

# TS-MULE: Local Interpretable Model-Agnostic Explanations for Time Series Forecast Models

Udo Schlegel<sup>1</sup>, Duy Lam Vo<sup>1</sup>, Daniel A. Keim<sup>1</sup>, and Daniel Seebacher<sup>1</sup>

University of Konstanz, Germany  
u.schlegel@uni-konstanz.de

**Abstract.** Time series forecasting is a demanding task ranging from weather to failure forecasting with black-box models achieving state-of-the-art performances. However, understanding and debugging are not guaranteed. We propose *TS-MULE*, a local surrogate model explanation method specialized for time series extending the LIME approach. Our extended LIME works with various ways to segment and perturb the time series data. In our extension, we present six sampling segmentation approaches for time series to improve the quality of surrogate attributions and demonstrate their performances on three deep learning model architectures and three common multivariate time series datasets.

**Keywords:** Explainable AI · LIME · Time Series

## 1 Introduction

Time series forecasting is an essential task with applications in a broad range of domains, such as industrial process control, finance, and risk management, since predicting future trends and events is a critical input into many types of planning and decision-making processes [1]. Recently, deep learning methods have increasingly found their way into the field of time series forecasting as a result of their successful application in other domains such as natural language processing [2] and object detection [3]. A major drawback of such models is that, due to their non-linear, multi-layered structure, they are black box models that suffer from a lack of explainability. Such a lack of explainability prevents deep learning from being used in production in sensitive domains, such as healthcare [4], as opposed to statistical methods [5], or is complicated by laws, such as the EU General Data Protection Regulation [6], which enforces a right for explanations. Thus, agencies such as DARPA introduced the explainable AI (XAI) initiative [7] to promote the research around interpretable Machine Learning (ML).

Gaining the necessary understanding of these complex models to provide explanations globally for the whole input space is often infeasible, leading to the development of methods that provide only local explanations of the underlying prediction function, such as LIME [8]. LIME is an XAI technique that can explain the predictions of any classifier by learning and providing an interpretable surrogate model around the classification. An advantage of LIME in terms of interpretability is that it perturbs the input by changing components that make

sense to humans (e.g., words or parts of an image), even if the model is using much more complicated components as features (e.g., word embeddings) [8].

For images, such interpretable components can be superpixels, which are a perceptual grouping of pixels, or for texts, it can be individual words or sentences. However, finding such semantically meaningful components for univariate or even multivariate time series data is not trivial. Segmenting the time series into fixed-width windows might miss meaningful elements between windows by weighting them equally or are larger or smaller than the chosen window size. Thus, such a fixed segmentation can potentially miss important subsequences in the time series by splitting them. One possible approach could identify motifs in the time series. Such motifs are subsequences of the time series very similar to each other. However, even optimized algorithms can have a worst-case complexity of  $\mathcal{O}(n^2)$  [9] and are, thus, not suitable to identify potential patterns beforehand.

To tackle such issues, we propose *TS-MULE*, an extension to LIME by improving the segmentation, for local explanations of univariate and multivariate time series. We provide five novel algorithm approaches to provide a meaningful segmentation of time series to enable local interpretable model-agnostic explanations of time series forecasting models. To provide such meaningful segmentation, we incorporate the matrix profile [10] as well as the SAX transformation [11] and extend the results of these algorithms with binning or top-k approaches to incorporate the findings of these techniques. We evaluate these segmentation algorithms against each other and the baseline of a uniform segmentation on three standard forecasting datasets with three different black-box models.<sup>1</sup>

## 2 Related Work

An important distinction when selecting methods for explaining complex machine learning models is for which user group these XAI methods must be accessible. Most of the proposed XAI methods used, especially for time series deep learning models, are usually only accessible to model developers. For instance, by examining the activation of latent layers [12], or via relevance backpropagation [13]. However, especially for other groups, particularly model users (see Spinner et al. for an overview of user groups [14]), such approaches are less practical since explanations need to be provided at a higher level of abstraction. Available approaches with a higher level of abstraction currently come primarily from the computer vision domain for explaining image classifications [15].

