MASCOTS: Model-Agnostic Symbolic COunterfactual explanations for Time Series

Dawid Płudowski[1] [0009−0001−2386−7497] ✉,
Francesco Spinnato[3,4] [0000−0002−3203−6716],
Piotr Wilczyński[1,7] [0009−0009−8418−3062],
Krzysztof Kotowski[5] [0000−0003−2596−6517],
Evridiki Vasileia Ntagiou[6] [0000−0003−3403−2863],
Riccardo Guidotti[3,4] [0000−0002−2827−7613], and
Przemysław Biecek[1,2] [0000−0001−8423−1823]

[1] Warsaw University of Technology, Poland
[2] University of Warsaw, Poland
[3] University of Pisa, Italy
[4] ISTI-CNR, Pisa, Italy
[5] KP Labs, Poland
[6] European Space Operations Centre, Germany
[7] ETH Zürich, Switzerland

Abstract. Counterfactual explanations provide an intuitive way to understand model decisions by identifying minimal changes required to alter an outcome. However, applying counterfactual methods to time series models remains challenging due to temporal dependencies, high dimensionality, and the lack of an intuitive human-interpretable representation. We introduce MASCOTS, a method that leverages the Bag-of-Receptive-Fields representation alongside symbolic transformations inspired by Symbolic Aggregate Approximation. By operating in a symbolic feature space, it enhances interpretability while preserving fidelity to the original data and model. Unlike existing approaches that either depend on model structure or autoencoder-based sampling, MASCOTS directly generates meaningful and diverse counterfactual observations in a model-agnostic manner, operating on both univariate and multivariate data. We evaluate MASCOTS on univariate and multivariate benchmark datasets, demonstrating comparable validity, proximity, and plausibility to state-of-the-art methods, while significantly improving interpretability and sparsity. Its symbolic nature allows for explanations that can be expressed visually, in natural language, or through semantic representations, making counterfactual reasoning more accessible and actionable.

Keywords: Explainable AI (XAI) · Counterfactual Explanations · Time Series · Model-Agnostic Explanations.

1 Introduction

Time series classification (TSC) plays a crucial role in various fields, including healthcare, climate science, and engineering. Its wide-ranging applications have

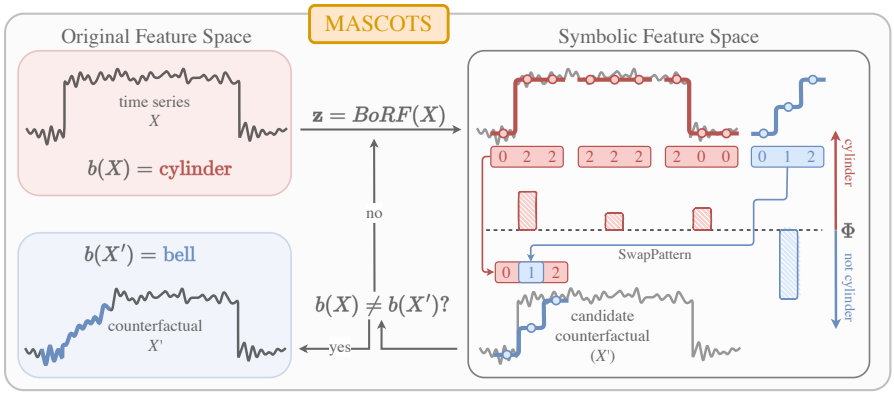

Fig. 1. A simplified graphical abstract of MASCOTS. Given a time series X classified by a black-box model b as *cylinder*, X is transformed into a symbolic representation, $\mathbf{z}$, using the Bag-of-Receptive-Fields (BoRF) [36]. In this semantic space, X is represented as symbolic subsequences. To identify a candidate counterfactual, the most important positive and negative patterns (*0,2,2* and *0,1,2* in the illustration) are selected. The negative pattern is then *swapped* within the time series to generate a candidate counterfactual. If the black-box model's predicted class remains unchanged, the process repeats. Otherwise, the final counterfactual is returned.

driven the development of increasingly powerful predictors capable of achieving remarkable classification accuracy. Both univariate and multivariate time series classification have gained significant research interest, as demonstrated by recent "bake-offs" [35,32,38], which periodically benchmark the top-performing classifiers. The findings from these evaluations are consistent: the most effective classifiers are powerful hybrid ensembles, such as MultiRocket-Hydra [10], Hive-Cote 2 [31], and InceptionTime [17]. These models leverage both prediction and feature spaces from multiple underlying algorithms, resulting in state-of-the-art accuracy. However, their complexity renders them black-box models, meaning they lack interpretability from a human perspective.

This work aims to enhance the interpretability of black-box models in time series classification through Explainable Artificial Intelligence (XAI) techniques [5]. While XAI offers diverse tools for explaining complex models, most approaches have been developed for tabular or image data, with time series explainability only recently gaining traction [39]. In this domain, explanations commonly take the form of saliency maps [30], highlighting the most relevant part of observations contributing to the classification outcome, or subsequence-based explanations such as shapelets [43] focusing on significant sub-patterns within a time series. Instead, this work centers on instance-based explanations, where entire time series serve as the primary explanatory objects. Specifically, we focus on counterfactual explanations, i.e., minimal semantically valid modifications to an input time series that alter the classification outcome of a black-box

model [40,18]. Counterfactual explanations are particularly powerful because, unlike feature importance methods [30,15], they indicate what must change in a given instance to achieve a different classification result [14]. They are useful as they facilitate reasoning about the cause-effect relationships between observed features and classification outputs [6].

In this work, we introduce MASCOTS, a novel model-agnostic method for generating counterfactual explanations in time series. MASCOTS leverages symbolic representations to significantly enhance interpretability while maintaining fidelity. As depicted in Figure 1, MASCOTS initially converts a given time series into a Bag-of-Receptive-Fields (BoRF) representation [36], capturing essential symbolic patterns. It then identifies those patterns most strongly influencing classification outcomes, both positively and negatively. By iteratively substituting negative patterns, MASCOTS modifies the original time series until it obtains a valid counterfactual, ensuring an interpretable and robust explanatory process.

Our contributions are as follows. *(i)* In contrast to existing methods that depend on model-specific architectures or autoencoder-based sampling [42,37], MASCOTS is fully interpretable and model-agnostic. This allows it to be applied broadly across time series classifiers without making any assumptions about the internal mechanisms of the black-box models. *(ii)* To the best of our knowledge, MASCOTS is the first approach that employs a symbolic subsequence-based semantic space for explainability in the time series domain, providing a semantic-rich counterfactual generation process that does not depend on nearest unlike neighbors (NUN) [7]. *(iii)* MASCOTS facilitates both visual and natural language counterfactual explanations, improving the interpretability of the counterfactual generation process. *(iv)* Furthermore, MASCOTS enables explanations for state-of-the-art ensemble models, a task where most existing methods fail due to their reliance on internal model structures. Through extensive evaluation on benchmark datasets, we demonstrate that MASCOTS achieves comparable performance to state-of-the-art techniques regarding validity, proximity, and plausibility while significantly improving the sparsity of counterfactual explanations. By bridging symbolic representations with counterfactual reasoning, MASCOTS represents a significant step forward in explainability for time series models.

