Topological Feature Generation in Automated Machine Learning for Time Series Forecasting

Valerii Pokrovskii[1], Ilia Revin[1], Vadim A. Potemkin[1], Sergey Kasianov[1], Nikolay O. Nikitin [1]
[1]ITMO University, Saint Petersburg, Russia
e-mail: vdpokrovskii@itmo.ru

Abstract

Effective time series forecasting remains challenging due to the complex and non-linear nature of real-world temporal data.

In this paper, we propose a novel approach named AutoTopo. It leverages topological concepts to existing AutoMLs to capture the underlying structure of time series data. We aim to improve forecasting performance by enriching the feature space with topological features that describe the shape and patterns of the time series.

Unlike traditional methods that rely on hand-crafted features or increase model complexity, our approach focuses on enhancing the feature field by incorporating topological invariants. We apply techniques from algebraic topology to extract meaningful features that are robust to noise and capture the essential characteristics of the time series. To improve forecasting accuracy further, we also developed an automatic window size estimation algorithm based on autocorrelation, frequency, and statistical information. It allows us to adaptively adjust the window size to better capture the underlying patterns in the data.

The experiments on an M4 benchmark demonstrate the effectiveness of AutoTopo against SOTA. We show that generation of topological features and automatic window size selection could increase forecasting efficiency for existing AutoML approaches.

Introduction

Time series forecasting (TSF) is an essential and challenging task in various industrial, environmental, and social applications (Mahalakshmi, Sridevi, and Rajaram 2016). The proper structure and parameters of modeling pipelines should be designed to obtain an effective forecast. Various models can be used in the pipeline - from classical statistical models to machine learning and deep learning approaches (Lim and Zohren 2021). The pipeline design can be selected using Automated Machine Learning (AutoML), which is especially important for ensemble and hybrid solutions (Hajirahimi and Khashei 2023). While some AutoML frameworks support the time series forecasting task (Alsharef et al. 2022), its predictive performance and practical applicability still need to be improved. The pre-trained deep learning models also have the applications for TSF task (Garza and Mergenthaler-Canseco 2023). However, it can be considered computationally expensive and non-flexible.

In the paper, we aim to propose a lightweight approach for time series forecasting that can handle various multi-scale time series efficiently. To achieve the competitiveness with SOTA solutions, we solve several problems. First, the time series forecasting task adds several hyperparameters to the data preprocessing. The main one is 'window size' for the lagged trajectory matrix. The forecast quality can decrease significantly in case of incorrect window size selection. Also, the multi-scale processes require several different windows in one pipeline. It raises the problem of automated decomposition of time series (Sarafanov, Pokrovskii, and Nikitin 2022).

In the paper, we designed the *AutoTopo* approach to meet challenges and make good forecasts for a wide range of time series. Introducing an algorithm for automatic window size selection solves the first problem. The second problem is solved by (1) decomposing time series to some components that are predicted separately and then composing them to forecasted original time series; (2) using the composition of some machine learning models (3) involvement of topological data analysis; stage to pipeline. The experimental validation is conducted using open benchmark M4 (Makridakis, Spiliotis, and Assimakopoulos 2019) showed effectiveness of proposed approach on several kinds of tasks.

Related works

Statistical approaches

Since the beginning of data-driven time series forecasting, there have been a couple of approaches like autoregressive (AR-) models (Shumway and Stoffer 2017), exponential smoothing models (Gardner Jr. 1985), and other classical models. These are parametric models that rely on domain knowledge. Despite wide machine learning models spreading there are many time series forecasting problems where classical models show themselves well. The most known and widely used classic approaches are: (1) Naive forecast, (2) Moving average forecast, (3) Autoregressive forecast models, and (4) Exponential smoothing models.

These models are simple, but their fitting and tuning require domain knowledge and experience. Also, data-related applicability limitations make those models hard to use in cases with large amounts of time series.

Machine learning approaches

Another way to forecast time series is using machine learning models like linear models, trees, forests, gradient boostings, support vector machines, and so on (Mahalakshmi, Sridevi, and Rajaram 2016; Masini, Medeiros, and Mendes 2023). It requires the preliminary stage of transforming the initial time series to trajectory matrix (or Hankel matrix) (Golyandina, Nekrutkin, and Zhigljavsky 2001) that can be used with classical machine learning models.

