Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

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, Moscow, Russia
[2]Sber, Moscow, Russia
[3]Skoltech, Moscow, Russia
[4]MIPT, Moscow, Russia
[5]ISP RAS Research Center for Trusted Artificial Intelligence, Moscow, Russia
alina.kostromina@gmail.com

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.

1 Introduction

A fundamental task in time series analysis is forecasting, which involves predicting future values $X_{t+1:H}$ over a horizon H at timestep t, given the historical data $X_{1:t}$ and known covariates $Z_{1:H}$ for all time points [Salinas *et al.*, 2020; Kim *et al.*, 2024]. Later in this paper, we consider multivariate data comprising multiple time series within the dataset.

To advance forecasting capabilities, various libraries have been developed to benchmark state-of-the-art (SoTA) models, including tslib [Wang *et al.*, 2024], neuralforecast [Olivares *et al.*, 2022], uni2ts [Aksu *et al.*, 2024] (for in-context learning models), pytorch-forecasting [Beitner, 2020] (has earlier deep learning models), BasicTS [Shao *et al.*, 2024]. However, they are suboptimal for real-world scenarios. They often rely on fixed forecasting strategies, offer constrained support for exogenous variables, and face difficulties in the usage of custom datasets. These factors limit their suitability for industrial scenarios. A potential solution is a time series library that seamlessly handles diverse datasets, including non-aligned ones or those with exogenous features.

Indeed, some of the issues above are partially covered in existing practically oriented time series libraries. Almost all such libraries use global approach [Januschowski *et al.*, 2020; Montero-Manso and Hyndman, 2021] (useful for non-aligned series) and support exogenous variables: Darts [Herzen *et al.*, 2022], sktime [Király *et al.*, 2025], gluonts [Alexandrov *et al.*, 2020], tslearn [Tavenard *et al.*, 2020], skforecast [Amat Rodrigo and Escobar Ortiz, 2024], AutoTS [Catlin, 2021] (global for aligned time series only), ETNA [Alekseev *et al.*, 2021], mlforecast [Morales, 2021]. However, some of them could not use exogenous features as input, like tslearn [Tavenard *et al.*, 2020], and tsspiral [Cerliani, 2023] or work only in the multivariate setting, such as tsai [Oguiza, 2023].

Nevertheless, the full predictive potential of time series models remains unexplored, as the impact of forecasting strategies [Taieb, 2014] has received limited attention. Classical time series forecasting is based either on a recursive strategy [Gustin *et al.*, 2018] or a multi-input-multi-output (MIMO) strategy [Bontempi, 2008; Bontempi and Taieb, 2011; Kline, 2004]. Recent research claimed that the question of which strategy to use is still open [Green *et al.*, 2024]. Notably, most state-of-the-art neural networks are trained using the MIMO strategy. However, incorporating additional forecasting strategies demonstrates that MIMO is not always optimal (see Section 3). Only Darts, sktime, skforecast and tspiral have at least three strategies. Nevertheless, they offer only a narrow pool of preprocessing methods. For example, none of them is integrated with the LastKnownNormalizer — subtraction or division on the last element in available history. Our experiments show that this rarely used preprocessing enhances forecasting performance significantly (see Section 3).

Given the weaknesses of existing time series libraries, this paper introduces **Tsururu** (Fig. 1), the modular framework for both practitioners and researchers that makes combinable global/multivariate approaches, different forecasting strategies, and models. This flexibility connects state-of-the-art research with real-world business applications.

2 Framework Design

The Tsururu architecture (Fig. 1) facilitates the comparison of configurations and the development of task-specific forecasting systems, incorporating strategies with support for exogenous variables.