The structure of this work is as follows: Section 2 examines related research on counterfactuals, while Section 3 outlines the background of our proposed methodology. Section 4 elaborates on the approach, followed by experimental results and analysis in Section 5. Lastly, Section 6 presents the conclusions.

2 Related Works

The simplest counterfactual models for time series rely on classical distance-based algorithms, such as K-Nearest Neighbors (KNN) or Nearest Unlike Neighbor (NUN) [7], which identify the closest existing instance of a different class. In this category, we find approaches like Native Guides [9] and TimeX [12], where time series are perturbed to generate counterfactuals while adhering to desirable properties such as proximity, sparsity, plausibility, and diversity. However, these

approaches are limited to univariate data. CoMTE [1] extends this framework to multivariate data by modifying time series from the training set, and computing the minimal number of substitutions necessary to change the original classification. Since CoMTE primarily operates by identifying and substituting similar patterns from different classes, it may struggle to produce meaningful counterfactuals when the dataset lacks sufficiently close examples. AB-CF [25] and DiscoX [3] take a different approach by focusing on local patterns. They extract fixed-length subsequences using a sliding window or identify discords via the matrix profile [44], replacing them with the nearest counterparts from the desired class. Given their emphasis on locality, these methods may fail to capture global temporal dependencies, potentially leading to counterfactuals that are locally valid but globally unrealistic. Finally, in [20], KNN is employed to locally and globally *tweak* time series, altering the outcome of specific black-box models. However, this approach is also restricted to univariate data. Contrary to classical distance-based approaches, our proposal, MASCOTS, does not rely on NUNs or Euclidean distance, as it performs perturbations in a semantic space produced by an interpretable transformation, i.e., the Bag-of-Receptive-Fields [36].

More complex approaches leverage evolutionary algorithms [16] and generative models, such as autoencoders. Autoencoders come in various forms, including recurrent neural networks, such as LSTMs, which have been used in [23] to extend the concept of Contrastive Explanation Methods to time series. Convolutional neural networks (CNNs) have also been utilized, as seen in LASTS [37], as well as methods that test both, such as LatentCF++ [42]. The central idea of these approaches is to perturb time series within a simplified latent space, ensuring that counterfactuals remain closer to the distribution of the training set. In this sense, autoencoders can play an indirect role as a loss component that assesses counterfactual *plausibility*. Sub-SpaCE [34] exemplifies this approach by evaluating the plausibility of counterfactuals generated through a genetic algorithm with tailored mutation and initialization strategies. This method encourages modifications in a minimal number of subsequences, producing highly sparse explanations. Plausibility is assessed similarly in TeRCE [2], which leverages the shapelet transform to identify the most relevant shapelets for a given class, pinpoint their locations in the input instance, and replace them with values derived from the NUN. While these methods are model-agnostic, their main limitation is the requirement to train a separate autoencoder for each dataset [37], which makes their usage across a wide range of tasks impractical. In contrast, MASCOTS, does not require any generative model to produce a counterfactual.

Counterfactual explanations can also be model-specific, i.e., designed for a particular black-box model. With the exception of [20], which targets the Random Shapelet Forest [19], most model-specific approaches focus on explaining neural network-based methods. One such example is Glacier [41], an extension of LatentCF++ designed to generate counterfactuals for any deep-learning-based model. While this approach is both promising and extensively tested, it is limited to univariate data. Other notable methods include CELS [26] for univariate data and M-CELS [27] for multivariate data, both of which employ a gradient-based

strategy. These methods utilize three interdependent modules to produce sparse counterfactuals, leveraging a learned saliency map to guide perturbations. The primary limitation of gradient-based model-specific approaches is that, while certain deep learning architectures are highly effective for TSC, the current state-of-the-art consists primarily of hybrid ensemble methods [32]. These ensembles often integrate multiple classifiers, making it difficult, or even impossible, to compute gradients, thus restricting the applicability of such techniques. MASCOTS does not share these limitations as it is model-agnostic.

Finally, to the best of our knowledge, most of the aforementioned methods rely exclusively on visualizations to convey counterfactual modifications. In contrast, we enable the generation of counterfactual explanations in a structured, human-understandable manner using natural language descriptions. Specifically, given a time series X and its counterfactual X', the transformation can be articulated through a structured statement such as: *"To change the prediction of a black-box model from class c_i to c_j, the time series X needs to contain pattern a instead of b."* One approach that might seem similar is PUPAE [11], but the key difference is that it relies on a predefined set of templates that require domain expertise to construct. In contrast, MASCOTS derives its explanations directly from the time series patterns, eliminating the need for manually designed templates. This makes MASCOTS not only more flexible but also broadly applicable across different domains without requiring specialized knowledge.

3 Background

This section provides all the necessary concepts to understand our proposal.

Definition 1 (Time Series Data). *A time series dataset, $\mathcal{X} = \{X_1, \ldots, X_n\} \in \mathbb{R}^{n \times d \times m}$, is a collection of n time series. A time series, X, is a collection of d signals (or channels), $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\} \in \mathbb{R}^{d \times m}$. A signal, $\mathbf{x}$, is a sequence of m real-valued observations sampled regularly, $\mathbf{x} = [x_1, \ldots, x_m] \in \mathbb{R}^m$.*

When $d = 1$, the time series is *univariate*, for $d > 1$ it is *multivariate*. Time series datasets can be used in a variety of tasks. This work focuses on supervised learning, particularly Time Series Classification (TSC).

Definition 2 (Time Series Classification). *Let $\mathcal{X}$ be a time series dataset and $\mathbf{y} \in \{1, \ldots, c\}^n$ its corresponding labels vector, where c is the number of classes. The goal of Time Series Classification is to train a model f that maps each time series $X_i \in \mathcal{X}$ to a predicted label $\hat{y}_i$, such that $f(X_i) = \hat{y}_i$ for all $i \in \{1, \ldots, n\}$. This yields the predicted label vector $\hat{\mathbf{y}} = [\hat{y}_1, \ldots, \hat{y}_n] \in \{1, \ldots, c\}^n$.*

The goal of time series classification is to ensure that the trained model f predicts a label $\hat{\mathbf{y}}$ that closely matches the true labels $\mathbf{y}$, typically by minimizing a classification loss function during training. Many models produce probability distributions over classes, i.e., $\hat{Y} \in [0, 1]^{n \times c}$, with $\hat{\mathbf{y}}$ determined by the highest probability. While maximizing accuracy is important, providing explanations

for the predictions of a given model is becoming more and more relevant. This work focuses on a specific type of explanation, i.e., *counterfactual explanations*. Counterfactual time series show the minimal changes in the input data that lead to a different decision outcome [39]. Formally:

Definition 3 (Counterfactual). *Given a classifier f that outputs the decision $\hat{y} = f(X)$ for an instance X, a* counterfactual *consists of an instance X' such that the decision for f on X' is different from $\hat{y}$, i.e., $f(X') \neq \hat{y}$, and such that X' is* similar *to X, and that X' is* plausible.

