Finding the DeepDream for Time Series: Activation Maximization for Univariate Time Series

Udo Schlegel[1,*], Daniel A. Keim[1] and Tobias Sutter[1]

[1]University of Konstanz, Universitätsstraße 10, Konstanz, 78464, Germany

Abstract

Understanding how models process and interpret time series data remains a significant challenge in deep learning to enable applicability in safety-critical areas such as healthcare. In this paper, we introduce *Sequence Dreaming*, a technique that adapts Activation Maximization to analyze sequential information, aiming to enhance the interpretability of neural networks operating on univariate time series. By leveraging this method, we visualize the temporal dynamics and patterns most influential in model decision-making processes. To counteract the generation of unrealistic or excessively noisy sequences, we enhance *Sequence Dreaming* with a range of regularization techniques, including exponential smoothing. This approach ensures the production of sequences that more accurately reflect the critical features identified by the neural network. Our approach is tested on a time series classification dataset encompassing applications in predictive maintenance. The results show that our proposed *Sequence Dreaming* approach demonstrates targeted activation maximization for different use cases so that either centered class or border activation maximization can be generated. The results underscore the versatility of *Sequence Dreaming* in uncovering salient temporal features learned by neural networks, thereby advancing model transparency and trustworthiness in decision-critical domains.

Keywords

Explainable AI, Time Series Classification, Activation Maximization

1. Introduction

Techniques such as Activation Maximization [1] and DeepDream [2] have emerged as approaches to make the complex inner workings of deep neural networks (DNNs) more transparent. These methods illuminate the black box of neural networks by visualizing what neural networks learn, significantly improve model interpretability, and provide valuable insights into diagnosing model behavior [3]. Activation Maximization focuses on identifying the input patterns that maximize the response of specific neurons or layers, revealing the features and patterns a model perceives as most salient [1]. Meanwhile, DeepDream leverages the layers of neural networks to generate intricate, dream-like (unreal-looking) images that highlight the learned features in a visually compelling way [2]. Together, these techniques enhance our understanding of how neural networks process information and guide the development of more transparent, effective, and interpretable AI systems. Through the lens of Activation Maximization and DeepDream,

TempXAI@ECML-PKDD'24: Explainable AI for Time Series and Data Streams Tutorial-Workshop, Sep. 9th, 2024, Vilnius, Lithunia

*Corresponding author.

u.schlegel@uni.kn (U. Schlegel); daniel.keim@uni.kn (D. A. Keim); tobias.sutter@uni.kn (T. Sutter)

https://www.vis.uni-konstanz.de/mitglieder/schlegel/ (U. Schlegel)

0000-0002-8266-0162 (U. Schlegel); 0000-0001-7966-9740 (D. A. Keim); 0000-0003-1226-6845 (T. Sutter)

we can unravel the intricacies of neural networks, paving the way for advancements in AI that are comprehensible [3].

In the evolving landscape of neural network interpretation, adapting Activation Maximization for time series data extends the understanding of how deep learning models perceive and process temporal information. *Sequence Dreaming* enables visualization of the intricate temporal features and dynamics the network has learned to recognize by manipulating time series data to amplify the patterns that maximally or targeted activate specific neurons within a model. This method sheds light on the model's decision-making process and unveils the temporal sequences and patterns deemed most significant by the neural network. In doing so, *Sequence Dreaming* bridges the gap between the opaque decision-making of deep learning models and the tangible insights they derive from sequential data, offering another lens through which to interpret and refine models trained on time series similar to shapelet learning [4]. This approach promises to enhance model transparency, facilitate diagnostic analysis, and inspire the development of models for handling the complexities of sequential data analysis.

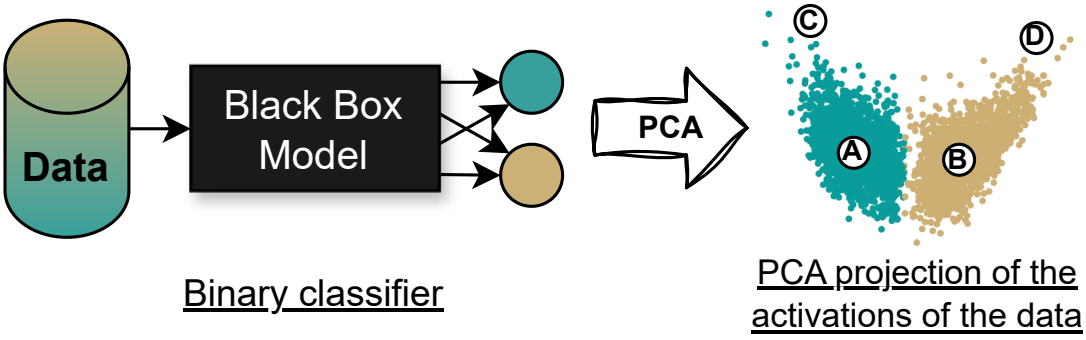
In this paper, we introduce *Sequence Dreaming*, an adaptation of the activation maximization techniques for DeepDream [2, 5] explicitly tailored for time series data. By applying this method, we aim to enhance the interpretability of deep learning models that process time series data, shedding light on the temporal dynamics or patterns these models capture. To refine and control the generation of artificial time series that maximize neuron activations, we extend *Sequence Dreaming* with a suite of regularization techniques, including an α-norm, total variation, time point smoothness, Gaussian smoothing, and random noise reinitiation to produce more realistic and informative visualizations. These modifications are designed to mitigate overfitting to noise and emphasize the underlying patterns critical for model decisions. We test our approach on a time series classification dataset tackling predictive maintenance. Our methodology advances the field of model interpretability for time series analysis and offers a framework for improving the transparency and trustworthiness of models deployed in critical decision-making domains.