Multi-series prediction strategies. Tsururu supports both **Global** and **Multivariate** approaches for all models for handling multiple time series [Januschowski *et al.*, 2020; Montero-Manso and Hyndman, 2021]. The Global approach fits a single model to all time series, treating them as independent, while the Multivariate approach allows the model to capture dependencies between them. Moreover, each deep learning model in Tsururu supports **Channel Independence**

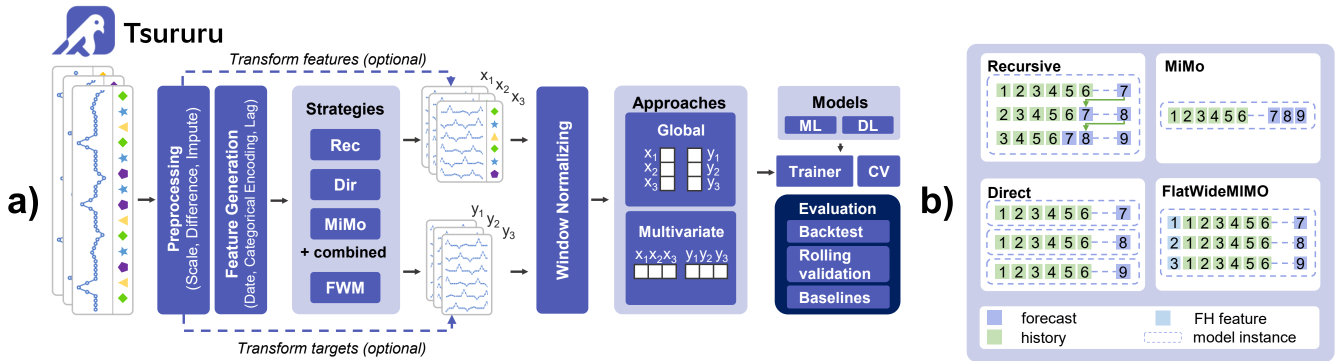

Figure 1: a) The architecture of the proposed Tsururu framework. Most existing libraries typically support either (1) global and multivariate forecasting or (2) multiple forecasting strategies, but not both. Tsururu enables the exploration of all possible combinations of available ML and DL models with various forecasting strategies, preprocessing techniques, and the global/multivariate setting. b) The inference stage of the forecasting process for each strategy. FH stands for forecasting horizon.

(CI) and **Channel Mixing (CM)** modes [Han *et al.*, 2024]. The last one controls the interaction between series in the multivariate setting.

Multi-step-ahead prediction strategies. Different strategies can be applied to multi-step forecasting. **Recursive (Rec)** [Weigend, 2018] 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. MH denotes the model horizon, i.e., the number of points which the model outputs in a single step. Tsururu also supports the hybrid **Recursive-MIMO (Rec-MIMO)** strategy ($MH > 1$), which follows the recursive strategy but generates multiple-step predictions at each iteration instead of a single point. **Direct (Dir)** [Weigend, 2018] uses separate models for each prediction step with model horizon length, constructing the full forecast horizon. **MIMO** [Bontempi, 2008] trains a single model to simultaneously predict the entire forecast horizon ($MH = H$). **FlatWideMIMO (FWM)** uses a single model to predict a specific point in the forecasting vector, with the horizon index explicitly provided as an input feature. While rarely used, this strategy can be effective.

Pipeline and Data Transformations. Tsururu's pipeline applies sequential transformations to time series data, categorized into three types: **Series-to-Series** are used for data preprocessing and feature generation, **Series-to-Features** build a "wide" series matrix with lagged versions of generated features, **Features-to-Features** perform window-based processing, such as LastKnownNormalizer (LKN), based on normalizing values by the most recent observed one in available history, i.e. in corresponding row of the "wide" series matrix. Tsururu supports separate transformations for features and targets, allowing more data preparation flexibility.

Models. Tsururu offers classical ML and deep learning models. For ML, it includes boosting methods like CatBoost [Prokhorenkova *et al.*, 2018] and SketchBoost [Iosipoi and Vakhrushev, 2022], with SketchBoost chosen for its speed in GBDT training [Friedman, 2001]. DL models are grouped into linear (DLinear [Zeng *et al.*, 2023], CycleNet [Lin *et al.*, 2024]), CNN-based (TimesNet [Wu *et al.*, 2023]), and Transformer-based (PatchTST [Nie *et al.*, 2023], GPT4TS [Zhou *et al.*, 2023]).

