with default hyperparameters. Furthermore, contrary to McElfresh et al. (2023), we only used datasets with size larger than 2,000 samples, resulting in TabPFN having significantly lower performance, even compared to XGBoost with default parameters.

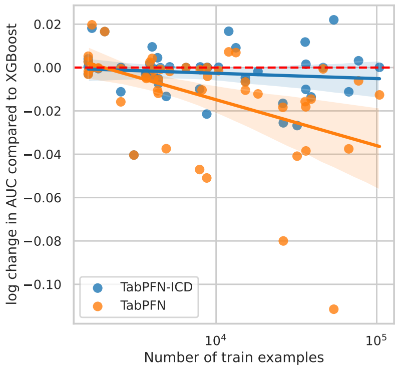


Figure 3: Log change in AUC that TabPFN-ICD and TabPFN have over XGB as a function of training set size.

In Figure 3, we see that TabPFN's performance drops as a function of the training set size compared to XGBoost ($p_{\text{value}} < 10^{-3}$ for rejecting the null hypothesis "$\mathcal{H}_0$: slope is 0"). This confirms our original motivation and echoes past work (Feuer et al., 2023) that for many large datasets, subsampling a small subset of it is not enough to guarantee strong performance. With TabPFN-ICD, we observe that the trend in performance follows the one of XGBoost ($p_{\text{value}} \approx 0.45$), confirming that we are able to distill information about the dataset into a small number of samples.

5 Discussion and conclusion

We have proposed a straightforward yet innovative method for in-context distillation, drawing inspiration from both dataset distillation and prompt tuning techniques. This approach enables us to significantly enhance the scalability of TabPFN, allowing it to handle very large datasets while maintaining competitive performance against state-of-the-art algorithms. We believe that our method can open up new research avenues for data distillation and in-context learning for foundation models.

References

Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 6679–6687, 2021.

Omar Benjelloun, Shiyu Chen, and Natasha Noy. Google dataset search by the numbers. In *The Semantic Web – ISWC 2020*, pp. 667–682, 2020.

Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Pieter Gijsbers, Frank Hutter, Michel Lang, Rafael Gomes Mantovani, Jan van Rijn, and Joaquin Vanschoren. OpenML benchmarking suites. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SigKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Samuel Dooley, Gurnoor Singh Khurana, Chirag Mohapatra, Siddartha Naidu, and Colin White. Forecastpfn: Synthetically-trained zero-shot forecasting. *arXiv preprint arXiv:2311.01933*, 2023.

Benjamin Feuer, Chinmay Hegde, and Niv Cohen. Scaling tabpfn: Sketching and feature selection for tabular prior-data fitted networks. *arXiv preprint arXiv: 2311.10609*, 2023. URL `https://arxiv.org/abs/2311.10609v1`.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations*, 2023.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *Conference on Empirical Methods in Natural Language Processing*, 2021. doi: 10.18653/v1/2021.emnlp-main.243.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *Annual Meeting of the Association for Computational Linguistics*, 2021. doi: 10.18653/v1/2021.acl-long.353.

Junwei Ma, Apoorv Dankar, George Stein, Guangwei Yu, and Anthony Caterini. Tabpfgen–tabular data generation with tabpfn. In *NeurIPS 2023 Second Table Representation Learning Workshop*, 2023.

Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Ganesh Ramakrishnan, Micah Goldblum, Colin White, et al. When do neural nets outperform boosted trees on tabular data? *arXiv preprint arXiv:2305.02997*, 2023.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations*, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Timur Sattarov, Marco Schreyer, and Damian Borth. Findiff: Diffusion models for financial tabular data generation. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pp. 64–72, 2023.

Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C Bayan Bruss, and Tom Goldstein. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*, 2021.

Qi Tang, Guoen Xia, Xianquan Zhang, and Feng Long. A customer churn prediction model based on XGBoost and MLP. In *2020 International Conference on Computer Engineering and Application (ICCEA)*, pp. 608–612, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation. *arXiv preprint arXiv: 1811.10959*, 2018. URL `https://arxiv.org/abs/1811.10959v3`.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*, 2023.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=mSAKhLYLSsl`.