Source code for *Sequence Dreaming* and results of the experiments are online available at:
https://github.com/visual-xai-for-time-series/sequence-dreaming

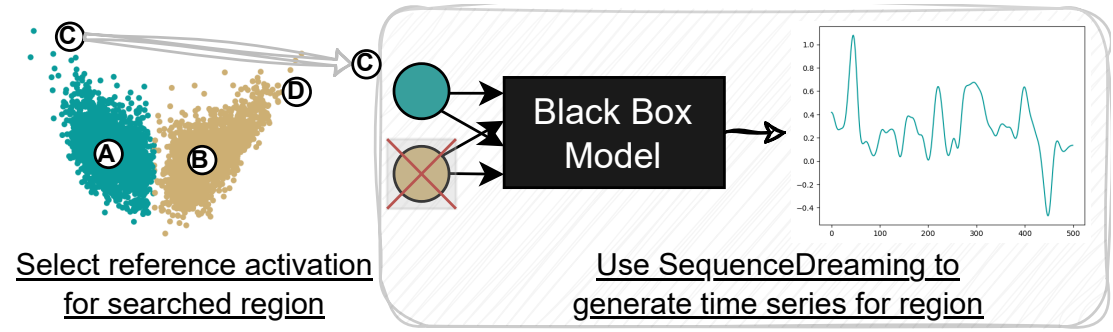
2. Related Work

In explainable artificial intelligence (XAI), elucidating the operational intricacies of machine learning algorithms, particularly for time series data, is imperative for advancing the field's theoretical and practical applications. Theissler et al. [4] delineate a comprehensive framework for examining time series XAI, emphasizing the necessity of addressing interpretability at multiple analytical levels. This multi-tiered perspective is instrumental in unraveling the complex dynamics and temporal dependencies inherent in time series data, thereby facilitating a deeper understanding of model behavior.

Activation Maximization, within this context, emerges as a crucial methodology for probing the internal representations developed by neural networks during the training process [1]. It accomplishes this by optimizing input signals to obtain maximal responses from specific neurons or layers, effectively revealing the features and patterns deemed most significant by the model for its decision-making processes. DeepDream enhances Activation Maximization by



(a) A black box model gets trained on binary data to classify between two classes. The logits before the softmax can then be projected again with PCA to generate a visual representation of the predictions. The classifier splits the predictions into two clusters with the logits. However, we are interested in the different parts of the clusters. What are the cluster centers (A) or (B)? How do the cluster edges look like (C) or (D)? How can we recreate the data of these interesting regions?



(b) Select a reference activation such as a class cluster center (A) or a maximization such as (C). Here, we want to create a maximization, so (C) is selected. We also want the selected activation of the class. Thus, the other class activation gets deactivated. Through gradient optimization, the input of the black box model gets slowly adapted to generate a time series for the selected region.

Figure 1: General approach split up into two pipelines. First, a projection of the activations of the data into 2D to find interesting regions to focus on. Second, the *Sequence Dreaming* approach generates time series in selected regions of interest to find salient features of the model.

incorporating regularization techniques into the optimization process, guiding the model to modify inputs (typically images) to emphasize the learned features while maintaining or even enhancing the visual coherence and interpretability of the outputs [2].

Thus, DeepDream not only highlights the features that activate certain neurons but does so in a manner that produces aesthetically intriguing and richly detailed images, thereby making the abstract concepts learned by the network more tangible and understandable to humans [5]. One of the earliest examples of reconstructing images based on a neuron activation is provided by Yosinski et al. [6] incorporating Gaussian blur, cropping by pixel contribution, and cropping by pixel norm without including a loss function. Nguyen et al. [7] demonstrate that DNNs can be induced to make high-confidence predictions for images that are either nonsensical

or unrecognizable to human observers, highlighting significant vulnerabilities in the models' interpretability.

Integrating Activation Maximization techniques within time series analysis significantly enhances our understanding of neural network predictions, thereby improving the transparency and accountability of these models for critical applications, for instance, in Schlegel et al. [8]. However, as highlighted in the literature, applying Activation Maximization to time series data presents unique challenges, not least due to the complexity of temporal data [9]. Various studies, including those by Yoshimura et al. [9] and [10], have explored alternative approaches, such as utilizing the spectral domain and Fourier transformations on short time series, often focusing beyond the last layer of the network. Meanwhile, Ellis et al. [11] have proposed a novel method that involves perturbing the spectral domain to achieve Activation Maximization, although these results are often overly focused on periodicity. This indicates a pressing need for a more systematic approach to effectively apply Activation Maximization in the context of time series data, aligning with the broader objectives of explainable and interpretable machine learning models as advocated by the XAI community [9].

3. Activation Maximization

Activation Maximization emerges as a promising technique designed to reveal the features most salient to individual neurons within a trained neural network, enabling interpretability [2]. Post-training, the focus shifts to decoding the learned representations by identifying the inputs that provoke maximal activation in specific neurons. This process, integral to Activation Maximization, aims to uncover the features or patterns to which a neuron is most responsive by iteratively refining the input to maximize a neuron's activation, providing deeper insights into the model's internal representations [1]. However, applying this technique to time series data introduces novel challenges necessitating regularization strategies.