Data flow. In Tsururu, data moves through a structured pipeline. First, time series pass through **preprocessing**. Next, some **relevant features are generated**: categorical encodings and datetime features. Once these features are in place, Tsururu operates at the index level, converting "long" into "wide" series through lag transformations. This protocol prevents leakage from future values, supports dynamic feature generation, and reduces memory overhead. Afterward, Tsururu applies the beforehand chosen multi-step forecasting **strategy** and selects either the Global or Multivariate approach. **Window normalization** can then be used to rescale each observation in wide data to the most recent known value, thus helping to cope with local shifts in the data distribution. Finally, the prepared dataset and model are passed to the **Trainer** module. The Trainer is responsible for the training process. It also uses **cross validator (CV)** to generate validation splits. When multiple splits exist, a separate model is trained for each split, and their predictions are averaged during inference. The validation set can also be used for **early stopping**. The final model is then validated using backtesting and rolling validation methods to ensure robustness (note that it is not the same as CV in Trainer).

3 Experimental Results

Setup. We examine such models as SketchBoost, DLinear, PatchTST, GPT4TS, and CycleNet, exploring both global and multivariate approaches (with either Channel Independence or Channel Mixing) and forecasting strategies: Recursive (with model horizon MH equal to 1 or 6), MIMO, and FlatWideMIMO. These models are evaluated on the ILI dataset[1], a challenging weekly time series with annual periodicity and distinct temporal structures, to showcase the library's capabilities and highlight the importance of non-default model-approach-strategy combinations.

For all settings, we used a cosine-based scheduler for over 50 epochs, with the learning rate set to 0.0001, and per-

[1]https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html

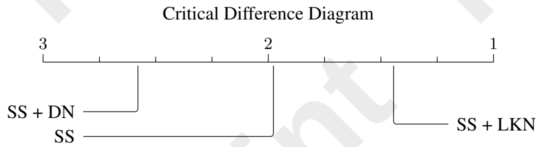

Figure 2: A critical difference diagram visualizes the ranking of preprocessing methods across other fixed hyperparameters of the pipeline. Methods not connected by a horizontal line are significantly different. Here, we consider StandardScaler (SS) with LastKnownNormalizer (LKN) or DifferenceNormalizer (DN)

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

Table 1: Comparison of hyperparameters of data manipulation pipeline. For boosting, there is no Multivariate CI mode by construction; only Multivariate CM mode is available. MH stands for the model horizon of the Recursive strategy.

formed parameter updates at the end of each batch. We fixed batch size to 32, history to 96, and horizon to 24. The model hyperparameters (i.e., hidden dimension, the number of attention heads, etc.) were left unchanged from their earlier configurations in the initial works [Zhou *et al.*, 2023; Zeng *et al.*, 2023]. In cases where the authors had not tested their models on ILI, we retained the hyperparameters for ETTh1 [Zhou *et al.*, 2021].

Results. As shown in Figure 2, LKN significantly outperforms default preprocessing strategies, demonstrating its effectiveness despite not being adopted in common libraries. We use delta-mode normalization (based on subtraction from the current value, the previous one for DifferenceNormalizer (DN), and the most recent one in available history for LKN).

Table 1 presents the results of an ablation study on hyperparameters of the data manipulation pipeline, analyzing the impact of date and id features inclusion, forecasting mode, and strategy selection on model performance. The id features improved model accuracy, while including date features led to worse performance for both neural networks and GBDT. The Global approach outperformed the multivariate one, achieving the lowest rank and median MAE across all models. The MIMO strategy ranked best for neural networks, while Rec-MIMO ($MH = 6$, see Section 2) achieved the lowest median MAE. For GBDT, Rec-MIMO ($MH = 6$) is the best strategy in rankings and median MAE. Thus, this hybrid approach is rarely used but is a competitive alternative.

Table 2 provides the independent ranking of models based on test and validation MAE. In this evaluation, we also considered an alternative scaling approach using ratio normalization without an initial StandardScaler for boosting models. Our results confirm that GPT4TS outperformed all other models. Notably, the Rec-MIMO strategy with $MH = 6$ achieved the best overall test MAE. However, on the validation set, the highest-ranked strategy was MIMO. Interestingly, FlatWideMIMO combined with boosting models also ranked among the top strategies, demonstrating that GBDT can be competitive when paired with non-standard multi-step-ahead forecasting approaches. The diversity of top-ranked models and strategies underscores the importance of exploring rarely used model-strategy combinations.