A APPENDIX

A.1 DATASET DETAILS

We chose 48 large datasets from OpenML (Bischl et al., 2021) to test the generalization performance of TabPFN-ICD against other state-of-the-art methods. The details of these datasets are listed in Table 2

Table 2: 48 large datasets from OpenML

	did	# instances	# train	# feat	# classes	# cat	class imbalance ratio
Amazon employee access	34539	32769	26215	9	2	9	16.258065
Click prediction small	190408	39948	31958	11	2	6	4.937941
GesturePhaseSegmentationProcessed	14969	9873	7897	32	5	0	2.957393
JapaneseVowels	3510	9961	7968	14	9	0	2.070513
MagicTelescope	3954	19020	15216	10	2	0	1.843049
MiniBooNE	168335	130064	104050	50	2	0	2.563478
PhishingWebsites	14952	11055	8843	30	2	30	1.257019
Satellite	167211	5100	4080	36	2	0	65.885246
ada agnostic	3896	4562	3648	48	2	0	3.035398
adult-census	3953	32561	26048	14	2	8	3.153061
adult	7592	48842	39072	14	2	8	3.179271
artificial-characters	14964	10218	8174	7	10	0	2.362500
bank-marketing	14965	45211	36168	16	2	9	7.546314
cardiotocography	9979	2126	1700	35	10	0	10.767442
churn	167141	5000	4000	20	2	4	6.054674
connect-4	146195	67557	54045	42	3	42	6.896104
eeg-eye-state	14951	14980	11984	14	2	0	1.227923
electricity	219	45312	36248	8	2	1	1.355449
elevators	3711	16599	13279	18	2	0	2.235624
eye movements	3897	10936	8748	27	3	3	1.484321
first-order-theorem-proving	9985	6118	4894	51	6	0	5.238462
house 16H	3686	22784	18226	16	2	0	2.378940
jannis	168330	83733	66985	54	4	0	22.845070
jungle chess 2pcs raw endgame complete	167119	44819	35855	6	3	0	5.317959
kc1	3917	2109	1687	21	2	0	5.438931
kr-vs-kp	3	3196	2556	36	2	36	1.093366
magic	146206	19020	15216	10	2	0	1.843049
mfeat-fourier	14	2000	1600	76	10	0	1.000000
mfeat-karhunen	16	2000	1600	64	10	0	1.000000
mfeat-morphological	18	2000	1600	6	10	0	1.000000
mfeat-zernike	22	2000	1600	47	10	0	1.000000
mushroom	24	8124	6498	22	2	22	1.074713
numerai28.6	167120	96320	77056	21	2	0	1.020982
nursery	9892	12958	10366	8	4	8	13.190840
optdigits	28	5620	4496	64	10	0	1.029279
ozone-level-8hr	9978	2534	2026	72	2	0	14.828125
page-blocks	30	5473	4377	10	5	0	178.590909
pendigits	32	10992	8792	16	10	0	1.086595
phoneme	9952	5404	4322	5	2	0	2.408517
pollen	3735	3848	3078	5	2	0	1.001300
satimage	2074	6430	5144	36	6	0	2.454910
segment	146822	2310	1848	16	7	0	1.000000
shuttle	146212	58000	46400	9	7	0	4558.500000
spambase	43	4601	3680	57	2	0	1.539683
splice	45	3190	2552	60	3	60	2.161501
sylvine	168912	5124	4098	20	2	0	1.000977
wall-robot-navigation	9960	5456	4364	24	4	0	6.729008
wilt	146820	4839	3871	5	2	0	17.521531

A.2 Baseline Details

For the baselines, we leverage the TabZilla GitHub repository, accessible at `https://github.com/naszilla/tabzilla`. We also use a training batch size of 128 over 100 epochs for all models.