It allows using a wide range of existing approaches for regression problems and does not need special time series forecasting experience. All regression models do not have any primary limitations and can be fitted to any time series in a light-weight way. However, the proper design of the ML pipeline is critical to achieving a high-quality forecast.

Deep learning approaches

Deep learning models are also widely used for time series forecasting. The most known model types are used in time series forecasting (Benidis et al. 2022): simple multilayer perceptrons (MLP), convolution neural network (CNN), recurrent neural networks (RNN), and transformers (including pre-trained one (Garza and Mergenthaler-Canseco 2023)).

However, the complicated deep architectures can be considered redundant (Zeng et al. 2023) for time series forecasting cases. For this reason, we take the Neural Basis Expansion Analysis (NBEATS) (Oreshkin et al. 2019) as one of the SOTA deep learning models and use it in comparison. NBEATS is an MLP-based deep neural architecture with backward and forward residual links. The network has two variants: (1) In its interpretable configuration, NBEATS sequentially projects the signal into polynomials and harmonic basis to learn trend and seasonality components; (2) In its generic configuration, it substitutes the polynomial and harmonic basis for identity basis and larger network's depth.

AutoML approaches

The time series forecasting pipeline design can be improved using AutoML (Meisenbacher et al. 2022). There are a lot of solutions that use different approaches and may vary greatly: from AutoArima, which tunes AR-based models and works with time series natively, to AutoGluon (Shchur et al. 2023) or FEDOT (Nikitin et al. 2021) that derives time series forecasting problem to regression problem and construct the ensemble-based pipelines. Work (Jiao et al. 2020) proposes an AutoML approach for time series anomaly detection and classification called TimeAutoML. But their the most contribution is creation algorithm for time series representation learning AutoML algorithm. It based on combining deep learning approach (attention and recurrent neural networks) with Bayesian hyper-parameters optimization. Authors claims representation of time series is provided by algorithm may be useful for time series forecasting tasks too. Despite power of modern neural network architectures its main drawback is lack of interpretability and big computational effort is needed to fit deep learning models.

AutoTopo

We designed and implemented an effective algorithm for the automated topological features generation named *AutoTopo*. It is based on a multi-stage approach: (1) Selecting window size parameter for lag transformation; (2) Performing lag transformation with selected window size; (3) Performing topological features extracting from each lag. Thus, these features can be used as predictors for any regression model: (4) Combining topological features with lag features; (5) Modification of evolutionary algorithm's search space. The overall structure of the Proposed approach is described in Figure 2.

Window size selection

The optimal window size selection is the most critical stage of time series analysis, as it directly affects the quality of subsequent analyses and forecasts. The methods "Dominant Fourier frequency," "Highest Autocorrelation," and "Multi-Window Search" (Ermshaus, Schäfer, and Leser 2023) offer different approaches to solving this problem.

When choosing a method for estimating the optimal window size, practitioners should consider the nature of the time series data and the underlying patterns. Thoughtful choice of window size improves the accuracy and reliability of analyses, contributing to better-informed decision-making in various areas.

Based on these considerations, we chose the auto-correlation (AC) method for determining the optimal window (Algorithm 1).

Algorithm 1: Auto-correlation-based Window Estimation

Require: ts - time series, $window_{\text{min}}, window_{\text{max}}$
1: ts_len $\leftarrow$ length of time_series
2: acf_values $\leftarrow$ compute$(\text{ts}, \text{nlags} = \frac{\text{ts_len}}{2})$
3: peaks, _ $\leftarrow$ find_peaks(acf_values)
4: peaks $\leftarrow$ filter_peaks(peaks, window_min, window_max)
5: **if** no peaks in range$(window_{min}, window_{max})$ **then**
6: **return** $window\_min$
7: **end if**
8: max_peak_index $\leftarrow$ index of peak with maximum AC
9: **return** peak at max_peak_index

This approach is based on the assumption that the lag of the highest auto-correlation coefficient corresponds to the window size that best captures the periodicity of the time series.