Similarity (or proximity) usually refers to a distance metric, while plausibility depends on the counterfactual domain and is assessed by verifying that the instance is not merely an adversarial example and remains semantically coherent with the dataset. Other relevant metrics include *sparsity*, i.e., the number of features altered to generate a counterfactual, and *validity*, i.e., the ability of a counterfactual method to produce a valid counterfactual.

Counterfactuals can be obtained in several ways; here, we propose to adopt *surrogate models*. Explanations based on surrogate methods can clarify the behavior of black-box models, b, by employing a secondary, more interpretable model, g, to approximate the behavior of the primary black-box model, i.e., $b(X) \simeq g(X)$ [5]. By doing so, the surrogate model seeks to provide insights into the complex model's decisions by mimicking its outputs while remaining inherently more interpretable. A great advantage of surrogates is that they are model-agnostic, i.e., they can explain any black-box without any assumption about its inner components. Surrogates can be trained on the original raw time series data or after processing it into a more interpretable tabular representation.

There exist several symbolic representation-based methods for time series classification, such as MrSEQL [24], MrSQM [33], and SCALE-BOSS-MR [13], to name a few. However, in this work, we adopt the Bag-of-Receptive-Fields (BoRF) [36] as our interpretable tabular representation, as it was shown to achieve better overall accuracy [36], with a very fast prediction time. BoRF extends the classical Bag-of-Patterns [4] and, akin to the Bag-of-Words approach in text analysis, converts a time series into a vector of pattern counts. This transformation is achieved by sliding a potentially strided and dilated window along the time series to extract all possible receptive fields of a specified length, w. In this context, a receptive field is just a time series subsequence that can have gaps inside, i.e., can skip observations. These subsequences are then standardized and discretized into *words* of length $l \leq w$ using the Symbolic Aggregate Approximation (SAX) [28]. SAX uses the Piecewise Aggregate Approximation (PAA) [21] to segment each subsequence into equal-sized segments and then compute the mean value for each segment. Finally, these values are quantized using a set of breakpoints, obtained through the quantiles of the standard Gaussian distribution, which bin values in equiprobable symbols, α. Thus, from each time series, X, a set of patterns is extracted, where a single pattern is denoted as $\mathbf{p} = [\alpha_1, \ldots, \alpha_l] \in \mathbb{A}^l$, with $\mathbb{A}$ being a set of finite symbols. Each symbolic pattern is bidirectionally hashed into an integer k, allowing for both encoding and decoding, and is then stored in the Bag-of-Receptive-Fields. Formally:

Definition 4 (Bag-of-Receptive-Fields). *Given a time series dataset* $\mathcal{X} \in \mathbb{R}^{n \times d \times m}$ *and a set of* h *patterns, a Bag-of-Receptive-Fields is a tensor* $\mathcal{Z} \in \mathbb{N}^{n \times d \times h}$*, where* $z_{i,j,k}$ *is the number of appearances of the hashed SAX pattern* k *in the signal* j *of time series* i*.*

The Bag-of-Receptive-Fields $\mathcal{Z}$ can be flattened into $Z \in \mathbb{N}^{n \times r}$, where r is the total number of patterns across channels, i.e., $r = dh$. An example of such a representation is shown in Figure 1 (top-right), where a time series is represented as four patterns (counts are omitted for better readability). Z can then be used as a training set for any standard tabular classifier, g, offering the advantage of interpretable features, specifically, the count of occurrences of each pattern within a time series. Finally, the classifier can be interpreted using any standard explainer, such as feature importance-based methods like SHAP [30].

Definition 5 (Feature Importance). *Given a single row of* Z*,* $\mathbf{z} \in \mathbb{R}^r$*, a feature importance matrix,* $\Phi = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_r] \in \mathbb{R}^{r \times c}$*, contains the contribution of each feature value* $z \in \mathbf{z}$ *towards predicting each possible class* c*.*

In the following section, we exploit feature importance in the Bag-of-Receptive-Fields semantic space, and propose a counterfactual technique based on symbolic patterns, which iteratively produces interpretable perturbations to generate counterfactual explanations for time series black-box classifiers.

4 Methodology

In this section, we introduce MASCOTS, a Model-Agnostic Symbolic COunterfactual explanation method for Time Series classification. MASCOTS combines the Bag-of-Receptive-Fields (BoRF) representation with symbolic transformations inspired by Symbolic Aggregate Approximation (SAX) to enhance interpretability while maintaining fidelity to the original data and model.

MASCOTS takes as input a time series X, a surrogate model g, a training dataset $\mathcal{X}$ for the surrogate, a black-box model b, an attribution method e, and a penalty hyperparameter λ. In essence, MASCOTS trains a surrogate model on a Bag-of-Receptive-Fields representation of the time series dataset. This surrogate serves as an interpretable proxy for the black-box, allowing the extraction of pattern relevance for classification using a feature attribution method. The resulting feature importance then guides semantic perturbations of the time series by modifying symbolic words. The approach is detailed step-by-step in the following sections, and illustrated in Figure 1.