For XGBoost (Tuned), we tuned the hyperparameters over 5 rounds, which led to the following best hyperparameters: max depth $= 11$, $\alpha = 0.00023870306386021943$, $\eta = 0.15557408249528354$ and $\lambda = 0.6046093787689293$

A.3 Additional Results

Table 3: AUC per dataset

	RandomForest	TabPFN-ICD	MLP	saint	TabNet	TabPFN	vime	XGBoost	XGBoost (Tuned)
dataset									
Amazon-employee-access-34539	0.732	0.759	0.626	0.815	0.721	0.670	0.500	0.805	0.866
Click-prediction-small-190408	0.668	0.639	0.584	0.622	0.599	0.619	0.571	0.680	0.652
GesturePhaseSegmentationProcessed-14969	0.788	0.845	0.711	0.750	0.778	0.775	0.657	0.864	0.902
JapaneseVowels-3510	0.980	1.000	0.970	1.000	1.000	0.997	0.993	0.999	1.000
MagicTelescope-3954	0.897	0.924	0.889	0.875	0.938	0.923	0.889	0.937	0.939
MiniBooNE-168335	0.958	0.979	0.949	0.962	0.939	0.951	0.911	0.979	0.984
PhishingWebsites-14952	0.979	0.992	0.989	0.992	0.994	0.983	0.978	0.992	0.995
Satellite-167211	0.993	0.993	0.988	0.771	0.908	0.991	0.827	0.994	0.988
ada-agnostic-3896	0.882	0.888	0.876	0.818	0.879	0.885	0.818	0.894	0.883
adult-census-3953	0.899	0.889	0.658	0.865	0.906	0.885	0.616	0.923	0.914
adult-7592	0.902	0.900	0.719	0.863	0.915	0.898	0.563	0.929	0.927
artificial-characters-14964	0.915	0.972	0.799	0.955	0.962	0.951	0.930	0.973	0.995
bank-marketing-14965	0.900	0.906	0.885	0.807	0.933	0.889	0.849	0.927	0.927
cardiotocography-9979	1.000	1.000	0.760	1.000	1.000	1.000	0.512	1.000	1.000
churn-167141	0.859	0.876	0.725	0.868	0.899	0.866	0.679	0.857	0.857
connect-4-146195	0.717	0.875	0.752	0.873	0.878	0.643	0.502	0.832	0.909
eeg-eye-state-14951	0.856	0.990	0.595	0.477	0.685	0.969	0.611	0.953	0.987
electricity-219	0.864	0.937	0.888	0.878	0.900	0.854	0.822	0.933	0.981
elevators-3711	0.841	0.936	0.762	0.652	0.935	0.932	0.718	0.917	0.928
eye-movements-3897	0.747	0.786	0.665	0.761	0.777	0.735	0.682	0.826	0.893
first-order-theorem-proving-9985	0.754	0.781	0.641	0.783	0.717	0.739	0.519	0.806	0.810