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

Table 2: Best 10 combinations model-strategy based on MAE on validation and test subsets.

Model	MAE (Original)	MSE (Original)	MAE (Ours)	MSE (Ours)
DLinear	1.081	2.215	1.037 ± 0.004	2.227 ± 0.010
PatchTST	0.754	1.319	0.716 ± 0.043	1.239 ± 0.103
GPT4TS	0.881	2.063	0.896 ± 0.024	2.028 ± 0.054
TimesNet	0.934	2.317	0.927 ± 0.031	2.033 ± 0.155
CycleNet	1.073*	2.400*	1.051 ± 0.013	2.345 ± 0.049

Table 3: Original vs. Ours. Metrics for the ILI dataset with a forecasting horizon of 24. Values marked with * indicate that the corresponding metric was not taken from the original paper but computed using the official implementation. Results are presented as mean ± standard deviation, calculated across three seeds.

Reproducibility. To validate the accuracy of our time series library, we reproduce results from previous studies using Tsururu and compare them to the metrics reported in the original papers (Table 3). The close alignment between these metrics demonstrates the fidelity of our implementations.

4 Conclusion

This paper introduces Tsururu, an open-source Python library for ablating all-with-all combinations of preprocessing, time series models, forecasting approaches, and strategies. It can be easily integrated with new SoTA models for fair benchmarking while providing key industrial features, such as exogenous variables and a Global approach to handling non-aligned time series. It also supports channel mixing and channel-independent forecasting, rarely used in existing libraries. Our experiments show the advantages of using rarely employed preprocessing (like LastKnownNormalizer) and combining strategies (like Recursive for PatchTST). Moreover, combining a Recursive strategy with the Global approach makes short time-series forecasting feasible, with extended horizons and support for custom datasets. The extended results for other datasets can be found in our repository. An ability to handle non-aligned series represents our essential advantage of combining regimes and strategies.

Future work includes incorporating Rectify [Taieb, 2014], and DirRec [Sorjamaa and Lendasse, 2006], building a universal neural network constructor, testing patching techniques, and supporting time series with mixed discretization (daily, monthly, weekly, etc.) within multivariate datasets.

Ethical Impact

The framework's methods and specifications do not directly relate to ethical concerns. However, high-stakes domains such as healthcare and finance, where time series are widespread, require careful attention. Before deploying our library in such contexts, a thorough evaluation is essential to ensure it does not support decisions that could negatively impact individuals or organizations.

Acknowledgments

The work of A. Savchenko was supported by a Research Center for Trusted Artificial Intelligence the Ivannikov Institute for System Programming of the Russian Academy of Sciences.

References

[Aksu *et al.*, 2024] Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Gift-eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

[Alekseev *et al.*, 2021] Andrey Alekseev, Julia Shenshina, Martin Gabdushev, Albina Munirova, Artem Levashov, Alexey Podkidyshev, Nikita Barinov, Dmitry Bunin, Alexander Chikov, and Artem Makhin. etna. https://github.com/etna-team/etna, 2021.

[Alexandrov *et al.*, 2020] Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, Lorenzo Stella, Ali Caner Türkmen, and Yuyang Wang. GluonTS: Probabilistic and Neural Time Series Modeling in Python. *Journal of Machine Learning Research*, 21(116):1–6, 2020.

[Amat Rodrigo and Escobar Ortiz, 2024] Joaquin Amat Rodrigo and Javier Escobar Ortiz. skforecast, 11 2024.

[Beitner, 2020] Jan Beitner. pytorch-forecasting. https://github.com/sktime/pytorch-forecasting, 2020.

[Bontempi and Taieb, 2011] Gianluca Bontempi and Souhaib Ben Taieb. Conditionally dependent strategies for multiple-step-ahead prediction in local learning. *International journal of forecasting*, 27(3):689–699, 2011.

[Bontempi, 2008] Gianluca Bontempi. Long term time series prediction with multi-input multi-output local learning. *Proc. 2nd ESTSP*, pages 145–154, 2008.

[Catlin, 2021] Colin Catlin. Autots. https://github.com/winedarksea/AutoTS, 2021.

[Cerliani, 2023] Marco Cerliani. tspiral. https://github.com/cerlymarco/tspiral, 2023.