For example, if the highest auto-correlation value was observed for n lag, we would take series values from n lag to the first lag, respectively.

For instance, we could consider a tripled sample from the Chinatown dataset (Bagnall et al. 2022). In this case, the AC method determines the optimal window length equal to 24 time points, which is the exact length of every time series in the dataset.

Topological features

Time series-based feature generation methods can be divided into two broad categories: based on entire time series data and

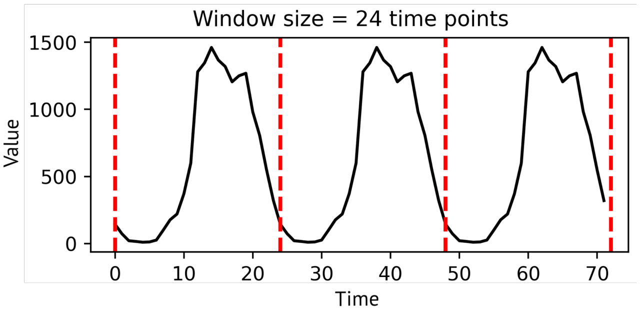

Figure 1: Auto-correlation based window selection result.

based on time series subsequences. The first group includes approaches summarizing the complete set of observations in the time series. The latter extract features from the subsequences of a time series, the so-called embedding vectors. One approach using a complete set of observations is meta-learning to select or combine machine learning models. It is also worth noting that the FRESH method automatically creates features from time series, extracts many features, and selects the most relevant ones.

However, compared to the approaches described above, a tiny part of the methods use time series subsequences, and the potential for systematic application of such approaches needs to be better understood. The main idea of the proposed approach is to use the topological and geometrical properties of input data, for example, its shape or connectivity, to extract informative features, which could be used further by other forecasting algorithms. Persistent homology is one way to describe such features numerically. It considers data as a point cloud and tries to detect the presence of holes using discretization and triangulation of the initial data space with simplicial complexes. A 2D plot called Persistence Diagram (PD) that indicates the birth and death of n-dimensional holes in the induced topological spaces is intended to describe the structure of those holes. Using different features, we can describe PD and use them as feature vectors for the model.

The first step of the topological features generation algorithm is to transform the initial time series f to a Hankel matrix containing a sequence of multidimensional embedding vectors. L be the length of the sliding window, d is stride parameter (by default equal 1). Thus, a set of subsequences is formed $\{f_i, f_{i+d}, ..., f_{i+(L-1)d}\}$ for $i = 0, ..., N - L$:

$$
\begin{aligned}
X_0 &= (f_t, f_{t+d}, f_{t+2d}, ..., f_{L-1}) \\
X_1 &= (f_{t+l}, f_{t+2d}, f_{t+3d}, ..., f_L) \\
&\vdots \\
X_{N-L} &= (f_{N-L}, f_{N-L+d}, f_{N-L+2d}, ..., f_{N-1})
\end{aligned}
$$

The second step is to create a finite metric space (X, d_{metric}), where d_metric is the chosen distance metric. Define the Vietoris-Rips complex of X as the filtered complex $VR_s(X)$ that contains a subset of as a simplex if all pairwise distances in the subset are less than or equal to r:

$$VR_s(X) = \left\{ [v_0, \ldots, v_n] \,\middle|\, \forall i, j \;\; d(v_i, v_j) \leq s \right\} \quad (1)$$

Vietoris-Rips persistence of (X, d_metric), is the persistent simplicial homology of $VR_s(X)$. Several features can be obtained using this simplicial homology:

- *Number of discontinuities*: Number of discontinuities for each homology dimension. In the 0^{th} dimension, the discontinuities indicate connected components, that is, vertices and filtration.
- *Maximum Hole Lifetime*: The maximum discontinuity lifetime identifies the most relevant simplicial complexes.
- *Sum of Betti Numbers*: The Sum of Betti Numbers can be considered an integral of the Betti Curve for each dimension.

After generating topological features, we combined them with lag features (real series values). A combination of different kinds of features is possible due to the composite model structure. In such representation, the pipeline could be represented as a directed acyclic graph. This pipeline is passed to an evolutionary algorithm as an initial assumption.

