# TSHAP: Fast and Exact SHAP for Explaining Time Series Classification and Regression

Thach Le Nguyen and Georgiana Ifrim

University College Dublin, Dublin, Ireland
{thach.lenguyen,georgiana.ifrim}@ucd.ie

**Abstract.** Attribution methods are essential for interpreting time series predictive models by quantifying the relevance of each time step for the prediction. State-of-the-art methods are often based on SHAP, an attribution method developed for tabular data. However, this has several challenges. First, SHAP is expensive to compute, especially for long time series, hence to speed it up it is usually approximated. Second, the impact of the background selection for emulating data 'missingness', essential to compute SHAP, remains understudied. Third, SHAP and more generally attribution methods for time series regression are notably lacking. In this paper, we address these limitations and propose TSHAP, a novel SHAP-based attribution method for time series classification and regression. TSHAP leverages a sliding window to group temporal data, enabling the **efficient computation of exact SHAP** values for each group. We further develop a methodology for the principled selection of background data. We evaluate TSHAP's performance and robustness using comprehensive experiments on synthetic and real-world time series datasets.

**Keywords:** Explainable AI · Time Series · Exact SHAP · Evaluation

## 1 Introduction

Attribution methods quantify the relevance of each input feature for predicting the target feature by a predictive model. The computed attributions are critical tools in Explainable Artificial Intelligence (XAI) to explain black-box models [11, 12]. Time series data are numerical data measured over a time period, where each value in the time series corresponds to a time step in this period. Time series data can be extremely long (e.g., millions of time steps) and have multiple channels (multivariate time series data). Given an input time series and a predictive model, the attribution methods calculate the attribution (relevance score) for each time step in the time series using only the output of the model. The attribution signifies the relevance of that time step to the prediction of the model given the input time series (e.g., Figure 1). These sample-specific attributions are also referred to as local attributions.

Several attribution methods have been introduced for tabular data, of which LIME [15] and SHAP [9] are the most prominent. SHAP, based on game theory,

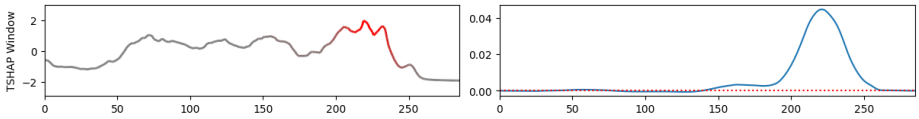


Fig. 1: Explanation of the ROCKET classifier using an attribution method. The left panel depicts a time series from the Coffee dataset of the UCR/UEA Archive, and the right panel shows the attribution profile. The attribution profile indicates the 200-250 range as the most relevant part of the time series for the classifier.

calculates the attribution as the Shapley value of each feature. The Shapley value is the contribution of the feature towards the overall payout, i.e., the prediction of the model. The exact SHAP method calculates the contribution of every possible feature coalition (i.e., subset of features), therefore is known for being computationally expensive. In addition, SHAP relies on a background dataset to emulate "missingness" in the data. When SHAP calculates the contribution of a feature coalition, it essentially substitutes the values of the missing features (features not in the coalition) with values drawn from the background dataset. In some works, the background data is also referred to as the baseline data.

While attribution methods such as LIME and SHAP can be directly applied to time series by treating each time step as an independent feature, this approach becomes computationally prohibitive for datasets with moderately long time series (e.g., length $> 100$). Furthermore, it fails to capture the inherent sequential dependencies within the time series. To mitigate these limitations, prior research [6, 13, 17] has explored grouping consecutive time steps into aggregated features. For example, a long time series can be segmented, with each segment being represented by a single feature, thereby reducing the computational burden and preserving local sequential information within each segment. While this approach reduces the computational cost, it can still increase rapidly with the number of segments. Moreover, careless segmentation can inadvertently break the sequential structure of time series. Thus effective and efficient grouping of time steps for time series attribution methods is still an ongoing challenge.