4.1 Counterfactual Generation Process

The pseudo-code of MASCOTS is reported in Algorithm 1. The first step is to train an interpretable surrogate model (lines 1-3). To achieve this, MASCOTS requires a training set $\mathcal{X}$, and obtains the corresponding black-box predictions, $\hat{\mathbf{y}} = b(\mathcal{X})$ (line 1). Next, the training data $\mathcal{X}$ is transformed into a Bag-of-Receptive-Fields

Algorithm 1: MASCOTS

Data: X - time series to explain, $\mathcal{X}$ - surrogate training dataset, g - surrogate model, b - black-box model, e - attribution method, λ - penalty
Result: Counterfactual X'

1 $\hat{\mathbf{y}} \leftarrow b(\mathcal{X})$; // Predict classes of surrogate dataset
2 $Z \leftarrow BoRF(\mathcal{X})$; // Convert surrogate dataset into Bag-of-Receptive-Fields
3 $g \leftarrow train(g, Z, \hat{\mathbf{y}})$; // Train surrogate
4 $\hat{y} \leftarrow b(X)$; // Predict class of time series to explain
5 $X' \leftarrow X$;
6 **while** $b(X') = \hat{y}$ **do** // While the black-box prediction does not change
7 $\quad \mathbf{z} \leftarrow BoRF(X')$; // Convert time series into Bag-of-Receptive-Fields
8 $\quad \Phi \leftarrow e(g, \mathbf{z})$; // Get attribution matrix
9 $\quad \Delta \leftarrow \text{GetPerturbation}(X', \hat{y}, \mathbf{z}, \Phi, \lambda)$; // Generate perturbation
10 $\quad X' \leftarrow X' + \Delta$; // Perturb time series
11 **return** X'

representation, Z, using BoRF [36] (line 2). Finally, the surrogate model is trained on Z and $\hat{\mathbf{y}}$, effectively learning to approximate the black-box predictions on the given dataset (line 3). The black-box is also used to predict the label of the time series whose prediction we are explaining, i.e., $\hat{y} = b(X)$ (line 4). Then, X', a copy of X, is created (line 5), and the counterfactual generation loop begins. The condition of the counterfactual generation loop (line 6) checks whether the black-box prediction on the perturbed time series X' matches that of the original time series X. While this is true, the loop proceeds as follows. X' is converted into a Bag-of-Receptive-Fields vector, $\mathbf{z}$, (line 7), the attribution method is used to produce a feature importance matrix, Φ, containing the contribution of each pattern value in $\mathbf{z}$ towards each class (line 8). Then, the GetPerturbation procedure is invoked to generate the perturbation (line 9), which is subsequently added to the time series (line 10). After this, the loop condition is re-evaluated, and the final counterfactual, X', is eventually returned.

Using an iterative algorithm reinforces the **validity** of generated counterfactuals by gradually shifting the black-box model's prediction toward the target class, as illustrated in Figure 2. However, multiple changes, even if meaningful, may overly distort the counterfactual, reducing plausibility. Similar to [41], we set a task-dependent iteration limit to balance these factors, halting the algorithm regardless of success. As shown in Section 5, the required iterations remain low on average, preventing excessive modifications.

4.2 Pattern Swapping

We illustrate the GetPerturbation procedure in Algorithm 2. It begins by utilizing the feature attribution matrix to identify the most important pattern for the predicted class, denoted as k^+. This corresponds to the symbolic word that has the highest relevance toward the classification outcome assigned to the time series X by the black-box model (line 1). We restrict the search to *contained*

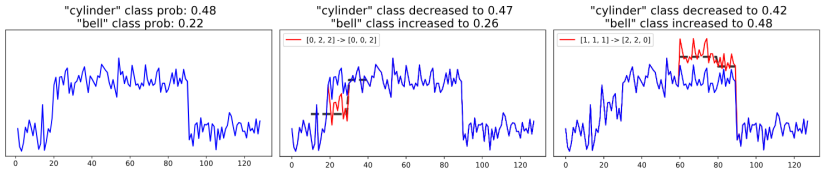

Fig. 2. Example of the iterative process of MASCOTS: a time series is gradually perturbed from a *cylinder* to a *bell*. Inserted patterns are marked by the black dashed line.

patterns, i.e., patterns where $z_k > 0$, as we want to perturb an existing pattern in X. Using the inverse hash function provided by BoRF, k^+ is then mapped back to its corresponding pattern vector (line 2), $\mathbf{p}^+$. Next, GetPerturbation identifies the most relevant pattern index opposing the predicted class, k^-, i.e., the index of the symbolic word that most strongly influences a classification different from $\hat{y}$ (line 3). Similar to line 2, the corresponding pattern vector, $\mathbf{p}^-$, is retrieved (line 4). The swapping step alters only these two patterns while the others remain unchanged, thus intrinsically encouraging the perturbation's **sparsity**. The penalty parameter λ is applied to penalize patterns that deviate significantly from $\mathbf{p}^+$ to ensure that the transformation remains meaningful and does not introduce too unrealistic modifications. A high value of λ further encourages selecting similar patterns, reducing the number of altered elements in the time series. This constraint enhances the **proximity** property of the counterfactual, ensuring that modifications remain minimal. At the same time, it enforces the algorithm to choose locally suboptimal steps, potentially requiring more steps to create the counterfactual. Proximity and sparsity, paired with the fact that the swap is performed between two patterns that both exist in the training dataset, $\mathcal{X}$, push the generated perturbation to remain within a reasonable semantic range, i.e., they promote **plausibility**. The vector $\mathbf{p}^+$ is then aligned to the time series to determine the channel, j, and starting timestamp, t, in X' where the perturbation will be applied (line 5). Since a given pattern can have multiple valid alignments within the time series, a random index is selected from the available options to introduce variability while maintaining realism. Finally, the perturbation matrix, Δ, is initialized (line 6) and populated using the PatternSwap function (line 7).

The PatternSwap function operates on a time series X and takes as input the two SAX patterns, $\mathbf{p}^+ = [\alpha_1^+, \ldots, \alpha_l^+]$ and $\mathbf{p}^- = [\alpha_1^-, \ldots, \alpha_l^-]$. Its primary objective is to perturb the subsequence from channel j, starting at index t, $\mathbf{x}_{j,t:t+w} = [x_{j,t}, \ldots, x_{j,t+w}]$, which corresponds to the pattern $\mathbf{p}^+$, so that when SAX is applied, this subsequence is instead encoded as $\mathbf{p}^-$. Let μ and σ be the mean and standard deviation of the subsequence, and let $\bar{\mathbf{x}} = [\bar{x}_1, \ldots, \bar{x}_l]$ be its segmented representation obtained via PAA, where each segment consists of w/l observations, which are averaged within that segment. To achieve the transformation, we first define the perturbation needed to shift the segmented value $\bar{x}_i$ from its original symbolic representation α_i^+ to the target α_i^-. This

Algorithm 2: GetPerturbation

Data: X - time series to swap, $\hat{y}$ - predicted class, $\mathbf{z}$ Bag-of-Receptive-Fields, Φ - attribution map, λ - penalty
Result: Perturbation - Δ