In time series classification, we can define a *time series* by $ts = (t_1, t_2, \ldots, t_m) \in \mathbb{R}^{m \times d}$ as an ordered set of m real-valued observations (or time steps), with dimensionality d [4]. For *univariate* time series, we have $d = 1$ and thus our $ts \in \mathbb{R}^m$. Given a trained model M and a target class c, the activation maximization is described as follows: Consider a score function $S_c : \mathbb{R}^m \to \mathbb{R}$ and let $S_c(I)$ denote the score of the class c from M computed by the classification layer of the model for an input $I = (x_1, x_2, \ldots, x_m) \in \mathbb{R}^m$, e.g., $I = ts$ and thus $S_c(ts)$. Thus, our previous time series ts works as an input for the model M. Next, we want to find an L_2-regularised input, such that the score S_c is maximized, i.e.,

$$\max_{ts} S_c(ts) - \lambda \|ts\|_2^2, \tag{1}$$

where $\lambda > 0$ is a regularisation parameter. By employing the back-propagation technique, we can identify an input, referred to as ts, that is locally optimal in terms of the model's criteria. Rather than modifying the network's weights, we hold them constant at the values established during the training phase and instead focus our optimization on the input ts itself. In our case, we focus on $S_c(ts)$ and, in the first step, remove the L_2-regularised input, focusing directly on the maximization of the activation score of the class, i.e., $\lambda = 0$. For this approach, we use a gradient ascent to fine-tune the input to increase the activation.

4. Regularization Tricks

Regularization plays a crucial role in Activation Maximization, primarily to ensure the generation of interpretable, meaningful, and visually coherent inputs [6]. It prevents the optimization process from overfitting to noise, which would otherwise lead to unrecognizable patterns that maximize neuron activation but lack relevance. By introducing regularization, such as Gaussian blur [6], the process is guided towards producing aesthetically pleasing inputs closer to the distribution of natural inputs, enhancing the interpretability of the results. This ensures that the generated inputs reflect genuine features the network learns from real-world data, offering clearer insights into its internal representations and improving our understanding of its decision-making processes.

4.1. Without regularization on the loss

Exploring regularization techniques that do not alter the loss function presents an approach in optimization we first want to explore, particularly within Activation Maximization using gradient ascent. Thus, we collect and reformulate the approaches from the literature [6, 9, 11] to time series. We focus on the following approaches:

Clamping to borders, as described by Yosinski et al. [6], ensures that the optimized input does not venture beyond the predefined input space, maintaining realism and coherence. This technique effectively keeps the activation maximization process within the bounds of the training data distributions.

L_2 **decay**, another technique highlighted by Yosinski et al. [6], imposes a regularization term that penalizes high-frequency noise in the generated input, promoting smoother and more interpretable features. This regularization helps focus on the essential features that the neuron detects rather than artifacts.

Random scaling introduces variability in the time points of the input during the optimization process, encouraging the network to identify and amplify scale-invariant time points. This technique enriches the diversity of patterns that activate the neurons, showcasing the model's robustness to scale variations.

Moving average smoothing is applied to the input to mitigate rapid fluctuations, ensuring that the input generation progresses smoothly toward enhancing meaningful patterns. This approach helps stabilize the time points, making it less susceptible to local optima.

Exponential smoothing, when applied to the input during the optimization process, emphasizes the significance of smoother inputs by assigning them greater weights. This method adapts the input dynamically, ensuring that the activation maximization is finely tuned based on the most recent trends and patterns in the data, thereby fostering a more responsive and effective approach to highlighting the neuron's preferences.

Gaussian blur filter, as employed by Yosinski et al. [6], aids in reducing high-frequency noise across the optimization iterations, thereby focusing the model's attention on broader, more significant patterns. This technique contributes to the production of more visually appealing and interpretable inputs.

Random reinitiation of the input, if the optimization process shows no significant change, acts as a reset mechanism to escape from potential plateaus in the activation landscape. This

strategy prevents stagnation, ensuring continuous exploration for more effective stimuli that maximally activate the target neuron.

Intuitive Explanation – Most of the approaches mentioned above are straightforward in their methodology. Generally, they aim to regularize the creation of the time series without altering the loss function, focusing solely on increasing the activation of the selected neuron. This often necessitates significant smoothing since there are no constraints on the loss function and, consequently, the gradients.

4.2. With regularization on the loss

After we adopt a regularization approach without modifying the loss function, utilizing gradient ascent as demonstrated by Yosinski et al. [6], we transition the approach imposing regularization directly on the loss function, thereby altering our optimization strategy. We no longer perform a full activation maximization but aim to approximate a specific target activation in advance. Thus, we shift from gradient ascent to gradient descent, employing traditional optimizers and thus integrating a specific loss term with, for example, stochastic gradient descent with momentum to refine our optimization process further.

In extending Equation 1, we incorporate total variation regularization, as outlined by Simonyan et al. [1], to enhance the visual clarity and reduce noise in the generated input, which also enables smoothing for the input time series. To optimize this extended formulation, we employ the Adam optimizer [12], known for its efficiency in handling sparse gradients and adaptive learning rates, complemented by weight decay for improved regularization on the input, following the approach suggested by Mahendran and Vedaldi [5]. Additionally, we meticulously adjust the normalization parameters for the L_2-Regularization (weight decay on the input) and Total Variation (on the input) and fine-tune the weighting of the loss terms to identify the most effective settings for our optimization objectives, ensuring a balanced and nuanced approach to maximizing activation as closely to our target as possible.

Thus, we get the formula from Mahendran and Vedaldi [5]

$$\min_{ts} \frac{|S_c(ts) - S_c(T)|^2}{|S_c(T)|^2} + \lambda_\alpha \cdot \|ts\|_\alpha^\alpha + \lambda_\beta \cdot TV(ts, \beta), \tag{2}$$

with $S_c(ts)$ as the score of the class c for an input ts and $S_c(T)$ as the score of the class c for a target T. The target T and $S_c(T)$ as a reference for the activation we want to achieve, as mentioned before. The parameter α changes the input range to be encouraged to stay within an interval if set to large values > 2, commonly set to $\alpha = 6$ [5]. The parameter β controls the total variation factor and, thus, how similar the inputs in the neighborhood should be. The parameter is normally $\beta > 1$ so that the weighting of the error between the time points can be adjusted. In our case, we want a confident prediction of a sample toward the class c after the softmax, e.g., $M(T) = [0, 1]$ for two classes and $c = 2$ for $c \in \{1, 2\}$.