Furthermore, the influence of background data on SHAP time series explanations is critical but under-researched. Common background options include all-zero (substituting feature values with zero) and real time series data. The all-zero background assumes zero represents the absence of information, or missing data, facilitating attribution calculations. When available, real data (e.g., training data) can also serve as background. While computationally efficient, the all-zero assumption is problematic in time series. We conducted an experiment using a ROCKET [4] model trained on 49 UCR and UEA [3] binary classification datasets, predicting on all-zero time series. In 42 instances, the model exhibited high confidence (probability $> 0.95$), demonstrating that zero can be informative, contrary to the underlying assumption of absent information (which should lead to less decisive probability). Therefore, it is important to study the impact of background time series on the computed explanation and to develop robust background selection strategies.

Most work for explaining time series models focuses on time series classification (TSC), where the prediction is a discrete category [10]. Another important research area is time series extrinsic regression (TSER) [21, 16] . In comparison to TSC, TSER in general and XAI for TSER are still under-explored areas. Theoretically, both LIME and SHAP-based attribution methods can be applied for time series regression as they both work for regression problems with tabular data. However, none of the previous works explored this possibility in depth for the time series domain. The main challenge is that the interpretation of a regression model is not well-formulated, and depends again on the selected background. The main contributions of this paper to address these challenges are:

- We present TSHAP, a novel SHAP-based attribution method that can be used to explain both time series classification and regression models. This method has two variants: TSHAP Window relies on a sliding window to efficiently compute exact Shapley values for each window while TSHAP ROI (Regions of Interest) uses the window attributions to find the important regions. To our best knowledge, we are the first to study attribution methods for TSER.
- We study the impact of background selection on SHAP computation and propose methods to select suitable background data for both classification and regression tasks. The methods are applicable to any SHAP-based attribution methods, and other methods that rely on background data, for time series.
- We present experiments to evaluate TSHAP on both synthetic and real datasets for TSC and TSER. We describe our evaluation methodology using a hypothetical model and ground truth attributions for the synthetic data. The results show that TSHAP can be computed more efficiently than existing methods and achieves high explanation quality (e.g., as measured using ground truth data and faithfulness measures). All our data and code is publicly available[1].

## 2 Background and Related Work

### 2.1 Background

We define a time series $x$ as a sequence of measurements over time:

$$x = \{x_1, x_2, \ldots, x_n\} \tag{1}$$

where $x_i$ is the measurement at time step $i$ and $n$ is the length of the time series. In this paper, we only consider univariate time series, hence $x_i$ is a scalar. A predictive model $f$ is a function that maps $x$ to a target value $o$ (i.e., $f(x) = o$). For the regression task $o \in \mathbb{R}$ is a continuous value. For the classification task, typically $o \in L$ is a label, in a finite set of labels L. However, many attribution

[1] https://github.com/mlgig/tshap

methods use the class probability instead of the predicted label to calculate the attributions (i.e., $o \in \mathbb{R}$ and $0 \leq o \leq 1$). This allows these attribution methods to adapt seamlessly between classification and regression. An attribution method $M$ takes a time series $x$ and the model $f$ as the input, and outputs the attributions:

$$M(x, f) = \{\phi_1, \phi_2, \ldots, \phi_n\} \tag{2}$$

where $\phi_i$ is the attribution value of $x_i$. In general, the attribution $\phi_i$ indicates how relevant $x_i$ is to the prediction $f(x) = o$. In order to use the attributions to explain the model $f$, we first need to define the explanation question for the machine learning (ML) task.

**Interpretation of regression attributions:** For regression tasks, the authors in [8] argue that a reference value $r$ is essential for meaningful explanations. Without it, the explanation question becomes ambiguous (e.g., *"Why is 'o' predicted?"*). Introducing $r$ allows for a clearer question: *"Why is o predicted instead of r?"*. The reference value is user-defined and depends on the specific purpose of the explanation. In this context, a positive attribution ($\phi_i > 0$) indicates that the presence of $x_i$ supports a prediction greater than $r$, while a negative attribution ($\phi_i < 0$) suggests support for a prediction less than $r$.

