

B.Sc. thesis Proposal “Continued-pretraining for TabPFN with time-series data“

Friedrich Romberg

1 Motivation

In today's present world, tabular data is one of the most common data formats used in various fields like healthcare, finance etc. But with the rise of machine learning and deep learning, this type of data has long been overlooked. Until 2023, gradient boost decision trees have dominated classification problems in tabular data, but with release of TabPFN, a transformer model pre-trained to solve artificially generated classification tasks from a synthetic prior, a new benchmark for tabular inference tasks has been set. In a single forward pass, TabPFN is able to solve classification, regression, and inference tasks.

There are two main ways to enhance the performance of this specific model even more. One of them is finetuning a pre-trained model and the other is continued pre-training a pre-trained model. The key difference between those two is that in finetuning one uses real-world data to train a pre-trained model for a specific domain like healthcare etc. and in continued pre-training one uses a specific prior for this domain to generate synthetical classification, regression, or inference tasks to further train the model.

Although there are already some models which achieve top rank results in time series data, the impact of finetuning or continued pretraining has not been studied and potential benefits remain uncovered.

2 Related Work

Continued pre-training on large real-world datasets in a specific domain yields superior downstream predictive accuracy in the specified domain compared to an only pre-trained model [Hollmann et al., 2025]. This validates the assumption that TabPFN performs better if we adapt weights to a downstream task.

In Kolberg et al. [2025] we saw that if we continue pre-training TabPFNv2 (so continued training on a synthetic prior) in a similar modality (tabular data with high feature and low sample count) will not decrease performance on non high dimension, low sample size data.

Hoo et al. [2025] introduces TabPFN-TS: A model that combines TabPFN-v2 with lightweight feature engineering to enable both point and probabilistic forecasting.

Dooley et al. [2023] is especially relevant since they created a synthetic prior for time-series data. They create the synthetic data by generating a trend, seasonal, and noise factor all either observed once a day, month, or year.

3 What We Want to Explore

The task for this thesis is to investigate how much a pre-trained TabPFNv2 model forgets if we continue pre-training to a new modality using a prior. For this task, we will use time-series as a new modality. TabPFNv2 can work with time-series as well, as Hoo et al. [2025] showed, although it has not seen time-series data during pre-training. Good performance in time-series data via preprocessing the data into a bunch of features is only a “byproduct” from a broad of pre-training data.

The questions we will address are:

1. Can we finetune TabPFN-TS on time-series data (and does it improve performance)?
2. Can we build a prior that generates realistic time-series data?
3. Can we continue pre-training TabPFN-TS with a given prior?
4. Which one of the following models performs best on various high scale benchmarks: TabPFN-TS, finetuned TabPFN-TS, or continued pre-trained TabPFN-TS?
5. (If we have time) Can we further improve performance by finetuning on multiple datasets similar to RealTabPFN [Hollmann et al., 2025]

4 Planned Experiments

To answer all the questions we defined above, we will conduct the following experiments.

We are using the code from TabPFN-TS. They used the time series datasets from GIFT-Eval as benchmark, so we are using datasets of this collection for finetuning.

For this, we will create a list of tasks we will carry out for every question we defined above.

1. For the first question, we first have to do the following things:
 - (a) Look at the TabPFN-TS code and check if it works properly. If not, fix the problems.
 - (b) Choose one or two datasets, used by Hoo et al. [2025] for evaluation, to finetune TabPFN-TS with.

2. The second question will be answered by completing the following tasks:

- (a) Look at the prior used in Dooley et al. [2023] and check if the prior can be used to continue pre-training of TabPFN-TS. If not: Look for another prior or adjust this one.
- (b) Continue Pre-training on TabPFN-TS with the given prior.
- (c) Use the datasets we utilized for finetuning in Step 1 to evaluate the continued pretrained model.

3. If there is still enough time left, or we encounter issues while answering question 2, we will address this question and work on the following tasks:

- (a) Finetune TabPFN-TS with the methods of Hollmann et al. [2025] on 4-5 datasets from GIFT-Eval used in Hoo et al. [2025] and select a new one for evaluation. Then cross-validate the datasets.
- (b) Apply continued pre-trained TabPFN-TS from Step 2 on each one of the datasets used above and evaluate if multi-dataset finetuning is better than continued pretraining on synthetic time-series data.

References

Samuel Dooley, Gurnoor Singh Khurana, Chirag Mohapatra, Siddartha V. Naidu, and Colin White. Forecastpfn: Synthetically-trained zero-shot forecasting. In *Advances in Neural Information Processing Systems*, volume 36, pages 2403–2426, 2023.

Niklas Hollmann, Akshay Garg, Muntasir Ali, Lennart Purucker, Samuel Mueller, and Frank Hutter. Realtabpfn: Improving tabular foundation models via continued pre-training with real-world data. *arXiv preprint arXiv:2507.03971*, 2025.

Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. From tables to time: How tabpfn-v2 outperforms specialized time series forecasting models. *arXiv preprint arXiv:2501.02945*, 2025.

Christopher Kolberg, Katharina Eggensperger, and Nico Pfeifer. Tabpfn-wide: Continued pre-training for extreme feature counts. *arXiv preprint arXiv:2510.06162*, 2025.