SBER AI Lab

Tsururu: A Python-based Time Series Forecasting Strategies Library

Alina Kostromina[1], Kseniia Kuvshinova[1,3], Aleksandr Yugay[2,4], Andrey Savchenko[1,5], Dmitry Simakov[1]

[1] Sber AI Lab, [2] Sber, [3] Skoltech, [4] MIPT, [5] ISP RAS Research Center for Trusted Artificial Intelligence, Moscow, Russia

alina.kostromina@gmail.com

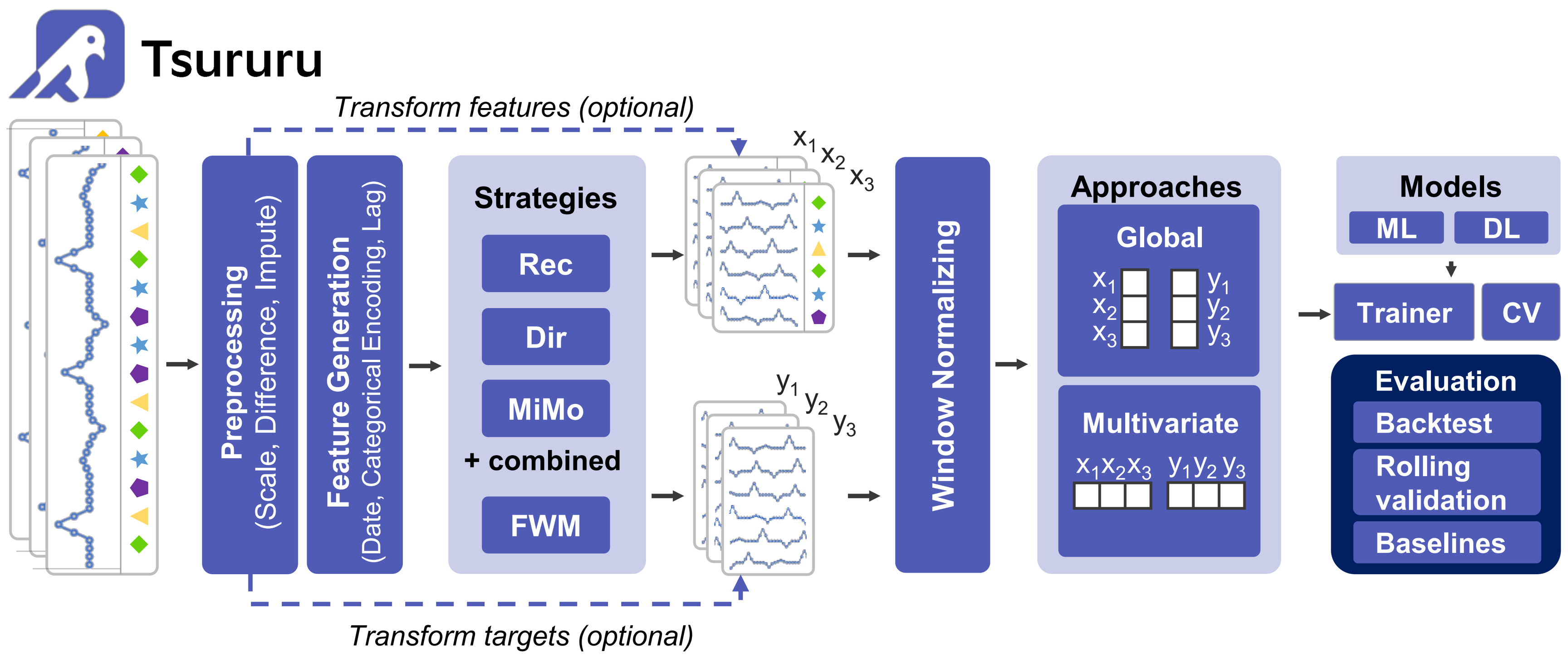
GitHub sb-ai-lab

GitHub Tsururu

Abstract

While current time series research focuses on developing new models, crucial questions of selecting an optimal approach for training such models are underexplored. Tsururu, a Python library introduced in this paper, bridges SoTA research and industry by enabling flexible combinations of global and multivariate approaches and multi-step-ahead forecasting strategies. It also enables seamless integration with various forecasting models. Available at https://github.com/sb-ai-lab/tsururu.