house-16H-3686	0.924	0.947	0.851	0.788	0.918	0.924	0.782	0.951	0.951
jannis-168330	0.771	0.819	0.774	0.816	0.849	0.771	0.739	0.840	0.848
jungle-chess-2pcs-raw-endgame-complete-167119	0.872	0.979	0.849	0.995	0.978	0.919	0.844	0.952	0.970
kc1-3917	0.810	0.823	0.768	0.820	0.806	0.826	0.791	0.789	0.825
kr-vs-kp-3	0.994	1.000	1.000	1.000	0.996	0.999	0.980	1.000	1.000
magic-146206	0.897	0.927	0.895	0.879	0.936	0.915	0.884	0.937	0.939
mfeat-fourier-14	0.975	0.976	0.945	0.989	0.973	0.976	0.930	0.983	0.980
mfeat-karhunen-16	0.992	0.998	0.998	0.997	0.998	0.993	0.999	0.997	0.997
mfeat-morphological-18	0.942	0.951	0.483	0.933	0.954	0.951	0.459	0.945	0.939
mfeat-zernike-22	0.963	0.978	0.975	0.902	0.978	0.980	0.974	0.969	0.964
mushroom-24	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
numerai28.6-167120	0.525	0.526	0.526	0.508	0.511	0.515	0.521	0.522	0.508
nursery-9892	0.943	1.000	0.826	0.995	0.998	0.996	0.999	1.000	1.000
optdigits-28	0.996	0.999	1.000	1.000	0.998	0.998	0.999	1.000	1.000
ozone-level-8hr-9978	0.933	0.945	0.931	0.938	0.914	0.945	0.832	0.910	0.920
page-blocks-30	0.962	0.967	0.643	0.960	0.951	0.971	0.606	0.984	0.968
pendigits-32	0.996	1.000	0.965	0.997	1.000	1.000	0.999	1.000	1.000
phoneme-9952	0.909	0.954	0.923	0.938	0.931	0.922	0.919	0.945	0.963
pollen-3735	0.489	0.462	0.496	0.548	0.499	0.462	0.504	0.507	0.480
satimage-2074	0.975	0.988	0.976	0.985	0.985	0.984	0.972	0.988	0.990
segment-146822	0.987	0.993	0.978	0.993	0.991	0.994	0.977	0.994	0.994
shuttle-146212	1.000	1.000	0.557	0.859	0.977	0.999	0.462	1.000	1.000
spambase-43	0.963	0.978	0.971	0.967	0.971	0.971	0.954	0.983	0.984
splice-45	0.989	0.967	0.955	0.995	0.981	0.957	0.919	0.992	0.994
sylvine-168912	0.958	0.967	0.916	0.505	0.953	0.964	0.870	0.976	0.980
wall-robot-navigation-9960	0.996	0.988	0.734	0.992	0.996	0.973	0.811	1.000	1.000
wilt-146820	0.982	0.994	0.893	0.661	0.993	0.996	0.710	0.990	0.989
median	0.928	0.967	0.864	0.878	0.939	0.951	0.824	0.953	0.969