TV corresponds to the total variation adapted and approximated with

$$TV(ts, \beta) = \sum_{i=1}^{m} (t_{i+1} - t_i)^\beta \tag{3}$$

as seen in Mahendran and Vedaldi [5] adapted to time series with just one dimension. The parameter λ_α weight the L_2 regularization term, while λ_β weights the TV.

Intuitive Explanation – The loss terms can be intuitively explained as follows: The first part ensures that the input score gradually moves toward the desired target score, where the activation of the input should resemble the activation of the target. In our case, the target is a high activation above two or three times above the average of the activations of the train data. The second part keeps the input values within a set range, provided that the parameter α is chosen carefully. Here, the normalization focuses on making outlier values more averaging out with the other values. Lastly, the final part ensures that the neighborhood has similar values without outliers, as the TV smooths the data with a β value greater than or equal to 1. With a high β, errors between time points are heavier weighted, thus minimizing the equation forces to lower the difference between time points.

4.3. *Sequence Dreaming* combining both

Sequence Dreaming combines the most promising regularization techniques from previous work on images with specialized ones on time series to generate an approach for time series deep learning models. By extending Equation 2 with an additional smoothness factor similar to TV but scaled for time series lengths, *Sequence Dreaming* achieves a more refined regularization process. This method employs a combination of the new loss extension, Gaussian blur, clamping, and random noise to ensure effective regularization, particularly when the loss is not significantly changing. Furthermore, *Sequence Dreaming* transitions from using Adam to a gradient descent method without momentum or other improvements, necessitating a few more optimization steps to achieve the desired results.

First, we extend Equation 2 by adding an additional regularization term

$$\min_I \frac{|S_c(ts) - S_c(T)|^2}{|S_c(T)|^2} + \lambda_\alpha \cdot \|ts\|_\alpha^\alpha + \lambda_\beta \cdot TV(ts, \beta) + \lambda_{sm} \cdot SM(ts), \tag{4}$$

where (SM)oothness $SM(ts)$ is defined as

$$SM(ts) = \frac{1}{m-1} \cdot \sum_{i=1}^{m-1} |t_{i+1} - t_i|. \tag{5}$$

SM takes the input, calculates the discrete difference between time points, uses the absolute value of the difference, and sums these up. The normalization of the length enables an equal weighting between every time point.

Why SM? – The inclusion of two smoothing factors, despite TV's inherent smoothing capabilities, arises from their distinct focuses and benefits. TV primarily aims to minimize errors between individual time points, emphasizing the coherence of each point in the time series. In contrast, the second smoothing factor, SM, targets the entire time series, enhancing overall normalization and providing a broader perspective on data regularization. While TV hones in on reducing local discrepancies, SM ensures a holistic approach, balancing the series as a whole and easing the optimization path in the loss landscape. This dual-faceted approach

enables a more comprehensive smoothing process, addressing micro-level and macro-level anomalies within the time series.

Another important discussion point is the starting point of the optimization. While it is common to begin with completely black or white images, some approaches use the mean of a chosen dataset or random values [5]. We provide a more directed approach to the starting input for optimization as time series starting at zero are often standardized and already incorporate biases. Specifically, we utilize the activations seen in Figure 1b to guide the search for the initial input. Different regions of the activations in Figure 1b can be used as starting points to steer optimization based on desired outcomes. For instance, in most cases, such as (A) in Figure 1b, starting with the closest sample to the mean activation of the dataset proves beneficial. This method enables the process to commence with a solid activation baseline, which can then be further optimized. Thus, *Sequence Dreaming* adapts an existing sample to align with the model's preferences, thereby maximizing the activation of the selected neuron.

To search for the most fitting hyperparameters, we perform a grid search due to the extensive range of possible hyperparameters that require fine-tuning, such as λ_{sm}. In some instances, we have an idea of the optimal direction for hyperparameters, such as α. However, determining these parameters in many cases is not straightforward and is highly sensitive to the dataset, particularly the weighting parameters for the loss term λ_{sm}. Through empirical testing, we discovered that while some parameters are sensitive, we can generally provide certain guidance towards the hyperparameters discussed in the evaluation section.

Intuitive Explanation – *Sequence Dreaming* enhances the previous loss function with an additional smoothing factor to ensure that every time point throughout the time series has similar importance. By focusing further on smoothing, we aim to optimize the resulting time series into a smooth version without large outliers, making it more realistic. Ultimately, this optimization results in a smooth time series with only patterns and outliers at important time points. Our implementation techniques further steer the direction to closely match the target activation by adding noise if the activation becomes too large and risks overshooting.

5. Evaluation of Results and Discussion

Evaluation is crucial because different approaches can generate activation maximization samples that are not always plausible for our dataset. Outside the projected data, these samples may still maximize neuron activation but in a different region, as illustrated in Figure 1b. Ellis et al. [11] assess their activation maximization approach by comparing the activation distribution against the data, performing a visual evaluation, analyzing the frequency domain, and examining frequency importance. Similarly, we will use a visual evaluation to present the generated time series. Subsequently, we will employ measures for outlier detection to assess the plausibility of the generated activation maximization input toward the model and the data. Finally, we will integrate out-of-distribution analysis and visual evaluation with distribution plots to compare *Sequence Dreaming* with other approaches.

