Understanding the Limits of Deep Tabular Methods with Temporal Shift

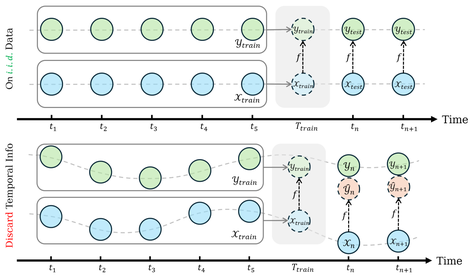
Hao-Run Cai & **Han-Jia Ye**, Nanjing University

LAMDA Learning And Mining from DatA

ICML International Conference On Machine Learning

Challenges in Learning from Temporal Tabular Data

Most machine learning approaches are built on the ***assumption of i.i.d. data***. However, tabular data are often ***collected over time***, resulting in ***temporal shifts*** at each time point.

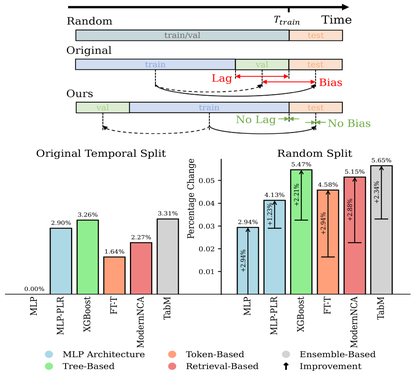


- How can existing models ***be better trained*** under temporal shifts?
- How can models be equipped with ***temporal adaptability***?

Why Temporal Splits Fail?

TabReD [Rubachev et al., ICLR'25] employs a ***temporal validation split***, utilizing earlier data for training and later data for model selection.

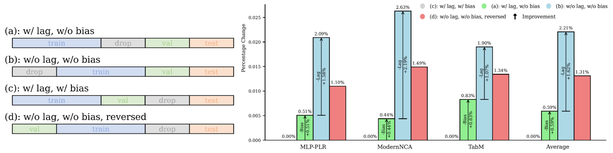
We discovered that even when ***randomly splitting*** the training and validation sets, the model outperformed the temporal split.



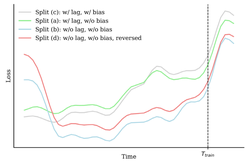
Start from the most intuitive difference:

- ***Training lag***: the training-available data closest to the test time are not used for training.
- ***Validation bias***: the shift degree of test set relative to validation set is more significant in temporal splits.

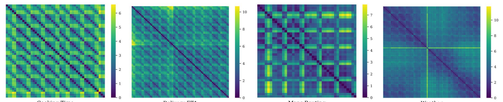
Ablation studies & ***loss distribution***: the effectiveness in reducing training lag and validation bias.



- Training lag↓: better *test-time performance*.
- Validation bias↓: *better generalization*.



MMD visualization confirms the empirical uniformity of temporal shifts across time slices.



Training Protocol

We introduce the following ***training protocol*** for temporal data:

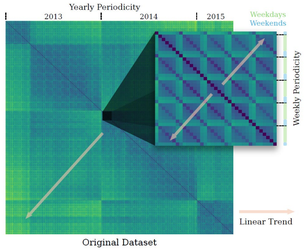
1. The *lag between training and test set* should be minimized.
2. The *validation bias* should be minimized.
3. Effective validation can be achieved in the *reverse temporal direction* by aligning the shift in the validation set with the actual shift between training and testing data.

	Splits	Avg. Imp.
Mean Performance ↑	Original	–
	Random	+2.17%
	Ours	**+2.18%**
Standard Deviation ↓	Original	–
	Random	+154%
	Ours	**+16.7%**

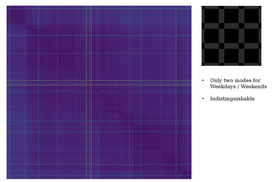
Comparable performance with ***better stability***.

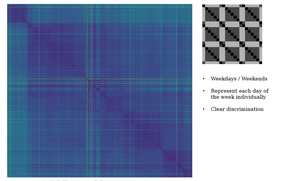
Model Representation

The ***loss of the rich temporal information*** during training.



- The original dataset *clearly reveals* both periodicity and trend.
- Yet they are *lost in the model representations*.
- Mitigate this phenomenon by introducing *temporal embedding*.





Temporal Embedding

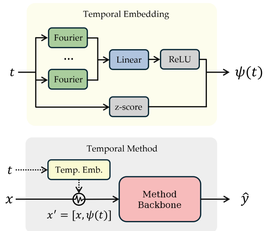
Compensate for temporal information via ***temporal embedding***.

$$\psi(t) = \left[\mathrm{ReLU}\left(\mathrm{Linear}\left(\mathrm{Periodic}(t)\right)\right), \mathrm{Trend}(t)\right]$$

$$\mathrm{Periodic}(t) = \left[\mathrm{Fourier}(t, T_1), \ldots, \mathrm{Fourier}(t, T_m)\right]$$

$$\mathrm{Fourier}(t, T) = \left[\sin\left(\frac{2\pi k t}{T}\right), \cos\left(\frac{2\pi k t}{T}\right)\right]$$

$$\mathrm{Trend}(t) = \text{z-score}(t)$$



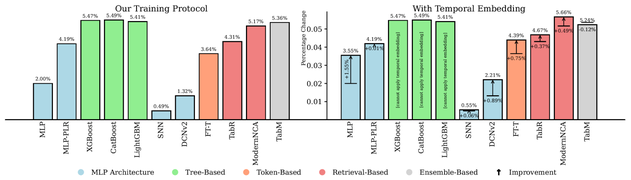
The states in the temporal cycles may exhibit a ***square wave*** form, *e.g.*, weekdays and weekends, requiring higher-order sine and cosine terms for precise representation.

Emb.	MLP	MLP-PLR	ModernNCA	Avg. Imp.
Num	−0.04%	−0.06%	−0.04%	−0.05%
Time	−0.70%	−0.15%	−0.32%	−0.39%
TabPFN-TS [Hoo et al., NeurIPS'24 WS]	+0.25%	–	−0.43%	−0.09%
PLR [Gorishniy et al., NeurIPS'22]	+0.70%	**+0.01%**	+0.02%	+0.25%
Ours	**+1.31%**	**+0.01%**	**+0.30%**	**+0.54%**

- ***Learnable*** & ***scalable***.
- Fourier expansion for precisely capture ***periodicity***.
- Uses normalized timestamps as a ***trend*** feature.
- Leads to a further 0.74% gain in performance.

Results

Performance comparison before and after adopting our temporal embedding into our training protocol on TabReD benchmark.



Performance rankings of original temporal split, random split, and our temporal split with and without our temporal embedding.

Splits	MLP	PLR	FT-T	SNN	DCNv2	TabR	MNCA	TabM	XGBoost	CatBoost	LGBM
TabReD	7.750	4.375	6.875	9.375	8.250	7.375	6.500	**3.125**	3.375	4.250	4.750
Random	8.250	5.625	5.625	10.250	9.625	8.000	4.750	**2.750**	3.125	3.125	4.875
Ours	8.000	5.750	7.500	9.500	8.375	8.125	4.875	4.000	3.375	**2.125**	4.375
Ours + temporal embedding	7.875	6.625	6.250	9.625	9.250	6.250	4.625	3.125	4.500	**2.875**	5.000

Contact

Paper Code Benchmark Survey