There are already first works that apply these concepts in time series classification and prediction. For example, the approach of Suresh et al. [16] replaces each time series observation with uniform noise to study the impact on model performances and thus determine feature importance. Since replacing features with out-of-domain noise can lead to arbitrary changes in model output, Tonekaboni et al. use data distributions to produce reliable counterfactuals [17]. Both previous approaches rely on observation-level replacement and thus, cannot identify important larger patterns in time series. Two recent approaches tackle this

<sup>1</sup> Source code and evaluation results are available at: <https://github.com/dbvis-ukon/ts-mule>

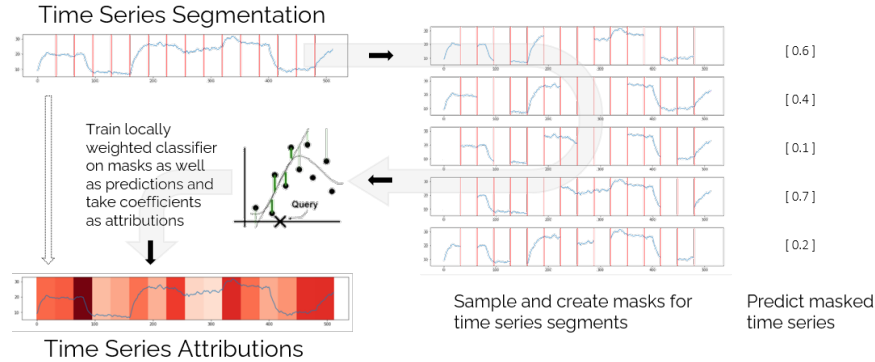
issue by using longer time segments as input for the perturbation and replacing it with, for instance, linear interpolations, constant values or segments from other time series [18], or with zeros, local or global mean values, or local or global noise [19]. However, both of these approaches rely on fixed window sizes. Thus they are incapable of modeling, e.g., semantically meaningful patterns in the time series, which can have variable lengths. Additionally, they might miss important patterns if the predefined window size is smaller or longer than the pattern or if patterns lie between the fixed time segments.

Hence, we provide an extension of the LIME approach to identify superpixels-like patterns, i.e., semantically related data regions, in time series data. This paper presents a set of suitable segmentation algorithms and evaluates their suitability for providing explanations under various data characteristics.

### 3 Post-hoc local explanations with LIME

Creating explanations for decisions of black-box models has various alternatives. One of these possibilities is the post-hoc approach LIME by Ribeiro et al. [8]. Local Interpretable Model-Agnostic Explanations, shortly LIME, uses an interpretable surrogate model to create explanations for black-box models. In the first step, a chosen sample to explain and a model to be explained are given as input to the approach. The sample is then segmented by a previously chosen segmentation algorithm, e.g., a superpixel segmentation for images [8]. LIME then creates masks for the sample deactivating segments or replacing them with non-informative values. In many cases, this step is called perturbation and is something different than the perturbation mentioned later. These newly generated (perturbed) samples are predicted with the input model to get new predictions. LIME collects these predictions and trains a new interpretable classifier, often a linear model, on the masks with the predictions as the target. In the case of a linear model, the coefficients are used to weigh the different input segments and to explain the model for the given sample. Fig. 1 demonstrate the described approach on time series with a uniform segmentation.

LIME is generally applicable for any data type, but there are some challenges due to the necessity of segmentation. Valuable segmentation makes sense to humans as it incorporates their domain knowledge. For instance, superpixel segmentation identifies perceptual groups in images, which in most cases correspond to a human interpretable object. As time series are generally hard to segment without domain knowledge, a general approach is rather difficult, even with domain knowledge not applicable. A forecasting black box model often just uses a window as input to predict the target value in many cases. Such a window is fixed beforehand and slides over the data, thus having no strict segmentation in itself. Finding such segmentation is a significant challenge for time series as it needs to be generally applicable.



**Fig. 1.** The LIME approach applied on time series starting with the uniform segmentation on a time series sample. Next, doing the masking, perturbing, and predicting step of LIME to generate more local samples. Afterward, a linear interpretable model is trained on the masks and predictions using local weighting. At last, extracting the coefficients of the model leads to the wanted attributions for the initial sample.