The search space of the evolutionary algorithm was also modified. Topological features operation was added to the list of candidates for mutation operation. Also, default window size for the new lagged operation was changed according to the estimated value.

Utilizing topological features is not a bullet solution. For this reason, this graph was used as one of the initial assumptions for AutoML. The evolutionary algorithm underlying the solution decides automatically which pipeline structure is better for the particular time series.

Experimental studies

Experimental setup

The setup includes a random sub-sample (700 series) from widely known time series forecasting benchmark M4. Yearly, quarterly, monthly, weekly, and daily frequencies were chosen. Selected forecast length corresponds to horizons proposed by benchmark creators (14 for daily, 18 for monthly, 8 for quarterly, 13 for weekly, and 6 for yearly).

We compared *AutoTopo* with a basic evolutionary-based AutoML algorithm, the evolutionary algorithm with only window size selection, static pipeline with window size selection and topological features generation, AutoGluon (one of the SOTA AutoML solutions for time series), and NBEATS (effective light-weight deep model). The experimental part of the paper does not aim to cover all SOTA solutions based on deep learning since it can be complicated to apply it in practice (Zeng et al. 2023).

The results obtained demonstrated that analyzed TSF approaches had similar performance on a large number of series. Due to that, series were filtered by selecting series where solutions had reasonable differences in their predictions. Thus, 530 series remained. All approaches ran once on each time series.

We configured a server based on AMD Ryzen 7 5800X (3.80 GHz) and 16GB memory for experiments.

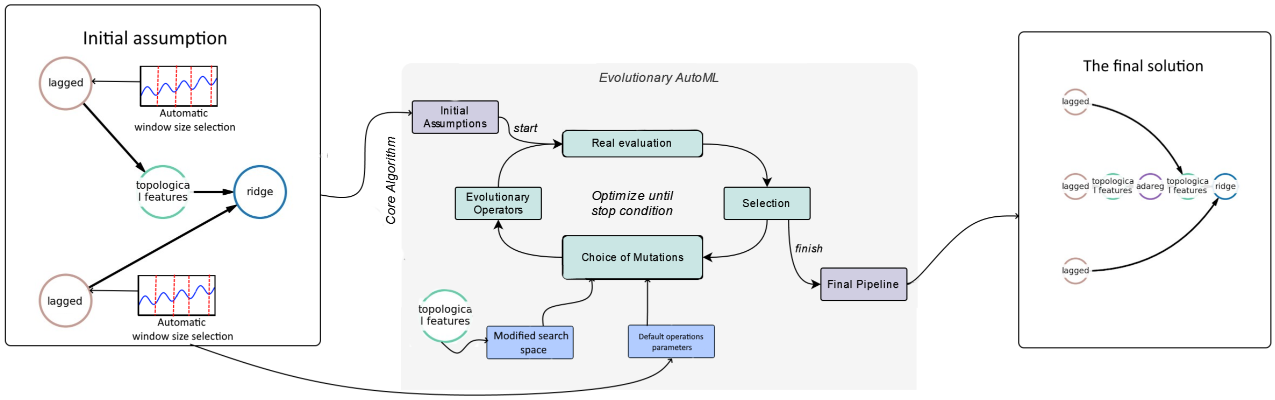

Figure 2: The overall structure of *AutoTopo*. Firstly, the optimal window size for lag operation is computed. Then, topological features are calculated for the derived lagged table. The following topological features and data features (the same lagged table) are combined as features for the ridge estimator. Next, such a pipeline is used as an initial assumption for a modified evolutionary algorithm. Finally, a composed and tuned pipeline is returned as a result.