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

[Green *et al.*, 2024] Riku Green, Grant Stevens, Zahraa Abdallah, et al. Time-series classification for dynamic strategies in multi-step forecasting. *arXiv preprint arXiv:2402.08373*, 2024.

[Gustin *et al.*, 2018] Matej Gustin, Robert S McLeod, and Kevin J Lomas. Forecasting indoor temperatures during heatwaves using time series models. *Building and Environment*, 143:727–739, 2018.

[Han *et al.*, 2024] 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.

[Herzen *et al.*, 2022] Julien Herzen, Francesco Lässig, Samuele Giuliano Piazzetta, Thomas Neuer, Léo Tafti, Guillaume Raille, Tomas Van Pottelbergh, Marek Pasieka, Andrzej Skrodzki, Nicolas Huguenin, et al. Darts: User-friendly modern machine learning for time series. *Journal of Machine Learning Research*, 23(124):1–6, 2022.

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

[Januschowski *et al.*, 2020] 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.

[Kim *et al.*, 2024] Jongseon Kim, Hyungjoon Kim, HyunGi Kim, Dongjun Lee, and Sungroh Yoon. A comprehensive survey of time series forecasting: Architectural diversity and open challenges. *arXiv preprint arXiv:2411.05793*, 2024.

[Király *et al.*, 2025] Franz Király, Markus Löning, Tony Bagnall, Matthew Middlehurst, Anirban Ray, Sajaysurya Ganesh, Martin Walter, George Oastler, Jason Lines, ViktorKaz, Benedikt Heidrich, Lukasz Mentel, Sagar Mishra, chrisholder, Daniel Bartling, Armaghan Shakir, Leonidas Tsaprounis, RNKuhns, Ciaran Gilbert, Mirae Baichoo, Felix Hirwa Nshuti, Hazrul Akmal, Alex-JG3, Guzal, Taiwo Owoseni, Patrick Rockenschaub, Pranav Prajapati, eenticott shell, and Sami Alavi. sktime/sktime: v0.36.0, February 2025.

[Kline, 2004] Douglas M Kline. Methods for multi-step time series forecasting neural networks. In *Neural networks in business forecasting*, pages 226–250. IGI global, 2004.

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

[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.

[Morales, 2021] Jose Morales. mlforecast. https://github.com/Nixtla/mlforecast, 2021.

[Nie *et al.*, 2023] 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 Representations*, 2023.

[Oguiza, 2023] Ignacio Oguiza. tsai - a state-of-the-art deep learning library for time series and sequential data. Github, 2023.

[Olivares *et al.*, 2022] Kin G. Olivares, Cristian Challú, Azul Garza, Max Mergenthaler Canseco, and Artur Dubrawski. NeuralForecast: User friendly state-of-the-art neural forecasting models. PyCon Salt Lake City, Utah, US 2022, 2022.

[Prokhorenkova *et al.*, 2018] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[Salinas *et al.*, 2020] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3):1181–1191, 2020.

[Shao *et al.*, 2024] Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Tao Sun, Guangyin Jin, Xin Cao, et al. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Transactions on Knowledge and Data Engineering*, 37(1):291–305, 2024.

[Sorjamaa and Lendasse, 2006] Antti Sorjamaa and Amaury Lendasse. Time series prediction using dirrec strategy. In *Esann*, volume 6, pages 143–148, 2006.

[Taieb, 2014] Souhaib Ben Taieb. Machine learning strategies for multi-step-ahead time series forecasting. *Universit Libre de Bruxelles, Belgium*, pages 75–86, 2014.

[Tavenard *et al.*, 2020] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, and Eli Woods. Tslearn, a machine learning toolkit for time series data. *Journal of Machine Learning Research*, 21(118):1–6, 2020.

[Wang *et al.*, 2024] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *arXiv preprint arXiv:2407.13278*, 2024.

[Weigend, 2018] Andreas S Weigend. *Time series prediction: forecasting the future and understanding the past*. Routledge, 2018.

[Wu *et al.*, 2023] 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 Representations*, 2023.

[Zeng *et al.*, 2023] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115, 2021.

[Zhou *et al.*, 2023] 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.