1 $k^+ \leftarrow \arg\max_{k:z_k \neq 0} \phi_{k,\hat{y}}$; // Id of most important word for prediction
2 $\mathbf{p}^+ \leftarrow hash^{-1}(k^+)$; // Retrieve important word for prediction
3 $k^- \leftarrow \arg\min_k \phi_{k,\hat{y}} + \lambda \|\mathbf{p}^+ - \mathbf{p}_k\|_1$; // Id of most important word against prediction
4 $\mathbf{p}^- \leftarrow hash^{-1}(k^-)$; // Retrieve most important word against prediction
5 $j, t \leftarrow align(X, \mathbf{p}^+)$; // Find pattern channel and timestamp alignment
6 $\Delta \leftarrow \mathbf{0}^{d \times m}$; // Initialize empty perturbation matrix
7 $\Delta_{j,t:t+w} \leftarrow \text{PatternSwap}(X, \Delta, \mathbf{p}^+, \mathbf{p}^-)$; // Swap pattern
8 **return** Δ

perturbation is given by the difference between $\bar{x}_i$ and the central value of the bin corresponding to α_i^-, denoted as $q^{\alpha_i^-}$. The perturbation is then denormalized using the inverse standardization formula, incorporating the subsequence's mean μ and standard deviation σ. Formally,

$$\delta_i = (q^{\alpha_i^-} - \bar{x}_i)\sigma + \mu, \tag{1}$$

where δ_i represents changes that must be added to all observations corresponding to $\bar{x}_i$ to move them into a different breakpoint bin. To apply this perturbation to the original subsequence $\mathbf{x}_{j,t:t+w}$, we produce $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_1, \ldots, \delta_l, \ldots, \delta_l]$ such that each δ_i is repeated w/l times. Thus, the perturbation is simply $\Delta_{j,t:t+w} = \boldsymbol{\delta}$, which can be summed to the original time series X (line 10, Algorithm 1). This transformation ensures that the local structure of the time series is preserved while allowing flexible modifications. Importantly, the locality of perturbations is not strictly enforced. A single perturbation can potentially influence an extended portion of the time series, depending on the subsequence length corresponding to the most important pattern $\mathbf{p}^+$. Even if the final modification to the time series is relatively large, it remains interpretable, as it can be succinctly described using only l values, where each subsequence segment of length w/l is shifted by a single scalar, reducing cognitive complexity.

5 Experiments

In this section, we assess MASCOTS on both univariate and multivariate time series classification datasets from the UEA and UCR repositories [8], comparing its performance against state-of-the-art methods[1]. We use the datasets listed in Table 1, all of which have been featured in multiple studies introducing counterfactuals for time series [41,27].

[1] The code is available at `https://github.com/ModelOriented/mascots`.

As baseline counterfactual explainers, we employ Glacier [41] and M-CELS [27], two recently proposed methods that achieve strong results without requiring extensive training of generative adversarial networks or multiple runs of genetic algorithms. For Glacier, we use its "uniform" variant, which offers the best trade-off in terms of *proximity* and *sparsity*. Gradients are computed in the latent space of a 1D-CNN autoencoder, as this setup has been shown to yield the highest validity scores. Due to Glacier's limitations, we restrict its evaluation to univariate data. For M-CELS, we adhere to the hyperparameter settings recommended by the authors, except for disabling `tvnorm` and `budget`, which we empirically found to improve validity. We use InceptionTime [17] as a black-box model due to its state-of-the-art performance in TSC and ability to provide model gradients required by M-CELS and Glacier. While MASCOTS does not rely on gradients, selecting InceptionTime ensures a fair comparison. Additionally, we include a qualitative example with MultiRocket-Hydra [10], a model that lacks gradient access and is therefore incompatible with most counterfactual explainers.

MASCOTS is parametrized by the choice of λ, the surrogate model, the attribution method, and the configuration of the BoRF transformation. Based on preliminary experiments, we set $\lambda \in \{0.0, 0.1\}$. This choice reflects a trade-off between proximity and validity that can vary depending on the dataset and task. In practice, while $\lambda = 0.1$ generally provided better overall performance, $\lambda = 0.0$ often remained a reliable default, highlighting the robustness of the method to this hyperparameter. The surrogate model, g, is a shallow neural network with two hidden layers of size 256, followed by a softmax function. Each layer, except the last one, is followed by a ReLU activation function. The network is trained by minimizing the categorical cross-entropy using ADAM as the optimizer [22], with a constant learning rate equal to 0.2, weight decay equal to 0.1, and dropout equal to 0.2. For each experiment, the batch size is set to 8, and the network is trained for a maximum of 1000 epochs, with early stopping triggered after 200 epochs without improvement on the validation score. After training, the checkpoint of the model that performs best on the validation set is retrieved and used in subsequent experiments. As the attribution method, e, we use Deep SHAP [30] with default hyperparameters.

BoRF automatically adapts its main hyperparameters to each dataset, taking into account the number of time steps and channels. To preserve the contiguity of subsequences, crucial for interpretability, dilation is fixed at 1, ensuring that symbols correspond to consecutive points in the original series. This avoids sparse perturbations, which can obscure the meaning of the generated counterfactual. Additionally, the alphabet size is set to 3, and the stride is defined as w/l (the ratio of word size to word length), enabling an efficient and informative transformation. Other parameters, including the number of SAX configurations, word size w, and length l, are dynamically adjusted to capture both local and global patterns. For each dataset, the smallest SAX words contain at least 8 points, while the largest are the highest power of two not exceeding the time series length. Each SAX word consists of either 2 or 4 symbols. The algorithm is limited to a maximum of 20 iterations to mitigate adversarial effects.

Table 1. Datasets description follows the notation introduced in Section 3: n (instances), d (channels), m (points), and c (classes). $ACC(b)$ and $FID(g)$ denote the accuracy of InceptionTime and the fidelity of MASCOTS' surrogate, i.e., how well $g(X)$ mimics $b(X)$, measured in terms of accuracy, respectively.

Dataset	n	d	m	c	$ACC(b)$	$FID(g)$
TwoLeadECG	23	1	82	2	1.00	1.00
GunPoint	50	1	150	2	1.00	1.00
Earthquakes	322	1	512	2	0.68	0.82
Coffee	28	1	286	2	1.00	1.00
Wine	57	1	234	2	0.80	0.84
ItalyPowerDemand	67	1	24	2	0.97	0.91
BasicMotions	40	6	100	4	0.50	1.00
Cricket	108	6	1197	12	0.24	0.60
Epilepsy	137	3	206	4	0.30	0.90
RacketSports	151	6	30	4	0.37	0.64

Evaluation Measures. We adopt the primary measures of counterfactual quality reported in prior studies [41,26]: *validity*, *proximity*, *sparsity*, and *plausibility*. The first three are formally defined below, while plausibility is assessed using Isolation Forest [29], where we report the fraction of counterfactuals classified as nominal (non-outliers). Additionally, we provide runtime comparisons for all algorithms using the same hardware setup[2].