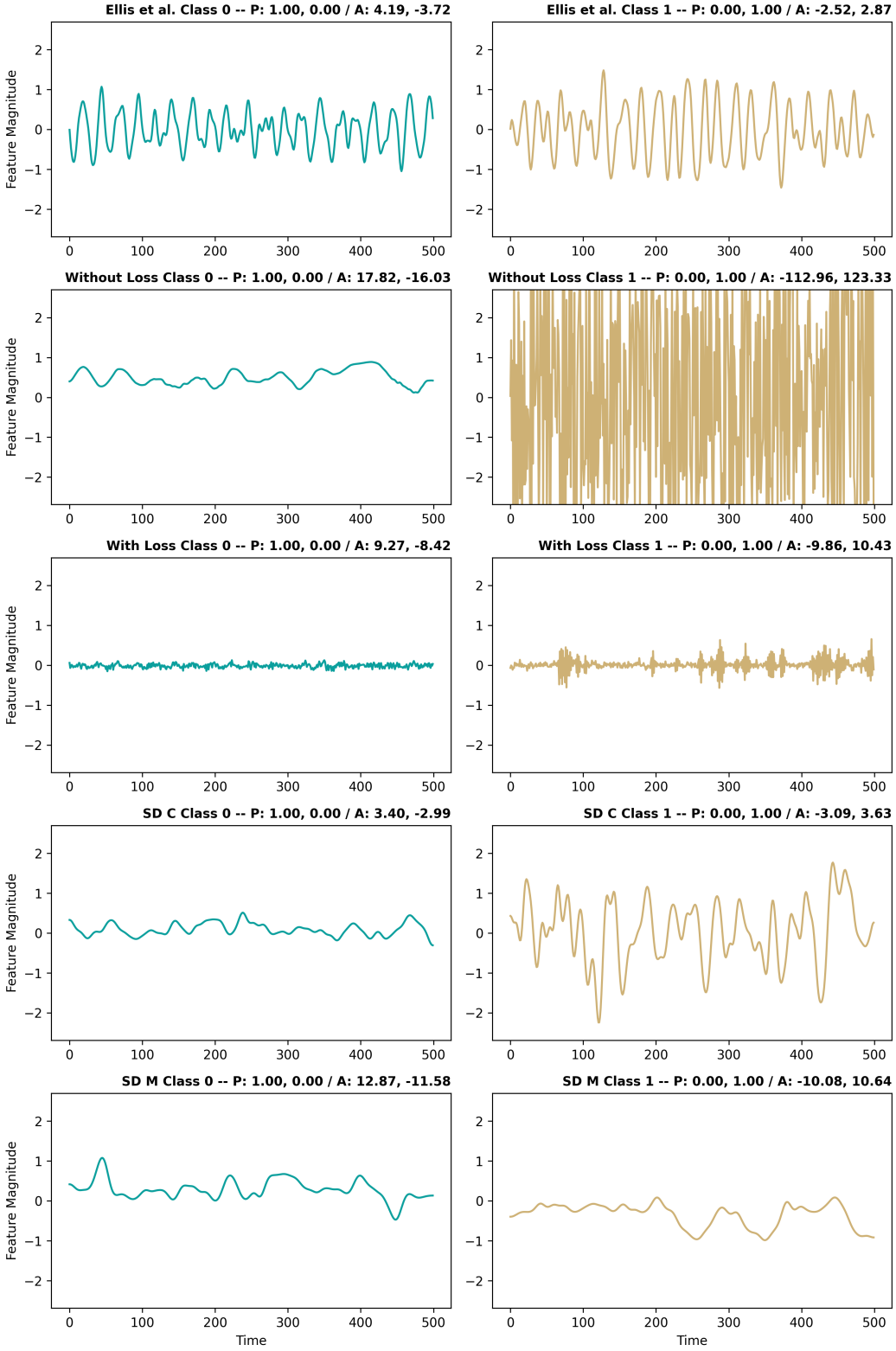


Figure 2: Comparison of the generated activation maximization time series with their corresponding predictions and activations with the two classes (first class, second class). Ellis et al. [11] demonstrate plausible time series. However, our *Sequence Dreaming* (SD) comes up with totally different ones while smoothing the without-loss and with-loss results.

The evaluation is conducted on the *FordA* dataset, a well-established univariate time series classification benchmark for anomaly detection in car engines. This dataset consists of time series samples with 500 time points each, comprising 3,601 samples for training and 1,320 for testing. However, all datasets available from the UCR benchmark repository [13] are compatible with the *Sequence Dreaming* source code available online. The model employed is a conventional *ResNet* architecture featuring three ResNet blocks, each containing three Conv1D layers, followed by a final linear classifier layer. The model is trained using the Adam optimizer [12] over 500 epochs. The model achieves an accuracy of *0.99* on the training set and *0.95* on the test set, approaching near *state-of-the-art* performance. For the hyperparameter search in *Sequence Dreaming*, a grid search is utilized with minimum and maximum values as outlined in Table 1. While the current parameter ranges are already relatively narrow, they can be further reduced to accelerate the search process. The hyperparameter ranges were determined based on literature references and empirical testing with respect to the loss function.

Table 1
Hyperparameter Search Space for *Sequence Dreaming* on the FordA Dataset using the ResNet Model. A grid search is applied, and the search space can be further reduced to streamline the search process and enhance the speed of the overall identification.

	steps	lr	α	β	σ	λ_α	λ_β	λ_{sm}
Min	5	1e-2	4	1	3	1e-5	1e-5	1e-1
Max	100	1e1	6	2	6	1e-1	1e-1	5e-1

5.1. Visual Evaluation

Figure 2 visualizes the different approaches as line plots for direct comparison. At the top, the method name is displayed, followed by the prediction using the generated time series, and finally, the activation of the neurons in the selected layer. The left side always represents the first class, while the right side represents the second class. The figure includes methods from Ellis et al. [11], the mentioned approach without loss change, the approach with loss change, *Sequence Dreaming* on the class center, and the maximization using *Sequence Dreaming*.