**Interpretation of classification attributions:** For classification tasks, the explanation question is straightforward:*"Why is o predicted as the label of x?"*. Attribution values indicate which time steps support or contradict this prediction. Unlike regression, the alternatives to $o$ are finite, hence reducing ambiguity. We will also show in Section 3.1 that there is actually an implicit reference value $r$ for classification attributions. For simplicity, we focus only on binary classification (i.e., positive versus negative labels). Typically, $\phi_i > 0$ signifies that the presence of $x_i$ supports the positive prediction, increasing model confidence, while $\phi_i < 0$ indicates lower confidence. The absolute value of $\phi_i$ indicates the "strength" of the support; $\phi_i = 0$ means $x_i$ is irrelevant to the model. For multiclass prediction tasks, we can easily model them as multiple binary classification tasks.

### 2.2 Attribution Methods

Two of the most well-known attribution methods in XAI are SHAP [9] and LIME [15]. LIME is a framework to explain any black-box classification model. LIME first perturbs the instance of interest to obtain a local dataset, then trains a linear model on the perturbed data set. This linear model is a local proxy model that approximates the black-box model in the neighbourhood of the instance of interest. Thus the black-box model can be explained (locally) by interpreting the linear model. While its original paper only covers classification models, the LIME implementation has also been extended to regression models.

On the other hand, SHAP is based on Shapley values, which is a methodology in game theory to determine the contribution of each player in a collaborative

game. In the context of machine learning, a feature is a player and the payout can be obtained from the model predictions. As exact SHAP is known to be expensive to compute, alternatives including KernelSHAP [9] and Shapley Values Sampling [22] can be used to estimate the attributions quicker. KernelSHAP approximates the Shapley values by solving a linear regression problem while Shapley Values Sampling samples random permutations of input features. Both methods have been used in the time series domain [13, 17].

**LIME-based time series attribution methods.** LEFTIST [6] is one of the first attempts to adapt LIME for time series. The method simply divides the time series to equal-length segments. In the same paper, the authors propose three different methods to perturb the time series data which are constant, interpolation, and random background. On the other hand, LIMESegment [19] is a combination of techniques including NNSegment, a segmentation method and Realistic Background Perturbation, a time series perturbation method using the underlying background frequency.

**SHAP-based time series attribution methods.** WindowSHAP [13] proposes three different ways to explain a time series classification model. The common idea is to use a window to group consecutive time steps. Stationary WindowSHAP segments the time series into equal segments before applying SHAP. Sliding WindowSHAP uses a sliding window instead. Dynamic WindowSHAP uses a strategy alike to binary search in which it keeps splitting the time series until a stop condition. Once WindowSHAP segments the time series, it uses KernelSHAP to compute SHAP attribution values. While KernelSHAP is faster to compute than exact SHAP, it was shown to behave poorly for many TSC tasks [17, 24].

TimeSHAP [1] is another SHAP-based attribution method but focused on multivariate time series. TimeSHAP groups the data channel-wise (each channel is a feature) and time-step wise (each time step is a feature). Moreover, it prunes distant-past data by assuming that only recent data is more important. Interestingly, TimeSHAP detects the important events by intersecting the important channels and important time steps.

**Other Methods.** Feature Ablation implemented in Captum[2] is an extremely fast attribution method. It simply replaces the feature value with values drawn from the background data then calculates the difference in model output as the attribution for that feature. This method was found to perform well on time series data when combined with fixed segmentation [17].

LASTS [20] is an explanation framework that can explain a black-box time series classifier in three different ways: a saliency map, prototypical and counterfactual exemplars, and rule-based explanation. A saliency map is similar to attributions where the relevant parts of the input time series $x$ is highlighted. A

[2] https://captum.ai/

prototypical exemplar is an artificial time series that is similar to $x$ and classified to the same class of $x$, while a counterfactual exemplar is an artificial time series that is also similar to $x$ but classified to a different class. Rules-based explanation provides human-friendly rules such as *"If the time series $x$ has this pattern then $f(x) = 0$ otherwise $f(x) = 1$"*.

## 3 Methodology

In this section, we detail our proposed methodology to explain time series models based on exact SHAP computation.

### 3.1 Background Data and Reference Value

In general, Shapley values explain the difference between the prediction $f(x)$ and the expected prediction $E(f(X))$. The expected prediction plays the role of the empty coalition ($\emptyset$) and is estimated using the average prediction on the background data $X_b$ (i.e., $E(f(X)) = E(f(X_b))$) [11]. In other words, the Shapley value calculation (e.g., Equation 7) depends on the background data. Consequently, using different background can lead to different attributions. This issue will also be demonstrated experimentally in Section 4.3. We propose the following strategies to select proper background data for SHAP-based time series attribution methods.