## 4 Finding suitable segmentation mappings

We propose *TS-MULE*, extending the LIME [8] approach for time series with novel segmentation algorithms. Our approach presents five segmentation techniques created for time series and three different replacement strategies.

### 4.1 Using static windows

*Uniform segmentation* is the most basic method to segment a time series into windows. In this approach, we split the time series  $ts = \{t_0, t_1, t_2, \dots, t_n\}$  into equally and non-overlapping  $m$ -sized windows  $ws = \{w_0, w_1, w_2, \dots, w_d\}$  with  $d = \lceil n/m \rceil$ . If  $n$  is not a multiple of  $m$ , the final windows may have more or less time points. We expand the uniform segmentation to exponential windows, which ignores the size  $m$  and has longer windows at the end. A time series  $ts$  in *exponential segmentation* is split into  $d = \log(n)$  windows and its length increases with  $[e^0], [e^1], [e^2], \dots, [e^d]$ . To cover all the points of the time series, in the final window, we adjust its length by  $n - \sum([e^0], [e^1], [e^2], \dots, [e^{d-1}])$ . A benefit of such segmentation is that we put more weight on the latest points with longer windows.

### 4.2 Using the Matrix Profile

A matrix profile is a vector that stores the z-normalized Euclidean distance between any subsequence within a time series and its nearest neighbor [10]. Such a matrix profile can be used to identify motifs as well as outlier subsequences in large time series [10]. We introduce the *slope* and *bin* segmentation based on the matrix profile on time series to incorporate local trends and patterns.

The *slope segmentation* has the parameters window size  $m$  as input for the matrix profile and  $k$  for the number of partitions for the segmentation. The basic idea behind this segmentation approach consists in the opportunity to find patterns in the time series using the matrix profile. By further focusing on the slope of the matrix profiles distances, we can identify drastic changes in the nearest neighbors to find not only possible patterns but also uncommon changes in the time series itself. Such a uncommon changes can be used as plausible splits for the segmentation as the pattern are still included in the segments. We calculate the matrix profile  $mp = \{d_1, d_2, \dots, d_j\}$  with our previously adjusted window size  $m$  so that  $j = n - m + 1$  to find interesting distances, e.g., to identify motifs. Afterward, we either calculate the gradient on the resulting matrix profile  $\nabla mp$  and take the absolute value  $|\nabla mp|$  to identify peaks as steep slopes. Or, depending on the configuration, we sort the resulting matrix profile vector ascending and compute the slope to identify jumps in distances to find significant changes in the time series. We sort the resulting vector in both cases and take the  $k$ -largest values to find segment borders. The time series indices of these  $k$  values segment our time series and describe drastic changes in the time series.