Validity measures the proportion of generated counterfactuals X'_i that lead to a different classifier prediction compared to the original instance X_i. It is defined as $\textit{validity}(\mathcal{X}, \mathcal{X}') = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}[f(X_i) \neq f(X'_i)]$, where $\mathbb{1}[f(X_i) \neq f(X'_i)]$ is an indicator function that equals 1 if the model's prediction changes and 0 otherwise. A high validity score indicates that most counterfactuals effectively alter the model's decision. Proximity quantifies the average distance between original instances and their corresponding counterfactuals: $\textit{proximity}(\mathcal{X}, \mathcal{X}') = \frac{1}{n \cdot d \cdot m}\sum_{i=1}^{n} \|X_i - X'_i\|$, where $\|X_i - X'_i\|$ represents the distance between X_i and X'_i. Lower proximity values indicate that counterfactuals remain close to the original data points, making them more realistic and interpretable. Sparsity captures the fraction of features that remain unchanged between the original and counterfactual instances: $\textit{sparsity}(\mathcal{X}, \mathcal{X}') = \frac{1}{n \cdot d \cdot m}\sum_{i=1}^{n}\sum_{j=1}^{d}\sum_{t=1}^{m} \mathbb{1}[x_{i,j,t} - x'_{i,j,t} = 0]$, where $\mathbb{1}[x_{i,j,t} - x'_{i,j,t} = 0]$ equals 1 if a feature remains unchanged and 0 otherwise. Higher sparsity values indicate that fewer features are modified, promoting more interpretable counterfactual explanations.

Results. The experimental results for MASCOTS and the baseline models are illustrated through box plots in Figure 3 and summarized with mean and standard deviation values in Table 2. For *validity*, MASCOTS and Glacier perform particularly well on univariate data, achieving approximately 41%–44% valid counterfactuals

[2] System: 8 cores AMD Rome 7742, 32GB RAM.

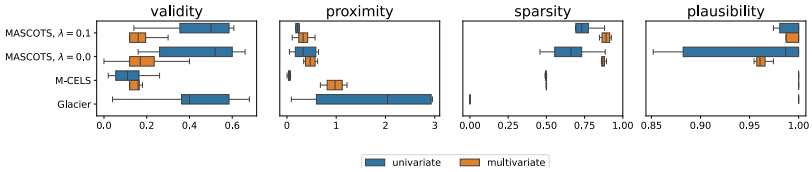

Fig. 3. Box-plots of evaluation measures. For *validity*, *sparsity*, and *plausibility*, the higher score is better, while for *proximity*, the smaller the better. MASCOTS stands out in *sparsity* with a decent *validity* and *proximity*. The difference between MASCOTS-$\lambda = 0.0$ and MASCOTS-$\lambda = 0.1$ suggests a trade-off between validity and other measures.

Table 2. Mean and standard deviation for the various evaluation measures aggregated over univariate and multivariate datasets. For MASCOTS, we add the average number of iterations required to flip a label. Best values are highlighted in bold.

		validity↑	*proximity* ↓	*sparsity* ↑	*plausibility* ↑	# iter. ↓
univariate	MASCOTS$_{\lambda=0.1}$	$\mathbf{0.44 \pm 0.18}$	0.24 ± 0.18	$\mathbf{0.71 \pm 0.12}$	0.98 ± 0.02	3.0 ± 1.0
	MASCOTS$_{\lambda=0.0}$	$\mathbf{0.44 \pm 0.21}$	0.35 ± 0.25	0.65 ± 0.15	0.84 ± 0.29	$\mathbf{2.7 \pm 1.3}$
	M-CELS	0.11 ± 0.08	$\mathbf{0.08 \pm 0.10}$	0.49 ± 0.00	0.99 ± 0.01	–
	Glacier	0.41 ± 0.23	2.32 ± 2.34	0.00 ± 0.00	$\mathbf{1.00 \pm 0.00}$	–
multiv.	MASCOTS$_{\lambda=0.1}$	0.15 ± 0.12	$\mathbf{0.33 \pm 0.19}$	$\mathbf{0.88 \pm 0.03}$	0.98 ± 0.02	$\mathbf{1.7 \pm 2.7}$
	MASCOTS$_{\lambda=0.0}$	$\mathbf{0.18 \pm 0.16}$	0.47 ± 0.13	0.87 ± 0.01	0.96 ± 0.00	2.4 ± 3.6
	M-CELS	0.12 ± 0.08	0.96 ± 0.23	0.49 ± 0.00	$\mathbf{1.00 \pm 0.00}$	–

on average. On multivariate datasets, both configurations of MASCOTS outperform M-CELS, although the overall low performance of both methods on multivariate data suggests the need for further investigation in future work. Notably, setting the λ parameter to 0.0 slightly improves the validity of counterfactuals generated by MASCOTS. In terms of *proximity*, MASCOTS and M-CELS produce counterfactuals that remain reasonably close to the original observations, while Glacier generates counterfactuals that are significantly more distant. For multivariate datasets, MASCOTS consistently outperforms M-CELS in proximity. Regarding *sparsity*, MASCOTS excels, altering fewer than 30% of the original features on average. In contrast, Glacier, due to its gradient-based nature, modifies every point at least slightly, preventing it from generating sparse explanations. M-CELS, on the other hand, alters approximately 50% of the time series to construct a counterfactual. Furthermore, we report the average number of iterations (i.e., the number of pattern swaps) required by MASCOTS to generate a counterfactual. This number varies across datasets and configurations but typically does not exceed 3. This suggests that, on average, MASCOTS can produce effective counterfactuals with only three meaningful semantic modifications. Finally, regarding runtime, performance is comparable with an average of 23 minutes for MASCOTS, 26 minutes for Glacier, and 55 minutes for M-CELS.

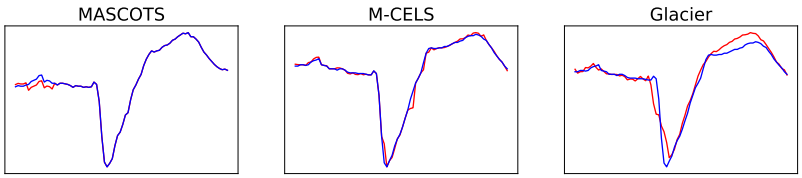

Fig. 4. Example of MASCOTS on the *TwoLeadECG* dataset to explain InceptionTime. MASCOTS is able to create sparse counterfactual which maintain its local structure. On the other hand, M-CELS and Glacier produce small (perhaps adversarial) undesirable changes to the original time series, varying in this way its initial shape.