**Selecting background data for regression models:** As discussed in Section 2.1, a reference value $r$ is needed for the explanation of regression models. The attribution methods then aim to explain the difference between $f(x)$ and $r$. As SHAP attributions explain the difference between $f(x)$ and $E(f(X_b))$, it implies the condition for the regression background data as:

$$E(f(X_b)) \approx r \tag{3}$$

A simple strategy to choose the background data following this condition is to draw samples from the training data such that $f(x) \approx r$.

**Selecting background data for binary classification models:** For binary classification models, if we use the output probability of the model instead of labels, then the explanation has a natural reference value which is the decision boundary of the output $r = 0.5$. As a result, the explanation question can be reformulated as *"Why is $o$ but not 0.5 the predicted probability of $x$?"*. From this perspective, the condition for the classification background data is:

$$E(f(X_b)) \approx 0.5 \tag{4}$$

We explored four classification background data options: (1) training data, (2) all-zeros, (3) training centroid, and (4) training balanced centroid. While

training data offers the most balanced expected prediction, its computational cost for SHAP can be high, due to the need for repeated SHAP computation on each training sample. Therefore, we only consider single-time-series backgrounds. The centroid (Equation 5) represents the average of all training time series. The balanced centroid (Equation 6), used for imbalanced datasets, averages class-wise centroids, mitigating majority class bias. Both centroid types are artificial time series (i.e., not actual training samples).

$$c_i = \frac{1}{|X|}\sum_{x}^{X} x_i \qquad (5) \qquad\qquad \bar{c}_i = \frac{1}{|L|}\sum_{l}^{L}(\frac{1}{|X_l|}\sum_{x}^{X_l} x_i) \qquad (6)$$

### 3.2 TSHAP: Sliding Window Attribution

TSHAP utilizes a sliding window to aggregate time steps. Time steps within the window are grouped into one feature, and those outside into another feature, effectively transforming the time series into a two-feature tabular format. Figure 2 showcases this approach.

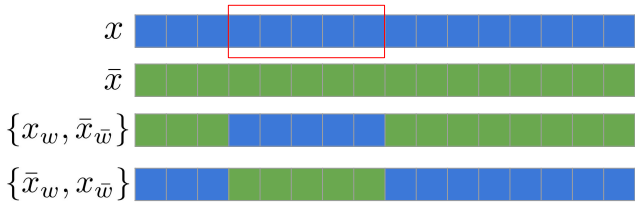


Fig. 2: An illustration of how TSHAP emulates the 'missingness' in the data with a background sample $\bar{x}$. We denote with $x$ the input time series and with $\bar{x}$ the time series where all data are missing. $\{\bar{x}_w, x_{\bar{w}}\}$ emulates that data inside the window are missing and $\{x_w, \bar{x}_{\bar{w}}\}$ emulates that data outside the window are missing.

A window $w$ is defined by its starting location $w_s$ and its length $w_l$. Let $\bar{w}$ be the group of time steps that are outside of the window $w$ (i.e., $w$ and $\bar{w}$ cover the entire time series). $x$ is the sample of interest that needs explanation and $\bar{x}$ is the background sample (drawn from the background data). The exact Shapley value of the sliding window $w$ can be calculated as follows:

$$\begin{aligned}\phi(w) &= \frac{1}{2}(f(x) - f(x_{\bar{w}}) + f(x_w) - f(\bar{x})) \\ &= \frac{1}{2}(f(x) - f(\{\bar{x}_w, x_{\bar{w}}\}) + f(\{x_w, \bar{x}_{\bar{w}}\}) - f(\bar{x}))\end{aligned} \qquad (7)$$

If there are multiple samples in the background dataset, the attribution is averaged across all the background samples (Equation 8). Finally, the attribution

of the time step $i$ is the average attribution of all the windows $w \in W_i$ that contain $i$ (Equation 9).

$$\phi(w) = \frac{1}{|X_b|} \sum_{\bar{x} \in X_b} \phi_{\bar{x}}(w) \qquad (8) \qquad\qquad \phi_i = \frac{1}{|W_i|} \sum_{w \in W_i} \frac{\phi(w)}{w_l} \qquad (9)$$

The TSHAP algorithm is summarised in Algorithm 1. Calculating the attribution for every sliding window (stride $s = 1$) can be expensive, therefore TSHAP only calculates the Shapley value after every stride (stride $s > 1$) and interpolates the attributions of the windows in between.