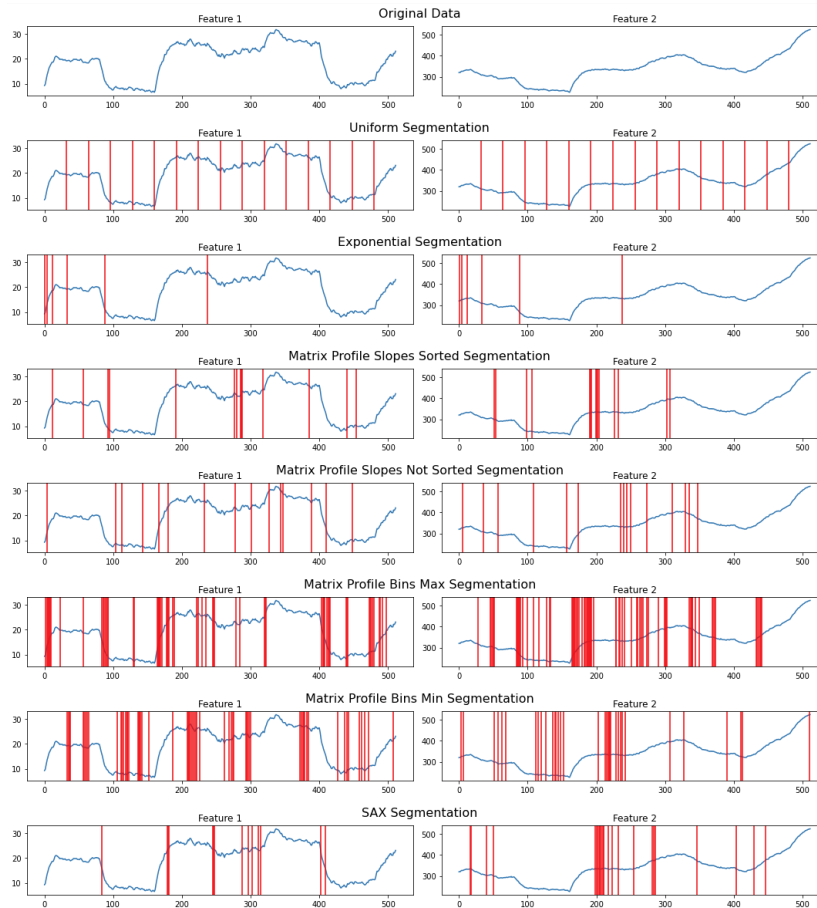
We further present *bin segmentation* based on the matrix profile with the same parameters  $m$  and  $k$  as above. Again the idea behind this approach enables finding patterns in the time series by not using the gradient to find drastic changes in the nearest neighbor but using bins to combine similar distances in the matrix profile to segments. We calculate and sort the matrix profile again. However, we further split the min-max range of the matrix profile into  $k$ -bins. Afterward, we label the  $k$ -bins numerically so that lower numbers have a low and higher a high matrix profile. We convert our matrix profile to the corresponding bin number and assign our base value to the *max* or *min* bin. Next, we slide over the resulting profile with a window length  $m$ . Due to the sliding window approach, a time point can be either in the segment  $seg_i$  or  $seg_j$ . For our *bins-min segmentation*, we assign the time point  $t_t$  to  $seg_i$  if  $bin_i$  is smaller than  $bin_j$ . Our *bins-max segmentation*, oppositely, uses the  $seg_i$  if  $bin_i$  is larger than  $bin_j$ .

### 4.3 Using the SAX transformation

*SAX segmentation* introduces a segmentation based on horizontal binning of a time series with  $k$  partitions as the parameter. The basic idea behind this segmentation approach includes the changes in the range of the values by splitting the overall distribution of possible values into bins. The SAX transformation [11] converts a time series  $ts$  into a sequence of symbols  $sa = \{s_0, s_1, s_2, \dots, s_n\}$  with  $s_i \in \{a, b, c, \dots\}$  based on a continuous binning of intervals in the vertical direction. We incorporate a base number of bins  $b = 3$  for the SAX algorithm and use repeating symbols as segments, e.g.,  $sa = \{a_1, a_2, a_3, b_1, b_2, a_1, c_1, c_2, c_3\}$  involves four segments leading to  $\{a_1, a_2, a_3\}$ ,  $\{b_1, b_2\}$ ,  $\{a_1\}$ ,  $\{c_1, c_2, c_3\}$ . At each iteration, the amount of bins is increased  $b+ = 1$  to finally achieve a previously selected  $k$  partitions as more bins generally convert to more partitions. For some cases, the exact partition size is not possible, and we allow a difference of ten percent to the selected partition size to mitigate such edge cases.

#### 4.4 Comparing the segmentation algorithms

Existing and proposed segmentation algorithms lead to different segments representing potentially suitable techniques for various data sets. Fig. 2 presents these algorithms on two differently scaled time series features. Especially, comparing the uniform segmentation with the others demonstrates the advantages of the other approaches. Depending on the algorithm, different segments are visible and present some more focused parts of the time series samples. Choosing from a broader range of techniques can lead to improved explanations for humans.



**Fig. 2.** Comparison of the different segmentation variants. Red stripes show segment splits. Some segmentation algorithms proposed end up with more as well as very short segments than the uniform segmentation with default parameters.