Table 1: Quantiles of MASE metric values for each period. Less value is better. The best values in columns are highlighted. The algorithm of window size selection is denoted as WSS. The algorithm of window size selection with topological features generation is denoted as AutoTopo

	MASE																								
Solution	**Daily**					**Weekly**					**Monthly**					**Quarterly**					**Yearly**				
Quantile	0,1	0,25	0,5	0,75	0,9	0,1	0,25	0,5	0,75	0,9	0,1	0,25	0,5	0,75	0,9	0,1	0,25	0,5	0,75	0,9	0,1	0,25	0,5	0,75	0,9
NBEATS	0,9	**1,36**	**2,24**	4,05	**6,16**	0,91	1,21	1,77	2,93	5,3	0,81	1,31	2,21	4,11	8,78	0,86	1,32	2,21	4,84	11,37	1,07	2,88	4,08	6,39	8,66
Autogluon	1,18	2,08	3,2	5,89	9,42	0,71	1,08	2,07	3,83	5,87	0,73	1,13	1,84	3,43	6,82	0,67	1,03	1,86	2,98	**5,56**	1,12	1,33	**2,21**	4,23	**6,03**
AutoML	1,05	1,5	2,78	3,79	6,8	0,63	1	1,41	2,89	4,59	**0,71**	**0,98**	1,86	3,46	6,3	0,72	1,05	1,77	2,89	5,69	1,07	1,45	3,19	4,8	6,71
Static pipeline + AutoTopo	1,07	1,62	2,84	4,77	7,69	**0,54**	**0,79**	1,46	2,74	5,03	0,78	1,21	2,13	4,15	8,72	0,68	1,01	1,73	3,05	6,57	0,82	**1,25**	2,7	4,15	5,51
AutoML + WSS	**0,81**	1,43	2,75	4,16	7,55	0,57	0,86	1,41	2,55	4,04	0,78	1,15	2,07	4,17	7,74	0,72	1	**1,5**	3,01	5,96	1,06	1,43	2,7	4,29	6,94
AutoML + AutoTopo	0,84	**1,36**	2,34	**3,68**	6,43	0,55	0,87	**1,27**	**2,34**	**3,84**	0,74	0,99	**1,58**	**2,98**	**6,25**	**0,56**	**0,85**	1,56	**2,83**	6,3	**0,79**	1,45	2,3	**3,99**	6,4

Benchmarking

We evaluated results by MASE metric.

$$\text{MASE} = \frac{\sum_{i=1}^{n} |Y_i - e_i|}{\frac{n}{n-1} \sum_{i=2}^{n} |Y_i - Y_{i-1}|} \quad (2)$$

After 0.1, 0.25, 0.5, 0.75. 0.9 quantiles were calculated for each framework. Quantiles are shown in Table 1.

Prediction comparison visualizations are shown in Figure 3. Wilcoxon paired test was used to estimate the significance of the results. AutoTopo's approach significantly differs from NBEATS on weekly, quarterly, monthly, and yearly data, and a comparison with the AutoGluon framework showed significant differences on daily, weekly, quarterly, and monthly data. We also calculated the number of wins for each solution. ("Win" implies the smallest metric on a particular time series compared to other solutions). AutoTopo approach obtained 125 wins, NBEATS - 87 wins, AutoGluon - 86 wins.

We also considered time elapsing for proposed approach. Single pipeline contains topological features extraction node (initial assumption from Fig.2) with fixed structure was evaluated (train and predict stage) with window sizes 10, 20, 50, 150, 250, 500. Time for each evaluation was clocked. Fig.4 shows time dependency from window size. Experiment showed that topological features generation may elapse quite big time comparing with classic approaches, but time does not grow fast with window size increasing.

Conclusions

In the paper, we propose the *AutoTopo* approach for increasing AutoML performance that combines the zero-shot estimation appropriate window size method for time series forecasting algorithms, topological features extractor, and evolutionary algorithm.

Analysis of obtained results. The results can be considered a proof-of-concept for a multi-stage approach with automatic window size selection and utilizing topological features as feature space for the evolutionary algorithm. It confirms that the proposed approach made it possible to increase performance in different kinds of tasks significantly.