**Algorithm 1:** Computing TSHAP Window Attributions

**Input:** Time series sample
**Output:** Attributions
Instantiate a set of sliding windows with stride $s$ and window length $w_l$.
**foreach** *Sliding window* $w$ **do**
  **foreach** *Background sample* $\bar{x}$ **do**
    Calculate the attribution of window $w$ with Equation 7.
  Calculate the average attribution of window $w$ with Equation 8.
Interpolate the attributions of the windows in between.
Calculate the attribution of each time step with Equation 9.

### 3.3 TSHAP ROI: Using Attribution to Identify Regions of Interest

For time series data, it is often useful to identify the important regions in time series that contribute the most to the model prediction. Moreover, in our experiments, we will show that TSHAP Window can mistakenly spread the attribution from relevant time steps to the nearby irrelevant time steps. In this section, we propose an efficient technique to mitigate this issue and identify these important regions using window attributions. The exact steps to search for the regions of interest are described in Algorithm 2.

## 4 Experiments

### 4.1 Attribution Methods

In this section, we consider the following attribution methods for comparison.

- **Time Series Classification:** TSHAP, WindowSHAP, Shapley Value Sampling, Feature Ablation, LIMESegment, LEFTIST.
- **Time Series Regression:** TSHAP, WindowSHAP, Shapley Value Sampling, Feature Ablation.

**Algorithm 2:** Computing TSHAP Regions Of Interest

**Input:** Time series sample
**Output:** ROI Attributions
1 Calculate all the TSHAP window attributions with Algorithm 1 (until Line 6).
2 Each window $w_i$ is marked as relevant if $|\phi(w_i)| > \epsilon$ where $\epsilon = 0.1 \times max(|\phi(w_i)|)$ is the relevant threshold.
3 Group the consecutive relevant windows.
4 **foreach** *Group of consecutive relevant windows* **do**
5 The region of interest $w_{ROI}$ is the combination of all windows in the group.
6 Calculate the attribution of the window $\hat{w}$ with Equation 8.
7 Attribution of each time step i inside $w_{ROI}$: $\phi_i = \frac{\phi(w_{ROI})}{length(w_{ROI})}$

We note that TSHAP has two variants: TSHAP-Window and TSHAP-ROI, while WindowSHAP has three variants: Stationary, Sliding, and Dynamic. Feature Ablation and Shapley Value Sampling are also included as they were found to be effective in the quantitative evaluation of [17]. For these methods, we use the tsCaptum package [18]. LIME-based methods (LIMESegment and LEFT-IST) are excluded from the regression experiments because their current implementation does not support regression models. For a fair comparison, we set the parameter window length at 10% of the length of the time series and stride $s = 5$ for TSHAP, WindowSHAP; the number of segments = 10 for Shapley Value Sampling, Feature Ablation. All remaining hyper-parameters are left with the default values.

### 4.2 Evaluation Methodology

We evaluate our proposed methods using a synthetic dataset and real datasets from the UCR archive. For the synthetic dataset, we build a *hypothetical model* which predicts the target value using human reasoning. Then we assign an attribution value to each time step in accordance to its relevance to the *hypothetical model*. Thus these attributions are the ground truth attributions $\Phi$ of the *hypothetical model*. Finally, we explain the *hypothetical model* with the attribution methods then compare the output with the ground truth attributions using cosine similarity, precision, recall, and F1. Previous works [2, 17] also used assigned-by-human attributions but with ML models to test synthetic data; however we argue that these attributions are not the ground truth attributions of the ML models. Cosine similarity is a well known metric to quantify the similarity between two (attribution) vectors. To calculate precision, recall, and F1, we construct the confusion matrix as in Table 1.