Table 4: F1 scores per dataset

	RandomForest	TabPFN-ICD	MLP	saint	TabNet	TabPFN	vime	XGBoost	XGBoost (Tuned)
dataset									
Amazon-employee-access-34539	0.943	0.920	0.942	0.947	0.943	0.736	0.942	0.943	0.946
Click-prediction-small-190408	0.832	0.765	0.832	0.833	0.832	0.603	0.832	0.835	0.831
GesturePhaseSegmentationProcessed-14969	0.425	0.640	0.410	0.432	0.464	0.463	0.332	0.602	0.678
JapaneseVowels-3510	0.821	0.994	0.823	0.977	0.983	0.932	0.918	0.963	0.980
MagicTelescope-3954	0.842	0.872	0.841	0.828	0.879	0.849	0.700	0.876	0.887
MiniBooNE-168335	0.895	0.934	0.886	0.909	0.818	0.882	0.811	0.933	0.941
PhishingWebsites-14952	0.924	0.962	0.944	0.948	0.959	0.934	0.846	0.947	0.967
Satellite-167211	0.990	0.986	0.990	0.984	0.990	0.990	0.986	0.990	0.990
ada-agnostic-3896	0.845	0.827	0.840	0.786	0.832	0.796	0.751	0.856	0.840
adult-census-3953	0.842	0.829	0.802	0.824	0.850	0.806	0.759	0.869	0.861
adult-7592	0.845	0.843	0.805	0.816	0.857	0.809	0.761	0.871	0.876
artificial-characters-14964	0.476	0.744	0.481	0.624	0.639	0.599	0.610	0.759	0.915
bank-marketing-14965	0.892	0.892	0.892	0.848	0.910	0.845	0.883	0.906	0.906
cardiotocography-9979	0.983	1.000	0.401	1.000	1.000	1.000	0.117	1.000	1.000
churn-167141	0.894	0.931	0.872	0.912	0.954	0.874	0.860	0.950	0.948
connect-4-146195	0.530	0.810	0.746	0.805	0.813	0.521	0.523	0.729	0.841
eeg-eye-state-14951	0.778	0.954	0.578	0.551	0.632	0.909	0.545	0.879	0.944
electricity-219	0.774	0.862	0.807	0.803	0.819	0.772	0.602	0.853	0.931
elevators-3711	0.817	0.897	0.769	0.707	0.891	0.875	0.691	0.872	0.881
eye-movements-3897	0.520	0.599	0.456	0.545	0.556	0.491	0.467	0.630	0.724
first-order-theorem-proving-9985	0.429	0.542	0.472	0.571	0.437	0.477	0.247	0.567	0.586
house-16H-3686	0.862	0.887	0.794	0.764	0.846	0.849	0.706	0.887	0.891
jannis-168330	0.630	0.681	0.668	0.681	0.703	0.550	0.637	0.694	0.714
jungle-chess-2pcs-raw-endgame-complete-167119	0.708	0.879	0.797	0.952	0.883	0.783	0.667	0.841	0.847
kc1-3917	0.872	0.871	0.848	0.848	0.867	0.796	0.848	0.863	0.858
kr-vs-kp-3	0.941	0.991	0.991	0.994	0.988	0.988	0.828	0.994	0.994
magic-146206	0.842	0.880	0.840	0.826	0.880	0.843	0.707	0.876	0.887
mfeat-fourier-14	0.745	0.793	0.663	0.858	0.790	0.752	0.584	0.861	0.837
mfeat-karhunen-16	0.925	0.925	0.955	0.954	0.945	0.895	0.970	0.933	0.944
mfeat-morphological-18	0.641	0.701	0.018	0.633	0.692	0.686	0.018	0.655	0.639
mfeat-zernike-22	0.704	0.802	0.786	0.469	0.783	0.803	0.776	0.757	0.779
mushroom-24	0.993	1.000	1.000	1.000	1.000	1.000	0.985	1.000	1.000
numerai28.6-167120	0.516	0.515	0.518	0.504	0.513	0.510	0.512	0.519	0.506
nursery-9892	0.863	0.999	0.948	0.921	0.956	0.944	0.984	0.997	0.997
optdigits-28	0.955	0.991	0.964	0.970	0.979	0.941	0.974	0.975	0.973
ozone-level-8hr-9978	0.945	0.939	0.937	0.937	0.945	0.922	0.937	0.937	0.941
page-blocks-30	0.967	0.962	0.941	0.961	0.942	0.934	0.912	0.966	0.964
pendigits-32	0.904	0.997	0.942	0.942	0.990	0.988	0.986	0.989	0.989
phoneme-9952	0.835	0.915	0.858	0.880	0.872	0.841	0.728	0.884	0.913
pollen-3735	0.499	0.439	0.514	0.504	0.494	0.439	0.506	0.506	0.499
satimage-2074	0.872	0.901	0.876	0.891	0.904	0.879	0.884	0.899	0.917
segment-146822	0.866	0.948	0.912	0.917	0.890	0.939	0.916	0.930	0.926
shuttle-146212	0.998	1.000	0.914	0.791	0.997	0.971	0.692	1.000	1.000
spambase-43	0.896	0.939	0.924	0.926	0.937	0.909	0.764	0.933	0.937
splice-45	0.944	0.877	0.838	0.954	0.944	0.823	0.685	0.947	0.941
sylvine-168912	0.901	0.920	0.844	0.499	0.904	0.908	0.499	0.938	0.942
wall-robot-navigation-9960	0.965	0.945	0.652	0.929	0.945	0.843	0.765	0.998	0.998
wilt-146820	0.969	0.980	0.948	0.946	0.983	0.977	0.946	0.979	0.979
median	0.862	0.899	0.840	0.853	0.887	0.847	0.760	0.893	0.921