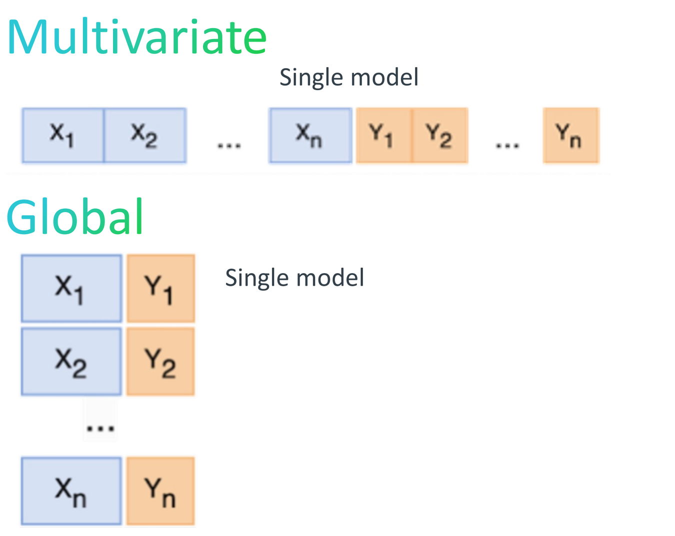
Framework Design



In Tsururu, data moves through a structured pipeline.

Multi-series prediction strategies [1, 2]:

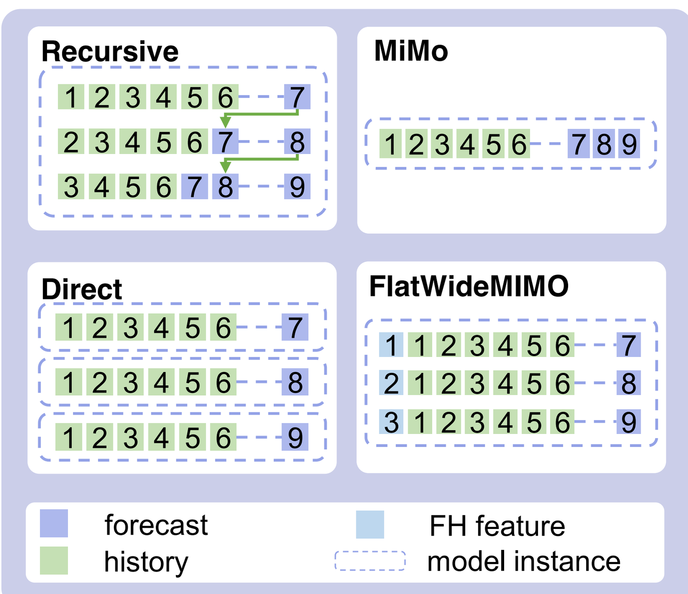
- **Global** — fits a single model to all time series, treating them as independent.
- **Multivariate** — fits a single model to all time series and allows the model to capture dependencies between them.
- Besides, each DL model supports **Channel Independence (CI)** and **Channel Dependence (CD)** modes [3].

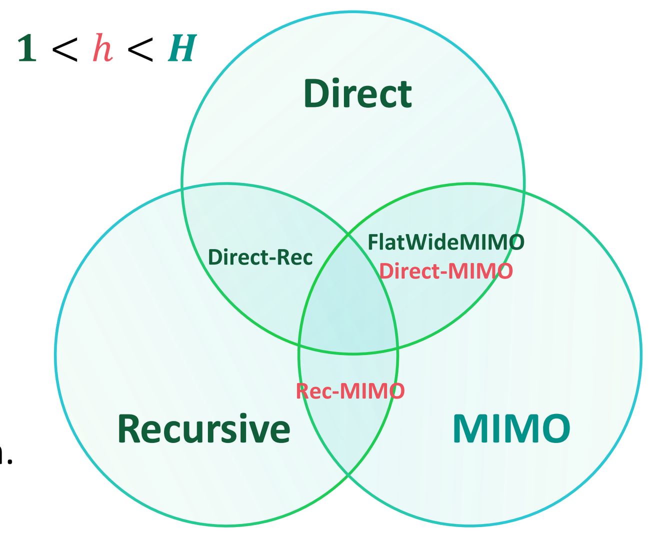


Multi-step-ahead prediction strategies:

h — model horizon (MH); H — full horizon.

- **Recursive** — trains a single model to predict the next point (MH = 1), iteratively extending predictions across the forecast horizon and using previous predictions to update the features in the test data [4].
- **Direct** — uses separate models for each point (MH = 1) in the forecasting vector [4].
- **MIMO** (multi-input-multi-output) — trains a single model to simultaneously predict the entire forecast horizon (MH = H) [5].
- **FWM** (flat-wide-mimo) — uses a single model to predict a specific point in the forecasting vector, with the horizon index explicitly provided as an input feature.
- **Hybrid strategies:** Recursive-MIMO, Direct-MIMO, in which MH > 1.