On the other hand, faithfulness analysis is a common approach [23] to evaluate classification attribution methods with real datasets. In this approach, the most relevant part of the time series (identified by the attributions) is perturbed to create a new time series. The comparison between the new prediction (on

Table 1: Confusion matrix for evaluating the attributions $\phi$ using the ground truth $\Phi$. True relevant means both attributions are relevant in the same direction (both support or oppose the prediction). True irrelevant means both attributions are irrelevant. False relevant means $\phi_i$ is relevant but $\Phi_i$ is irrelevant or relevant in the opposite direction. False irrelevant means $\phi_i$ is irrelevant but $\Phi_i$ is relevant.

| | True | False |
|---|---|---|
| Relevant | $\phi_i \times \Phi_i > 0$ | $\phi_i \neq 0$ and $\phi_i \times \Phi_i \leq 0$ |
| Irrelevant | $\phi_i = 0$ and $\Phi_i = 0$ | $\phi_i = 0$ and $\Phi_i \neq 0$ |

the new time series) and the original prediction (on the original time series) can indicate the faithfulness of the attributions. Equation 10 and 11 show how faithfulness is calculated in our experiments where $f$ outputs probability. $x_p$ is the resulting time series from perturbing the positive attribution part; $x_n$ results from perturbing the negative attribution part. The perturbation is done by substituting the top 10% most relevant data with zero.

$$\textit{faithfulness} = \frac{1}{|X|} \sum_{x \in X} (f(x) - f(x_p) + f(x_n) - f(x)) \tag{10}$$

$$= \frac{1}{|X|} \sum_{x \in X} (f(x_n) - f(x_p)) \tag{11}$$

Existing classification faithfulness evaluations [14, 24] often measure accuracy drops after perturbation. However, this method relies on the test set ground truth and the original accuracy of the predictions; moreover perturbation can inadvertently correct misclassifications, leading to inaccurate assessments. Our analysis instead measures changes in prediction probability regardless of prediction accuracy, ensuring the direction of change aligns with attributions. Perturbing time steps with positive attributions should reduce model confidence (i.e., $f(x) - f(x_p) > 0$) while perturbing time steps with negative attributions should increase it (i.e., $f(x_n) - f(x) > 0$). Higher positive faithfulness scores indicate greater attribution fidelity.

### 4.3 Synthetic Dataset

The **synthetic time series data** generation is inspired by the work of [24]. We use the original code and adapt it so each time series is created by inserting two segments of sine wave signals ($s_1$ and $s_2$) to a silent signal at two different places (Figure 3). The length of the time series is 200. There are 30 samples for the training set and 30 samples for the test set. For the regression problem, the target value is the sum of the frequencies of the sine waves. For the classification problem, a threshold $\tau = 60$ is used to separate the samples into two classes: below the threshold (negative) and above the threshold (positive).

**The hypothetical model** estimates the frequency of each sine wave segment by counting the number of cycles in the segment. Algorithm 3 and 4 are

implemented as the predict functions for the hypothetical regression model and classification model respectively.

**Algorithm 3:** Regression Prediction

**Input:** Time series sample
**Output:** Total frequency
1 Count the number of cycles in the first segment.
2 Estimate the frequency of the first segment from the number of cycles.
3 Count the number of cycles in the second segment.
4 Estimate the frequency of the second segment from the number of cycles.
5 Return the sum of the frequencies.

**Algorithm 4:** Classification Prediction

**Input:** Time series sample
**Output:** Label of the time series sample (positive or negative)
1 Estimate the frequency sum using Algorithm 3.
2 Return positive if frequency sum $> \tau$ otherwise return negative.

**The ground truth attributions $\Phi$:** Each time step $i$ is assigned with an attribution $\Phi_i$ calculated by Equation 12 for the regression problem and Equation 13 for the classification problem:

$$\Phi_i = \begin{cases} f_j - \frac{r}{2} & \text{if } i \text{ inside } s_j \\ 0 & \text{if } i \text{ not inside } s_j \end{cases} \quad (12) \qquad \Phi_i = \begin{cases} f_j - \frac{\tau}{2} & \text{if } i \text{ inside } s_j \\ 0 & \text{if } i \text{ not inside } s_j \end{cases} \quad (13)$$

where $s_j$ ($j = 1$ or 2) is the sine wave segment and $f_j$ is the frequency of $s_j$. The idea is that if the frequency of one segment $f_j < \frac{\tau}{2}$ then it is less likely that the time series is a positive sample (i.e. sum of frequencies $> \tau$) and if $f_j > \frac{\tau}{2}$ then it is more likely that the time series is a positive sample.