Table 5: Accuracies per dataset

	RandomForest	TabPFN-ICD	MLP	saint	TabNet	TabPFN	vime	XGBoost	XGBoost (Tuned)
dataset									
Amazon-employee-access-34539	0.943	0.941	0.942	0.947	0.943	0.640	0.942	0.943	0.946
Click-prediction-small-190408	0.832	0.823	0.832	0.833	0.832	0.547	0.832	0.835	0.831
GesturePhaseSegmentationProcessed-14969	0.501	0.644	0.481	0.470	0.486	0.459	0.454	0.616	0.689
JapaneseVowels-3510	0.832	0.994	0.855	0.977	0.983	0.932	0.918	0.963	0.980
MagicTelescope-3954	0.842	0.873	0.841	0.828	0.879	0.848	0.700	0.876	0.887
MiniBooNE-168335	0.895	0.934	0.886	0.909	0.818	0.879	0.811	0.933	0.941
PhishingWebsites-14952	0.924	0.962	0.944	0.948	0.959	0.934	0.846	0.947	0.967
Satellite-167211	0.990	0.988	0.990	0.984	0.990	0.988	0.986	0.990	0.990
ada-agnostic-3896	0.845	0.836	0.840	0.786	0.832	0.786	0.751	0.856	0.840
adult-census-3953	0.842	0.834	0.802	0.824	0.850	0.795	0.759	0.869	0.861
adult-7592	0.845	0.849	0.805	0.816	0.857	0.796	0.761	0.871	0.876
artificial-characters-14964	0.494	0.745	0.549	0.624	0.645	0.607	0.615	0.762	0.915
bank-marketing-14965	0.892	0.900	0.892	0.848	0.910	0.819	0.883	0.906	0.906
cardiotocography-9979	0.986	1.000	0.531	1.000	1.000	1.000	0.272	1.000	1.000
churn-167141	0.894	0.934	0.872	0.912	0.954	0.862	0.860	0.950	0.948
connect-4-146195	0.661	0.824	0.777	0.815	0.837	0.481	0.658	0.774	0.854
eeg-eye-state-14951	0.778	0.954	0.578	0.551	0.632	0.909	0.545	0.879	0.944
electricity-219	0.774	0.862	0.807	0.803	0.819	0.771	0.602	0.853	0.931
elevators-3711	0.817	0.899	0.769	0.707	0.891	0.874	0.691	0.872	0.881
eye-movements-3897	0.518	0.598	0.468	0.547	0.559	0.508	0.471	0.629	0.723
first-order-theorem-proving-9985	0.503	0.569	0.526	0.590	0.495	0.453	0.418	0.593	0.596
house-16H-3686	0.862	0.889	0.794	0.764	0.846	0.846	0.706	0.887	0.891
jannis-168330	0.644	0.685	0.675	0.685	0.711	0.533	0.644	0.701	0.720
jungle-chess-2pcs-raw-endgame-complete-167119	0.747	0.879	0.818	0.952	0.884	0.766	0.711	0.843	0.848
kc1-3917	0.872	0.877	0.848	0.848	0.867	0.773	0.848	0.863	0.858
kr-vs-kp-3	0.941	0.991	0.991	0.994	0.988	0.988	0.828	0.994	0.994
magic-146206	0.842	0.881	0.840	0.826	0.880	0.842	0.707	0.876	0.887
mfeat-fourier-14	0.750	0.795	0.705	0.860	0.790	0.765	0.620	0.860	0.840
mfeat-karhunen-16	0.925	0.925	0.955	0.955	0.945	0.895	0.970	0.935	0.945
mfeat-morphological-18	0.650	0.705	0.100	0.640	0.710	0.690	0.100	0.660	0.645
mfeat-zernike-22	0.730	0.800	0.790	0.490	0.800	0.810	0.775	0.750	0.775
mushroom-24	0.993	1.000	1.000	1.000	1.000	1.000	0.985	1.000	1.000
numerai28.6-167120	0.516	0.518	0.518	0.504	0.513	0.511	0.512	0.519	0.506
nursery-9892	0.873	0.999	0.960	0.921	0.956	0.944	0.984	0.997	0.997
optdigits-28	0.956	0.991	0.964	0.970	0.979	0.941	0.973	0.975	0.973
ozone-level-8hr-9978	0.945	0.945	0.937	0.937	0.945	0.909	0.937	0.937	0.941
page-blocks-30	0.969	0.960	0.954	0.964	0.947	0.920	0.931	0.965	0.964
pendigits-32	0.905	0.997	0.945	0.942	0.990	0.988	0.985	0.989	0.989
phoneme-9952	0.835	0.915	0.858	0.880	0.872	0.835	0.728	0.884	0.913
pollen-3735	0.499	0.457	0.514	0.504	0.494	0.457	0.506	0.506	0.499
satimage-2074	0.879	0.904	0.880	0.893	0.904	0.877	0.890	0.902	0.919
segment-146822	0.870	0.948	0.918	0.918	0.892	0.939	0.918	0.931	0.926
shuttle-146212	0.998	1.000	0.939	0.851	0.998	0.953	0.786	1.000	1.000
spambase-43	0.896	0.939	0.924	0.926	0.937	0.909	0.764	0.933	0.937
splice-45	0.944	0.878	0.837	0.953	0.944	0.821	0.708	0.947	0.940
sylvine-168912	0.901	0.920	0.844	0.499	0.904	0.908	0.499	0.938	0.942
wall-robot-navigation-9960	0.965	0.945	0.736	0.929	0.945	0.842	0.793	0.998	0.998
wilt-146820	0.969	0.979	0.948	0.946	0.983	0.975	0.946	0.979	0.979
median	0.866	0.902	0.841	0.855	0.887	0.844	0.762	0.894	0.923