Comparing the different approaches at a glance provides initial insights seen in Figure 2. Ellis et al. [11] produce quite an appealing time series, but the use of FFT introduces some periodicity in the generated time series. The generated time series without a change in the loss results in high values for the second class (1) and quite low values for the first class (0), indicating that this approach is not very effective. The time series generated with loss adjustment are not as smooth and tend to be somewhat noisy, particularly towards the second class. In contrast, *Sequence Dreaming* generates convincing time series for the class center and maximization, with distinct differences between the classes. Notably, the first class looks quite similar when *Sequence Dreaming* is targeted on the class center and class maximization, while the second class results in noticeably different generated time series.

5.2. Out-of-distribution evaluation

After visually comparing the activation maximization results, we aim to assess their performance using various quantitative methods. The distributions of the class' activations for the training data are particularly interesting in this context to compare them to the generated time series. Different methods exist for evaluating data based on distributions, though we focus on outlier analysis to assess the quality of the generated time series towards the training data.

In our case, we employ outlier analysis using the Mahalanobis distance [14] on time series data and activations. The Mahalanobis distance measures the distance between a point and a distribution, considering the correlations between variables. It is calculated by determining how many standard deviations away a point is from the mean of the distribution, considering the covariance matrix to account for the data's spread and orientation. This makes it a useful approach for identifying outliers, as points that are far from the mean in terms of the Mahalanobis distance are likely to be anomalies. We believe that applying this measure directly to the time series data is not ideal due to its diversity; only existing time series or those generated in the frequency domain would fit within the distribution. However, we do think this approach can effectively demonstrate how well the generated time series fit into the activations, revealing whether they are more on the border with high activation or more centered.

Table 2
Mahalanobis distance for the activations of the generated data and the training data activations. Min and Max correspond to the minimum and maximum distances of samples in the data based on the Mahalanobis distance for the data. Ellis et al. [11] perform quite well. However, *Sequence Dreaming* (SD) for the center works quite well, and also for the maximization, which needs to be more different than the previous ones to show that the neuron gets maximally activated.

Class	Min	Max	Ellis et al. [11]	Without Loss	With Loss	*SD* Center	*SD* Max
0	0.11	5.18	1.36	13.86	3.92	1.63	8.41
1	0.10	4.61	1.38	137.74	3.54	1.32	3.53

Evaluation against activations from the training dataset – Table 2 shows the Mahalanobis distance toward the activations of the training data compared to the generated time series by the different approaches. Min and Max correspond as a reference to the minimum and maximum distances of the samples in the training data itself. Ellis et al. [11] performs quite well, positioning very nicely between the min and max. However, due to the small values, these are rather center class prototypes and not activation maximization. The approach without loss changes performs rather poorly. In contrast, the approach with loss changes falls nicely between the min and max, even leaning more towards the borders. This indicates that the generated time series yields favorable results according to this measure. However, the visual evaluation shows that the results are not impressive. *Sequence Dreaming* center presents values similar to Ellis et al. [11], aligning well with the distribution but not reaching the borders. This outcome was expected since we generated these time series to be more centered, indicating that the approach works effectively based on this measure. Additionally, *Sequence Dreaming* max works quite well by providing larger numbers for the distance, suggesting that the generated time series are

anomalies and somewhat borderline activations as intended. Overall, our proposed *Sequence Dreaming* works very well for the activations seen in Table 2.

Table 3
Mahalanobis distance for the time series training data and the generated time series. Min and Max correspond to the minimum and maximum distances of samples in the data based on the Mahalanobis distance for the data. Even Ellis et al. [11] with using the frequency domain for the generation can only generate class one time series in the time series training data distribution, which is very surprising. *Sequence Dreaming* (SD) cannot produce a time series in the distribution, which suggests that the model learns more abstract patterns for the classification.

Class	Min	Max	Ellis et al. [11]	Without Loss	With Loss	*SD* Center	*SD* Max
0	1.23	55.53	23842.49	652852860.28	18435054.75	123537050.76	376120973.62
1	1.16	33.88	54.34	74930207.64	7966354.20	56484405.81	461642519.01

Evaluation against the training dataset – Table 3 presents the Mahalanobis distance from the original time series training data to the generated time series. Min and Max again correspond to the time series distances in the data to provide a baseline. Ellis et al. [11] can create a time series that can still be considered a close outlier to the distribution. However, by using the frequency domain to generate the time series, such a result is expected for most datasets since the correlation between time points is much more important for this approach. Despite this, a deep learning model can learn entirely different structures to recognize the class, making such an approach produce a rather misleading time series. The other approaches, as seen inTable 3, perform significantly worse in the Mahalanobis distance for the time series. This result makes sense, as the generated time series by the other approaches are quite different from those generated by Ellis et al. [11].

Our observation was rather correct, as this is not the best approach to compare to the original time series. However, we note that the outlier detection works for the activations, which can guide the generation of activation-maximized time series during creation and can be used to extend the *Sequence Dreaming* approach in future works. Other possible future extensions for evaluation include using autoencoders for outlier detection or out-of-distribution detection. Additionally, other outlier detection mechanisms based on uncertainty quantification toward the activation maximizations can be explored. These present possible future research opportunities.

5.3. Visual out-of-distribution evaluation

After inspecting the raw numbers and uncovering some interesting findings, we combine the visual observations' results and the numbers from the outlier detection into distribution plots. First, we use violin plots to show activations and the results of the generated time series, similar to Ellis et al. [11]. Next, we use projections (PCA) of the activations, as seen in Figure 1a.