Limitations. Despite efficiency on several kinds of tasks, we do not claim that it is a silver bullet solution for any task. We noticed that the proposed approach works well with time series with non-trivial but strongly periodical components. We assume that future research with meta-learning techniques

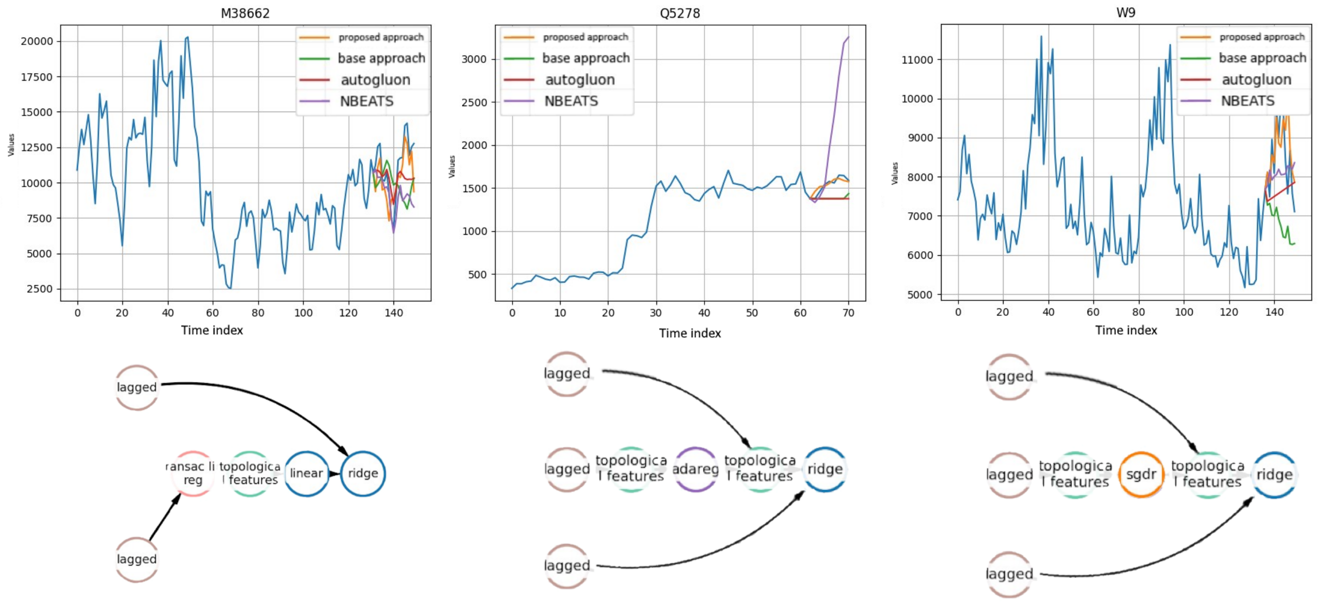

Figure 3: Examples of obtained topological-based pipelines. Forecasts are shown at the top; best pipeline structures are at the bottom. If the time series length is bigger than 150 elements, only the last 150 elements are shown. Decription of names of nodes: lagged - operation of modifying time series in trajectory matrix; ransac lin reg - RANSAC algorithm; linear - linear regression; ridge - linear regression with L2 regularization; adareg - AdaBoost regressor; sqdr - linear model with stochastic gradient descent.

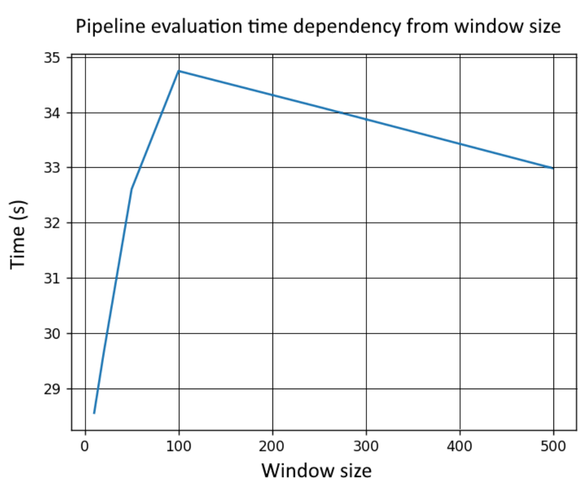

Figure 4: Time dependency (in seconds) from estimated window size.

could provide the ability to select the suitable strategy for each series in a zero-shot way.

Code and data availability

The software implementation of the proposed algorithms and scripts for the conducted experiments are available in the open repositories https://github.com/aimclub/FEDOT and https://github.com/aimclub/Fedot.Industrial. Both of it are under the BSD-3 license.