## 5 Evaluating *TS-MULE* on time series forecasting

The evaluation of our proposed segmentation and perturbation approaches is based on the perturbation analysis for fidelity by Schlegel et al. [15, 20] adapted to forecasting tasks using the mean squared error. As datasets for our evaluation, we use the Beijing Air Quality 2.5, Beijing Multi-Site Air Quality, and the Metro Interstate Traffic data to show the results on divers multivariate time series. For the air quality datasets, we use a fixed input size of 24. The metro traffic forecasting has an input length of 72. We use three different basic implementations of black-box models: a basic one-dimensional convolutional neural network, a deep neural network, and a recurrent neural network (LSTMs [21]).

The perturbation analysis by Schlegel et al. [20] consists of three steps: explanation generation, data perturbation based on explanations, and perturbation evaluation. At first, a selected dataset, e.g., the test data, is evaluated with a quality metric (e.g., accuracy), and explanations are generated for every sample. Next, every sample of the selected dataset is perturbed such that time points with high relevances for the explanation are replaced with non-information holding values. As non-information holding values for time series are challenging to find, we focus on the proposed ones (zero, inverse, mean) by Schlegel et al. [20]. Often the high relevance attributions are identified by using a threshold. Lastly, the perturbed data gets evaluated, and the quality metric change is calculated. The assumption is that a value change of the predicted data at highly relevant input positions decreases the quality metric performance of the model as the data loses valuable information. Such an assumption then leads to the conclusion that a working XAI technique decreases the performance more than a random change.

Zero	Beijing Air Quality 2.5			Beijing AQ Multi Site			Metro Interstate Traffic		
	CNN	DNN	RNN	CNN	DNN	RNN	CNN	DNN	RNN
Uniform	<b>2.31</b>	<b>4.24</b>	2.32	1.50	9.00	7.67	2.43	0.22	6.55
Exponential	0.56	1.12	1.41	0.62	0.16	<b>11.52</b>	0.55	0.01	0.62
Slopes	1.31	2.11	1.95	1.3	6.76	3.97	<b>3.39</b>	0.18	<b>9.29</b>
Bins Min	0.35	3.43	<b>3.6</b>	0.41	<b>10.46</b>	5.71	1.25	0.4	7.38
Bins Max	1.69	1.22	2.38	<b>1.52</b>	1.68	2.67	1.44	0.44	2.68
SAX	1.24	2.58	2.23	1.10	8.00	4.15	1.55	<b>1.16</b>	7.34

**Table 1.** Evaluation results of the perturbation analysis for every segmentation technique for three datasets and three models. We calculate the perturbation analysis results based on the percentage change to the original prediction and the randomized change. A larger value shows a better explanation.

We extend the assumptions to calculate a score for improved comparability of the results by focusing on the percentage increase in relation to a random change of the time series. Schlegel et al. [20] propose to take the 90th percentile value of the attribution values of the sample as a threshold. However, we have to scale our

*TS-MULE* values because we observed that depending on the segment count, the distribution of the attribution changes. Such a distribution change leads to either more or less highly relevant time points for the perturbation as, e.g., there are more attribution values above the threshold value. Thus, we take the initial prediction scores *orig*, the perturbed prediction scores *pert*, and the random position change prediction score *rand* and calculate the increase of the perturbed:  $pert_c = \frac{pert - orig}{orig}$  and random:  $rand_c = \frac{rand - orig}{orig}$ . We set these in relation to get our final score:  $score = \frac{|pert_c|}{|rand_c|}$ . A score below one depicts a worse performance than random guessing. Scores larger than one illustrate plausible explanations better than guessing. Through this scaling, the segmentation algorithms can be compared. Larger results demonstrate better segmentation. Table 1 presents such a perturbation analysis on fidelity with our proposed segmentation approaches.

Our preliminary results for a zero perturbation, see Table 1, show that *uniform* is working well for short time series windows (Beijing Air Quality with 24) while *slopes* generate better performances on long windows (Metro Interstate Traffic with 72). However, also our proposed *bins-min*, *bins-max* and *SAX* illustrate promising results for short windows and can be further tuned by adding more parameters. Also, by further adding a minimum length for segments, these algorithms can be improved. The DNN for the Metro Interstate Traffic dataset is interesting as non of the proposed segmentation strategies seem to work. However, such an effect can be caused as the model’s performance is way worse than the other two models. In general, the *uniform* segmentation works well as a starting point, but exchanging it with our proposed algorithms enables more diverse and improved attributions.