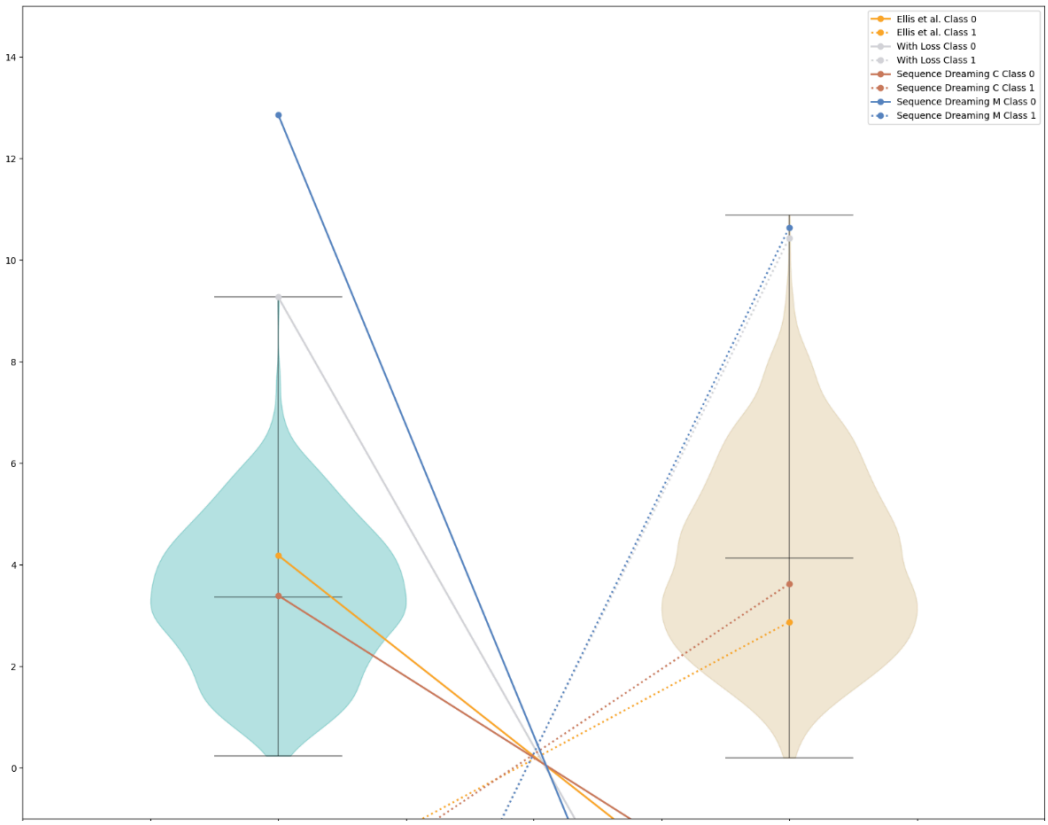


Figure 3: Violin distribution plot of the activations for the training data of the model for the class zero on the left and on the right for class one (first class, second class). Ellis et al. [11] and *Sequence Dreaming* center both are near the mean of the distribution on both classes, while *Sequence Dreaming* max is above the general activations. Interestingly, with loss shows good results on top of the activations.

The results are unsurprising of Figure 3 and align with the previous Mahalanobis distance numbers. However, Figure 3 illustrates how closely the activations of the generated time series from Ellis et al. [11] and *Sequence Dreaming* center are aligned. Interestingly, the "with loss" approach shows lower activations than the *Sequence Dreaming* max approach. We had to exclude the "without loss" data as the activations were too high and would have rendered the distribution plot unreadable without axis scaling. Ellis et al. [11] achieves an activation mean similar to that of the training data, suggesting that their method effectively maximizes activation

by highlighting patterns of interest to the model. However, *Sequence Dreaming* also captures some of these patterns without relying on the frequency domain, producing more abstract, non-periodic results, which indicate some other learned representations by the model unseen to Ellis et al. [11]. Such a result can indicate that the model learns some values by heart to classify certain samples, which decreases the generalizability of the model.