All datasets, scripts and results for experimental studies are available in the repository https://github.com/ITMO-NSS-team/aaai24_auto_topo.

Acknowledgements

This work was supported by the Analytical Center for the Government of the Russian Federation (IGK 000000D730321P5Q0002), agreement No. 70-2021-00141.

References

Alsharef, A.; Aggarwal, K.; Sonia; Kumar, M.; and Mishra, A. 2022. Review of ML and AutoML solutions to forecast time-series data. *Archives of Computational Methods in Engineering*, 29(7): 5297–5311.

Bagnall, A.; Keogh, E.; Lines, J.; Bostrom, A.; Large, J.; and Middlehurst, M. 2022. UEA & UCR Time Series Classification Repository.

Benidis, K.; Rangapuram, S. S.; Flunkert, V.; Wang, Y.; Maddix, D.; Turkmen, C.; Gasthaus, J.; Bohlke-Schneider, M.; Salinas, D.; Stella, L.; et al. 2022. Deep learning for time series forecasting: Tutorial and literature survey. *ACM Computing Surveys*, 55(6): 1–36.

Ermshaus, A.; Schäfer, P.; and Leser, U. 2023. Window Size Selection in Unsupervised Time Series Analytics: A Review and Benchmark. In *International Workshop on Advanced Analytics and Learning on Temporal Data*, 83–101. Springer.

Gardner Jr., E. S. 1985. Exponential smoothing: The state of the art. *Journal of Forecasting*, 4(1): 1–28.

Garza, A.; and Mergenthaler-Canseco, M. 2023. TimeGPT-1. *arXiv preprint arXiv:2310.03589*.

Golyandina, N.; Nekrutkin, V.; and Zhigljavsky, A. 2001. *Analysis of Time Series Structure: SSA and Related Techniques*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. CRC Press. ISBN 9781420035841.

Hajirahimi, Z.; and Khashei, M. 2023. Hybridization of hybrid structures for time series forecasting: A review. *Artificial Intelligence Review*, 56(2): 1201–1261.

Jiao, Y.; Yang, K.; Dou, S.; Luo, P.; Liu, S.; and Song, D. 2020. TimeAutoML: Autonomous Representation Learning for Multivariate Irregularly Sampled Time Series. *ArXiv*, abs/2010.01596.

Lim, B.; and Zohren, S. 2021. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194): 20200209.

Mahalakshmi, G.; Sridevi, S.; and Rajaram, S. 2016. A survey on forecasting of time series data. In *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, 1–8.

Makridakis, S.; Spiliotis, E.; and Assimakopoulos, V. 2019. The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36.

Masini, R. P.; Medeiros, M. C.; and Mendes, E. F. 2023. Machine learning advances for time series forecasting. *Journal of economic surveys*, 37(1): 76–111.

Meisenbacher, S.; Turowski, M.; Phipps, K.; Rätz, M.; Müller, D.; Hagenmeyer, V.; and Mikut, R. 2022. Review of automated time series forecasting pipelines. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(6): e1475.

Nikitin, N. O.; Vychuzhanin, P.; Sarafanov, M.; Polonskaia, I. S.; Revin, I.; Barabanova, I. V.; Maximov, G.; Kalyuzhnaya, A. V.; and Boukhanovsky, A. 2021. Automated evolutionary approach for the design of composite machine learning pipelines. *Future Generation Computer Systems*.

Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting.

Sarafanov, M.; Pokrovskii, V.; and Nikitin, N. O. 2022. Evolutionary automated machine learning for multi-scale decomposition and forecasting of sensor time series. In *2022 IEEE Congress on Evolutionary Computation (CEC)*, 01–08. IEEE.

Shchur, O.; Turkmen, C.; Erickson, N.; Shen, H.; Shirkov, A.; Hu, T.; and Wang, Y. 2023. AutoGluon-TimeSeries: AutoML for Probabilistic Time Series Forecasting. In *International Conference on Automated Machine Learning*.

Shumway, R. H.; and Stoffer, D. S. 2017. *ARIMA Models*, 75–163. Cham: Springer International Publishing. ISBN 978-3-319-52452-8.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11121–11128.