## 6 Conclusion

We present *TS-MULE*, a local interpretable model-agnostic explanation extraction technique for time series. For *TS-MULE*, we extend the LIME approach with novel time series segmentation techniques and replacement methods to enforce a better non-informed values exchange. Thus, we contribute five novel time series segmentation algorithms and the *TS-MULE* framework for time series forecasting. We show on three forecasting datasets that *TS-MULE* performs better than randomly perturbing data and thus reveals relevant input values for the prediction of a model. Further, we demonstrate that our proposed segmentation algorithms lead to improved attributions in most cases. As future work, we want to compare the performance of *TS-MULE* against other XAI techniques applied to time series in the framework of Schlegel et al. [20]. We also want to identify shapelets to generate segments with more in-depth domain knowledge and to investigate into similar attribution techniques like SHAP [22].

## Acknowledgements

This work has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 826494.



## References

1. Douglas C Montgomery, Cheryl L Jennings, and Murat Kulahci. *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.
2. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *Advances in Neural Information Processing Systems*, 30, 2017.
3. Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.
4. Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.
5. Mooi Choo Chuah and Fen Fu. Ecg anomaly detection via time series analysis. In *International Symposium on Parallel and Distributed Processing and Applications*, pages 123–135. Springer, 2007.
6. European Union. European General Data Protection Regulation. Technical report, 2018.
7. Gunning, D. Explainable Artificial Intelligence (XAI) DARPA-BAA-16-53. Technical report, 2016.
8. Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144. ACM, 2016.
9. Abdullah Mueen, Eamonn J. Keogh, Qiang Zhu, Sydney Cash, and M. Brandon Westover. Exact Discovery of Time Series Motifs. In *SIAM International Conference on Data Mining SDM*, 2009.
10. Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *IEEE International Conference on Data Mining*, 2016.
11. Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2003.
12. Shoaib Ahmed Siddiqui, Dominique Mercier, Mohsin Munir, Andreas Dengel, and Sheraz Ahmed. Tsviz: Demystification of deep learning models for time-series analysis. *IEEE Access*, 7:67027–67040, 2019.
13. Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*, 2015.
14. Thilo Spinner, Udo Schlegel, Hanna Schäfer, and Mennatallah El-Assady. expAIner: A Visual Analytics Framework for Interactive and Explainable Machine Learning. *IEEE Transactions on Visualization and Computer Graphics*, 2019.
15. Udo Schlegel, Hiba Arnout, Mennatallah El-Assady, Daniela Oelke, and Daniel A. Keim. Towards a Rigorous Evaluation of XAI Methods on Time Series. In *ICCV Workshop on Interpreting and Explaining Visual Artificial Intelligence Models*, 2019.

16. Harini Suresh, Nathan Hunt, Alistair Johnson, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. Clinical intervention prediction and understanding using deep networks. *arXiv preprint arXiv:1705.08498*, 2017.
17. Sana Tonekaboni, Shalmali Joshi, David Duvenaud, and Anna Goldenberg. Explaining time series by counterfactuals, 2020.
18. Maël Guillemé, Véronique Masson, Laurence Rozé, and Alexandre Termier. Agnostic local explanation for time series classification. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 432–439. IEEE, 2019.
19. Felix Mujkanovic, Vanja Doskoč, Martin Schirneck, Patrick Schäfer, and Tobias Friedrich. timexplain—a framework for explaining the predictions of time series classifiers. *arXiv preprint arXiv:2007.07606*, 2020.
20. Udo Schlegel, Daniela Oelke, Daniel A Keim, and Mennatallah El-Assady. An empirical study of explainable AI techniques on deep learning models for time series tasks. *Pre-registration workshop NeurIPS*, 2020.
21. Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8), 1997.
22. Scott Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, 2017.