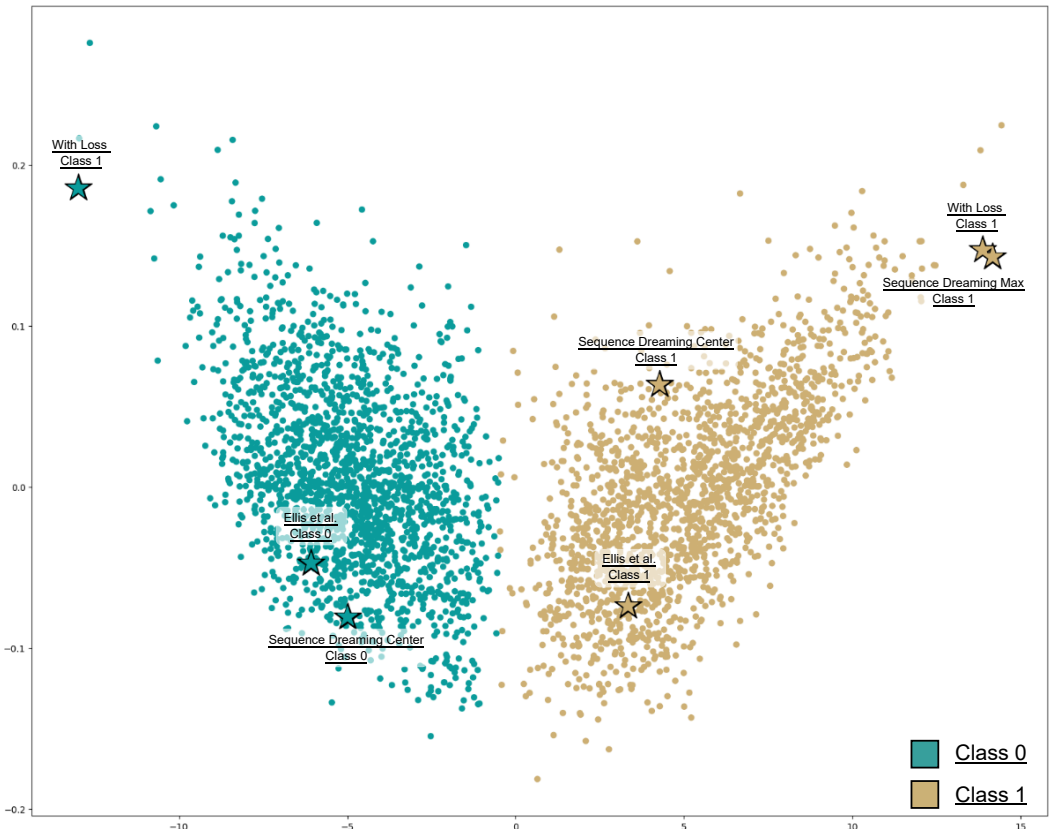


Figure 4: First two principal components of a PCA on the activations of the model for the training data for the two classes (first class, second class). Afterward, the generated time series are also included. However, we had to exclude Without Loss as these would be outside of the current viewport by a large margin. These would destroy the visualization in general. Our *Sequence Dreaming* approach captures in center and max quite well what we want to have with either class centers or borders. Ellis et al. [11] produce good class centers.

As we have seen in previous sections, the activations work quite well in evaluating if the generated time series are positioned as desired. We can use the approach shown in Figure 1a. By collecting all the activations of a dataset, we can generate a PCA projection using only the first two principal components, as illustrated on the right side of Figure 1a. We can then add the generated time series to obtain a figure such as Figure 4. Similar to the Mahalanobis distances in Table 2, we see that Ellis et al. [11] form quite nicely centered activation maximizations, as seen in (A) or (B) of Figure 1b. *Sequence Dreaming* can achieve similar results with *Sequence*

Dreaming center. Additionally, *Sequence Dreaming* can achieve maximum and border activations, such as (C) and (D) in Figure 1b. Thus, with slight changes to the hyperparameters, *Sequence Dreaming* can generate time series for different use cases, revealing some aspects of the inner processes of a deep learning model at the borders, as seen in Figure 4.

6. Conclusion and Future Work

In conclusion, we conducted a study on effective regularization for time series classification data without altering the loss function, using only gradient ascent and a modified loss function to achieve good results. Based on our observations, we introduced *Sequence Dreaming*, an adaptation of previous methods, incorporating enhanced regularization with increased smoothness. The results demonstrate that adding regularization terms to data transformation and the loss function can be effective. We learned that activation maximization with regularization terms is highly parameter-sensitive but can produce convincing results comparable to those of frequency domain approaches, which inherently have advantages. Our PCA projections of the activations particularly highlight how our approach can generate precise time series corresponding to specific activations with smooth properties.

For future work, several promising directions are poised to refine the *Sequence Dreaming* process for time series data. Among these, exploring advanced regularization techniques, such as introducing perturbations directed towards less contributing time points informed by attributions, offers a nuanced method for enhancing model interpretability while minimizing information loss. Additionally, the adoption of genetic or evolutionary algorithms, inspired by the works of Nguyen et al. [7] and Xiao and Kreiman [15], presents an intriguing avenue for optimizing activation maximization through a process that mimics natural selection, potentially uncovering novel and highly effective stimuli. Moreover, incorporating wavelet transforms, akin to Fourier transformations, could provide a more comprehensive analysis of time series by capturing both the frequency and location in time of significant features, thereby offering a richer representation of the data for time series activation maximization generation. Also, incorporating attributions more heavily in the process could lead to more plausible activation maximization time series as these can regularize the generation process. However, selecting a working attribution technique is not trivial and needs another careful consideration [16].

Acknowledgments

This work has been partially supported by the Federal Ministry of Education and Research (BMBF) in VIKING (13N16242).

References

[1] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: visualising image classification models and saliency maps, in: International Conference on Learning Representations (ICLR), 2014.

[2] A. Mordvintsev, C. Olah, M. Tyka, Deepdream-a code example for visualizing neural networks, Google Research (2015).
[3] C. Olah, A. Mordvintsev, L. Schubert, Feature Visualization, Distill (2017). Https://distill.pub/2017/feature-visualization.
[4] A. Theissler, F. Spinnato, U. Schlegel, R. Guidotti, Explainable AI for Time Series Classification: A review, taxonomy and research directions, IEEE Access (2022).
[5] A. Mahendran, A. Vedaldi, Understanding deep image representations by inverting them, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015.
[6] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, Understanding neural networks through deep visualization, arXiv preprint arXiv:1506.06579 (2015).
[7] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015.
[8] U. Schlegel, D. Oelke, D. A. Keim, M. El-Assady, Visual Explanations with Attributions and Counterfactuals on Time Series Classification, arXiv preprint arXiv:2307.08494 (2023). arXiv:2307.08494.
[9] N. Yoshimura, T. Maekawa, T. Hara, Preliminary investigation of visualizing human activity recognition neural network, in: Conference on Mobile Computing and Ubiquitous Network (ICMU), 2019.
[10] N. Yoshimura, T. Maekawa, T. Hara, Toward understanding acceleration-based activity recognition neural networks with activation maximization, in: International Joint Conference on Neural Networks (IJCNN), 2021.
[11] C. A. Ellis, M. S. E. Sendi, R. Miller, V. Calhoun, A novel activation maximization-based approach for insight into electrophysiology classifiers, in: IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2021.
[12] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
[13] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, E. Keogh, The UCR time series archive, IEEE/CAA Journal of Automatica Sinica 6 (2019) 1293–1305.
[14] P. C. Mahalanobis, On the generalized distance in statistics, Sankhyā: The Indian Journal of Statistics, (1936).
[15] W. Xiao, G. Kreiman, Gradient-free activation maximization for identifying effective stimuli, arXiv preprint arXiv:1905.00378 (2019).
[16] U. Schlegel, D. A. Keim, A Deep Dive into Perturbations as Evaluation Technique for Time Series XAI, in: World Conference on Explainable Artificial Intelligence (xAI), 2023.