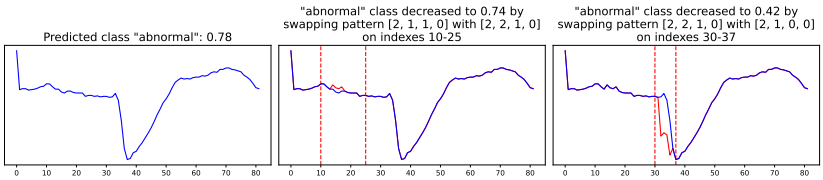

Fig. 5. Example of MASCOTS on the *TwoLeadECG* dataset to explain MultiRocket-Hydra. Changes are presented both as visualization and natural language. If positive and negative patterns have corresponding symbols (for example, the 1st, 3rd, and 4th symbols in the middle plot), MASCOTS does not change them.

Qualitative Examples. In Figure 4, we present the counterfactuals (in red) generated by each analyzed method for a randomly selected time series (in blue) from the *TwoLeadECG* dataset [8], which represents heart ECG signals. For MASCOTS, we set $\lambda = 0.1$ to achieve better proximity and sparsity. In this case, MASCOTS generates a counterfactual by modifying only a single pattern at the beginning of the signal while preserving its local structure. The changes produced by M-CELS are also minimal but are distributed across the signal. In contrast, Glacier introduces significant alterations to the original observation, disrupting the local structure of the signal.

We also provide an example on *TwoLeadECG*, focusing on explaining the MultiRocket-Hydra black-box model [10]. Unlike fully neural network-based models, MultiRocket-Hydra incorporates non-neural components, making methods such as Glacier and M-CELS inapplicable. The explanation, provided both visually and in natural language, is illustrated in Figure 5. The counterfactual is generated in two iterations. First, MASCOTS introduces a small modification between indexes 10 and 25. While this initial change does not significantly affect the black-box model's prediction, it alters the feature importance matrix Φ, enabling MASCOTS to identify a more impactful transformation in the second iteration. This final modification broadens the valley in the signal, ultimately flipping the model's classification from "abnormal" to "normal." The generated counterfactual can be expressed in natural language as follows: *"To obtain a counterfactual for*

an abnormal ECG, the pattern in indexes 10–25 *must be replaced with* $[2, 2, 1, 0]$*, followed by replacing the pattern in indexes* 30–37 *with* $[2, 1, 0, 0]$*."* These patterns differ in length—16 and 8, respectively—demonstrating the flexibility of MASCOTS in adapting its transformations. In both this example and Figure 5, the symbols 0, 1, and 2 correspond to "low," "medium," and "high" values in the time series, respectively, providing an intuitive interpretation of the modifications. This interpretation could be further enriched by domain experts, who may map these symbolic values onto a domain-specific representation space.

Indeed, although the symbolic representations used in MASCOTS are structurally syntactic, we refer to the resulting explanations as semantic to emphasize their interpretive potential. The symbolic substitutions themselves operate over abstract pattern spaces, but their interpretability arises when these patterns are contextualized through domain-specific knowledge. In practical scenarios, domain experts are often able to associate particular symbolic motifs with meaningful physiological events, behavioral signatures, or system states. Thus, while the mechanics of MASCOTS rely on syntactic operations, the explanations it produces are inherently semantic to the extent that they can be interpreted and acted upon by humans within a given application context.

6 Conclusion

In this article, we have introduced MASCOTS, a model-agnostic method for generating counterfactual explanations in univariate and multivariate time series classification. By leveraging the BoRF transformation and symbolic representations, MASCOTS enhances interpretability while maintaining fidelity to the original black-box model. Unlike prior approaches, it operates in a fully agnostic manner without the need of autoencoders and without relying on distances or nearest unlike neighbors. Our evaluation demonstrates its effectiveness, achieving high interpretability and sparsity while preserving validity and proximity.

A limitation of our approach, shared with existing competitors, is the relatively low validity performance on multivariate data. This highlights a research gap in the development of counterfactual methods tailored for multivariate time series, as well as the need for broader empirical evaluations to understand the underlying causes of this limitation. To address this, we plan to evaluate MASCOTS on more tasks and challenging real-world scenarios, such as the satellite telemetry domain. Further, we aim to extend our approach into a "user-in-the-middle" framework, allowing expert intervention at each iteration to refine counterfactual explanations. By selecting among proposed changes, experts can enhance the plausibility of generated counterfactuals, explore custom "what-if?" scenarios, and even create artificial observations for manual labelling and integration into training datasets. This interactive approach could further improve the adaptability and utility of counterfactual explanations in real-world applications.

Acknowledgments. This study has been partially funded by the European Space Agency grant 4000144194/23/D/BL "Assurance for space domain AI applications", by the SONATA BIS grant 2019/34/E/ST6/00052 funded by Polish National Science Centre (NCN), and by the Italian Project Fondo Italiano per la Scienza FIS00001966 "MIMOSA",

by the European Community Horizon 2020 programme under the funding schemes ERC-2018-ADG G.A. 834756 "XAI", G.A. 101070212 "FINDHR", G.A. 101120763 "TANGO", by the European Commission under the NextGeneration EU programme – National Recovery and Resilience Plan (Piano Nazionale di Ripresa e Resilienza, PNRR) Project: "SoBigData.it – Strengthening the Italian RI for Social Mining and Big Data Analytics" – Prot. IR0000013 – Av. n. 3264 del 28/12/2021, and M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR" - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI". The authors gratefully acknowledge the contributors of the datasets available through the UEA & UCR Repositories. Their efforts in collecting and sharing these datasets have been invaluable to this research.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ates, E., Aksar, B., Leung, V.J., Coskun, A.K.: Counterfactual explanations for multivariate time series. In: 2021 international conference on applied artificial intelligence (ICAPAI). pp. 1–8 (2021)
2. Bahri, O., Li, P., Boubrahimi, S.F., Hamdi, S.M.: Temporal rule-based counterfactual explanations for multivariate time series. In: 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 1244–1249 (2022)
3. Bahri, O., Li, P., Filali Boubrahimi, S., Hamdi, S.M.: Discord-based counterfactual explanations for time series classification. Data Mining and Knowledge Discovery **38**(6), 3347–3371 (2024)
4. Baydogan, M.G., Runger, G., Tuv, E.: A bag-of-features framework to classify time series. IEEE transactions on pattern analysis and machine intelligence **35**(11), 2796–2802 (2013)
5. Bodria, F., Giannotti, F., Guidotti, R., Naretto, F., Pedreschi, D., Rinzivillo, S.: Benchmarking and survey of explanation methods for black box models. Data Mining and Knowledge Discovery pp. 1–60 (2023)
6. Chen, Z., Silvestri, F., Wang, J., Zhu, H., Ahn, H., Tolomei, G.: Relax: Reinforcement learning agent explainer for arbitrary predictive models. In: Proceedings of the 31st ACM international conference on information & knowledge management. pp. 252–261 (2022)
7. Dasarathy, B.V.: Nearest unlike neighbor (nun): an aid to decision confidence estimation. Optical Engineering **34**(9), 2785–2792 (1995)
8. Dau, H.A., Bagnall, A., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Keogh, E.: The ucr time series archive. IEEE/CAA Journal of Automatica Sinica **6**(6), 1293–1305 (2019)
9. Delaney, E., Greene, D., Keane, M.T.: Instance-based counterfactual explanations for time series classification. In: Case-Based Reasoning Research and Development. pp. 32–47 (2021)
10. Dempster, A., Schmidt, D.F., Webb, G.I.: Hydra: Competing convolutional kernels for fast and accurate time series classification. Data Mining and Knowledge Discovery **37**(5), 1779–1805 (2023)
11. Der, A., Yeh, C.C.M., Zheng, Y., Wang, J., Zhuang, Z., Wang, L., Zhang, W., Keogh, E.: Pupae: Intuitive and actionable explanations for time series anomalies. In: Proceedings of the 2024 SIAM International Conference on Data Mining (SDM). pp. 37–45 (2024)