Data Transformations:

- **Series-to-Series** — data preparation and feature generation.
  - StandardScaler, DifferenceNormalizer.
- **Series-to-Features** — build a "wide" series matrix with lagged versions of generated features.
  - Seasonality features.
- **Features-to-Features** — perform window-based processing.
  - LastKnownNormalizer (normalizing values by the most recent observed one in available history).

Models:

- **ML:** CatBoost [6], SketchBoost [7].
- **DL:** Linear (DLinear [8], CycleNet [9]), CNN-based (TimesNet [10]), Transformer-based (PatchTST [11], GPT2 [12]).

Experimental setup

Models:

- SketchBoost, DLinear, PatchTST, GPT2, CycleNet.

Strategies:

- Global, Multivariate (both CI & CD).
- Recursive (MH=1), Recusrive-MIMO (MH=6), MIMO, FWM.

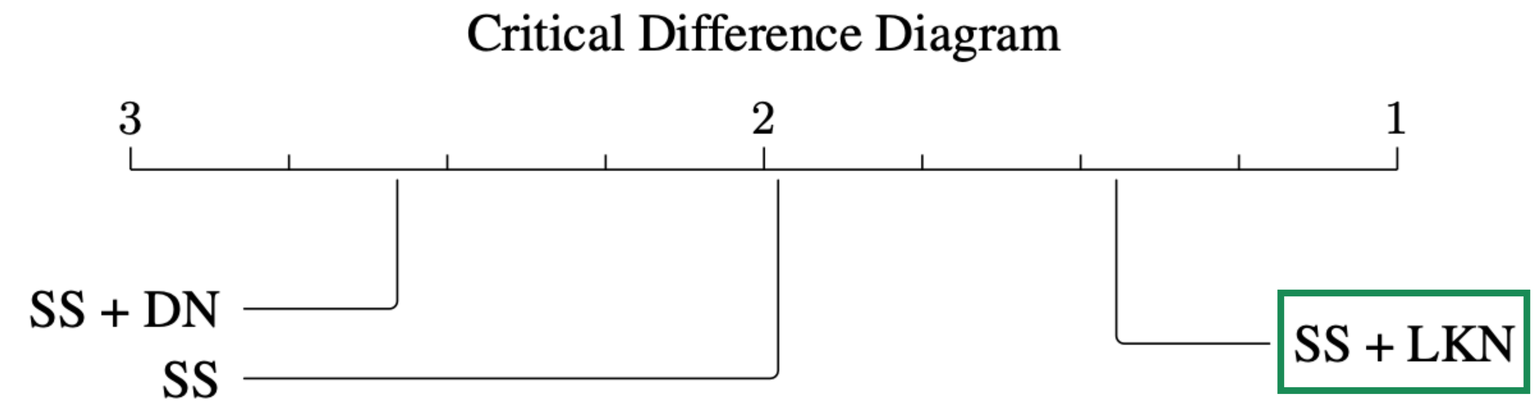
Dataset:

- ILI [https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html]
- *There are results for other datasets in our github repository.*

Other:

- 50 epochs. Batch size=32.
- LR=0.0001. Cosine-based scheduler.
- History=96. Horizon=24.
- Model hyperparameters were unchanged from initial works.

Study 1: Preprocessing



A critical difference diagram visualizes the ranking of preprocessing methods across other fixed hyperparameters of the pipeline. Methods not connected by a horizontal line are significantly different. Here, we consider StandardScaler (SS) with LastKnownNormalizer (LKN) or DifferenceNormalizer (DN)

- **Used delta-mode** (based on subtraction; delta- and ratio-modes are available)

Study 2: Features & Strategies

Hyperparam	Value	NN models		Boosting		Overall	
		Rank	Median MAE	Rank	Median MAE	Rank	Median MAE
Datetime Features	False	1.3819	1.0087	1.3333	1.6050	1.3743	1.0448
	True	1.6181	1.1323	1.6667	1.6174	1.6257	1.1785
ID Features	False	1.7262	1.0780	1.5714	1.6174	1.6952	1.1319
	True	1.2738	1.0024	1.4286	1.5898	1.3048	1.0611
Mode	Global	1.5476	1.0056	1.0952	1.5648	1.5476	1.0735
	Multivariate CI	2.2619	1.1217	NaN	NaN	2.2619	1.1217
	Multivariate CM	2.1905	1.1319	1.9048	1.6248	2.1905	1.2129
Prediction Strategy	FlatWideMIMO	3.9375	1.3080	2.8889	1.6208	3.7719	1.3543
	MIMO	1.7500	1.0280	2.4444	1.6072	1.8596	1.0621
	Recursive ($MH = 1$)	2.4167	1.0314	2.7778	1.6066	2.4737	1.0763
	Recursive ($MH = 6$)	1.8958	1.0228	1.8889	1.5816	1.8947	1.0541