12. Filali Boubrahimi, S., Hamdi, S.M.: On the mining of time series data counterfactual explanations using barycenters. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 3943–3947 (2022)
13. Glenis, A., Vouros, G.A.: Scale-boss-mr: Scalable time series classification using multiple symbolic representations. Applied Sciences **14**(2), 689 (2024)
14. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. Data Mining and Knowledge Discovery **38**(5), 2770–2824 (2024)
15. Guidotti, R., Monreale, A., Giannotti, F., Pedreschi, D., Ruggieri, S., Turini, F.: Factual and counterfactual explanations for black box decision making. IEEE Intelligent Systems **34**(6), 14–23 (2019)
16. Höllig, J., Kulbach, C., Thoma, S.: Tsevo: Evolutionary counterfactual explanations for time series classification. In: 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 29–36 (2022)
17. Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D.F., Weber, J., Webb, G.I., Idoumghar, L., Muller, P.A., Petitjean, F.: Inceptiontime: Finding alexnet for time series classification. Data Mining and Knowledge Discovery **34**(6), 1936–1962 (2020)
18. Karimi, A.H., Barthe, G., Balle, B., Valera, I.: Model-agnostic counterfactual explanations for consequential decisions. In: International conference on artificial intelligence and statistics. pp. 895–905. PMLR (2020)
19. Karlsson, I., Papapetrou, P., Boström, H.: Generalized random shapelet forests. Data mining and knowledge discovery **30**, 1053–1085 (2016)
20. Karlsson, I., Rebane, J., Papapetrou, P., Gionis, A.: Locally and globally explainable time series tweaking. Knowl. Inf. Syst. **62**(5), 1671–1700 (2020)
21. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Knowledge and information Systems **3**, 263–286 (2001)
22. Kingma, D.P.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Labaien, J., Zugasti, E., De Carlos, X.: Contrastive explanations for a deep learning model on time-series data. In: International Conference on Big Data Analytics and Knowledge Discovery. pp. 235–244 (2020)
24. Le Nguyen, T., Gsponer, S., Ilie, I., O'reilly, M., Ifrim, G.: Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations. Data mining and knowledge discovery **33**, 1183–1222 (2019)
25. Li, P., Bahri, O., Boubrahimi, S.F., Hamdi, S.M.: Attention-based counterfactual explanation for multivariate time series. In: International Conference on Big Data Analytics and Knowledge Discovery. pp. 287–293 (2023)
26. Li, P., Bahri, O., Boubrahimi, S.F., Hamdi, S.M.: Cels: Counterfactual explanations for time series data via learned saliency maps. In: 2023 IEEE International Conference on Big Data (BigData). pp. 718–727 (2023)
27. Li, P., Bahri, O., Boubrahimi, S.F., Hamdi, S.M.: M-cels: Counterfactual explanation for multivariate time series data guided by learned saliency maps. arXiv preprint arXiv:2411.02649 (2024)
28. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing sax: a novel symbolic representation of time series. Data Mining and knowledge discovery **15**, 107–144 (2007)
29. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 eighth ieee international conference on data mining. pp. 413–422 (2008)
30. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. Advances in neural information processing systems **30** (2017)

31. Middlehurst, M., Large, J., Flynn, M., Lines, J., Bostrom, A., Bagnall, A.: Hive-cote 2.0: a new meta ensemble for time series classification. Machine Learning **110**(11), 3211–3243 (2021)
32. Middlehurst, M., Schäfer, P., Bagnall, A.: Bake off redux: a review and experimental evaluation of recent time series classification algorithms. Data Mining and Knowledge Discovery **38**(4), 1958–2031 (2024)
33. Nguyen, T.L., Ifrim, G.: Mrsqm: Fast time series classification with symbolic representations. arXiv preprint arXiv:2109.01036 (2021)
34. Refoyo, M., Luengo, D.: Sub-space: Subsequence-based sparse counterfactual explanations for time series classification problems. In: World Conference on Explainable Artificial Intelligence. pp. 3–17 (2024)
35. Ruiz, A.P., Flynn, M., Large, J., Middlehurst, M., Bagnall, A.: The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data mining and knowledge discovery **35**(2), 401–449 (2021)
36. Spinnato, F., Guidotti, R., Monreale, A., Nanni, M.: Fast, interpretable, and deterministic time series classification with a bag-of-receptive-fields. IEEE Access **12**, 137893–137912 (2024)
37. Spinnato, F., Guidotti, R., Monreale, A., Nanni, M., Pedreschi, D., Giannotti, F.: Understanding any time series classifier with a subsequence-based explainer. ACM Transactions on Knowledge Discovery from Data **18**(2), 1–34 (2023)
38. Spinnato, F., Landi, C.: Pyrregular: A unified framework for irregular time series, with classification benchmarks. arXiv preprint arXiv:2505.06047 (2025)
39. Theissler, A., Spinnato, F., Schlegel, U., Guidotti, R.: Explainable ai for time series classification: A review, taxonomy and research directions. IEEE Access (2022)
40. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the gdpr. Harv. JL & Tech. **31**, 841 (2017)
41. Wang, Z., Samsten, I., Miliou, I., Mochaourab, R., Papapetrou, P.: Glacier: guided locally constrained counterfactual explanations for time series classification. Machine Learning pp. 1–31 (2024)
42. Wang, Z., Samsten, I., Mochaourab, R., Papapetrou, P.: Learning time series counterfactuals via latent space representations. In: Discovery Science: 24th International Conference, DS 2021. pp. 369–384 (2021)
43. Ye, L., Keogh, E.: Time series shapelets: a new primitive for data mining. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 947–956 (2009)
44. Yeh, C.C.M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H.A., Silva, D.F., Mueen, A., Keogh, E.: Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In: 2016 IEEE 16th international conference on data mining (ICDM). pp. 1317–1322 (2016)