Comparison of hyperparameters of data manipulation pipeline. For boosting, there is no Multivariate CI mode by construction; only Multivariate CM mode is available.

Study 3: Models & Strategies

rank	Model	Strategy	MAE (test)	Model	Strategy	MAE (val)
1	GPT4TS	Recursive ($MH = 6$)	0.7804	GPT4TS	MIMO	0.2713
2	GPT4TS	Recursive ($MH = 1$)	0.7822	GPT4TS	Recursive ($MH = 6$)	0.2833
3	PyBoost	FlatWideMIMO	0.7921	GPT4TS	Recursive ($MH = 1$)	0.2938
4	GPT4TS	MIMO	0.7926	PatchTST	MIMO	0.3005
5	PatchTST	Recursive ($MH = 6$)	0.8630	PatchTST	Recursive ($MH = 6$)	0.3050
6	PatchTST	MIMO	0.8769	DLinear	Recursive ($MH = 6$)	0.3169
7	PatchTST	Recursive ($MH = 1$)	0.8949	PatchTST	Recursive ($MH = 1$)	0.3180
8	DLinear	Recursive ($MH = 6$)	0.9193	DLinear	MIMO	0.3205
9	DLinear	MIMO	0.9220	PyBoost	FlatWideMIMO	0.3239
10	DLinear	Recursive ($MH = 1$)	0.9314	DLinear	Recursive ($MH = 1$)	0.3313

- The diversity of top-ranked models and strategies underscores **the importance of exploring rarely used model-strategy combinations.**

Best 10 combinations model-strategy based on MAE on validation and test subsets.

Conclusion & Future Work

- Tsururu supports ablating all-with-all combinations of preprocessing, models, multi-series and multi-step-ahead prediction strategies.
- Our experiments show the advantages of using rarely employed preprocessing (like LastKnownNormalizer) and strategies (like Recursive for PatchTST).
- Future work includes incorporating Rectify, DirRec; building a universal neural network constructor, testing patching techniques, and supporting mixed discretization within multivariate datasets.

References

[1] Tim Januschowski, Jan Gasthaus, Yuyang Wang, David Salinas, Valentin Flunkert, Michael Bohlke-Schneider, and Laurent Callot. Criteria for classifying forecasting methods. International Journal of Forecasting, 36(1):167–177, 2020.

[2] [Montero-Manso and Hyndman, 2021] Pablo Montero- Manso and Rob J Hyndman. Principles and algorithms for forecasting groups of time series: Locality and globality. International Journal of Forecasting, 37(4):1632–1653, 2021.

[3] Lu Han, Han-Jia Ye, and De-Chuan Zhan. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. IEEE Transactions on Knowledge and Data Engineering, 2024.

[4] Andreas S Weigend. Time series predic- tion: forecasting the future and understanding the past. Routledge, 2018.

[5] [Bontempi, 2008] Gianluca Bontempi. Long term time se- ries prediction with multi-input multi-output local learn- ing. Proc. 2nd ESTSP, pages 145–154, 2008.

[6] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categor- ical features. Advances in neural information processing systems, 31, 2018.

[7] Leonid Iosipoi and Anton Vakhrushev. Sketchboost: Fast gradient boosted decision tree for multioutput problems. *Advances in Neural Infor- mation Processing Systems*, 35:25422–25435, 2022.

[8] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series fore- casting? In Proceedings of the AAAI conference on artifi- cial intelligence, volume 37, pages 11121–11128, 2023.

[9] Shengsheng Lin, Weiwei Lin, HU Xinyi, Wentai Wu, Ruichao Mo, and Haocheng Zhong. Cyclenet: Enhancing time series forecasting through modeling peri- odic patterns. In The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024.

[10] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In International Conference on Learning Rep- resentations, 2023.

[11] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In The Eleventh International Conference on Learning Rep- resentations, 2023.

[12] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. Advances in neural information processing systems, 36:43322–43355, 2023.