**Classification Background Analysis.** In this experiment, we study the impact of the background selection on the classification attributions. We include five different choices of background: zero, centroid, balanced centroid, train data, threshold. Threshold background is a special sample that has two identical sine wave segments with the frequency $= 0.5 \times \tau$. This is an ideal background sample that is located on the decision boundary of the hypothetical classification model.

Table 2 shows the (average) cosine similarity between TSHAP attributions and the ground truth attributions. In addition, the table also includes the (average) probability of the predicted class of the background data. We note that

Table 2: Cosine similarity between TSHAP attributions and ground truth attributions. Runtime is the total time required to compute attributions for the test dataset.

| Background | Avg Cosine sim | Avg proba | Runtime(sec) |
|---|---|---|---|
| Zero | -0.174 | 1.0 | 0.06 |
| Train | **0.906** | **0.62** | 5.95 |
| Centroid | 0.708 | 0.99 | 0.06 |
| Balanced Centroid | 0.646 | 1.0 | 0.07 |
| Threshold | **0.911** | **0.54** | 0.07 |

attributions with different background data give very different results. An apparent trend is that the closer the background data is to the decision boundary (the average probability closer to 0.5), the more accurate the attributions, although using the train data comes with a higher cost of running time. This experiment highlights the issue of inappropriate background data (all-zero background in this case) and supports our proposed strategy on choosing the background data for classification attribution methods.

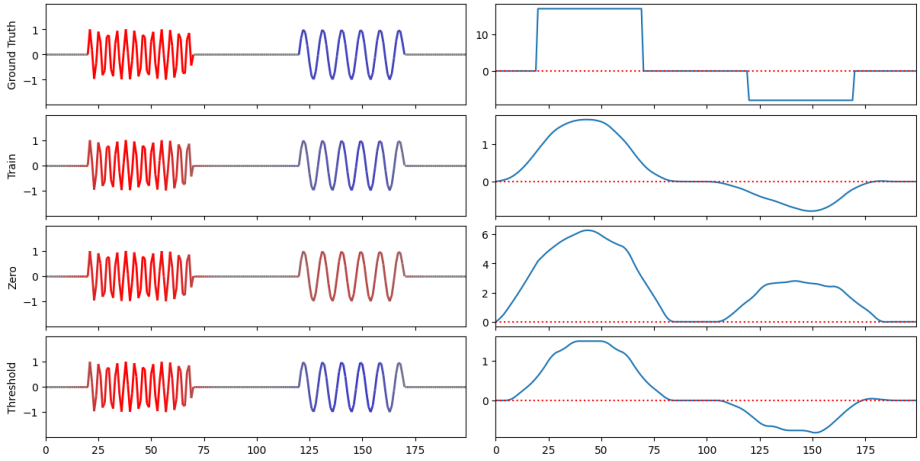


Fig. 3: TSHAP Window attributions on a synthetic time series using various background data. The left panel displays the time series, and the right panel shows the attribution profile. The time series is color-coded: red for positive and blue for negative attributions.

**Classification Attribution Methods.** In this experiment, we evaluate the attribution methods for classification problems using the synthetic dataset. For methods that require background data, we use the train data as the background. To establish a baseline, we included a random attribution method, assigning

values between $-1$ and 1 to each time step. Any effective attribution method should outperform this random attribution.

Table 3: Evaluating classification attribution methods using a synthetic dataset with a hypothetical model and ground truth attributions. Runtime is the total time required to compute attributions for the test set.

| Attribution Methods | Cosine | Precision | Recall | F1 | Runtime (secs) |
|---|---|---|---|---|---|
| TSHAP window | 0.906 | 0.486 | 0.975 | 0.646 | 5.95 |
| TSHAP ROI | 0.900 | 0.952 | 0.780 | **0.851** | **5.95** |
| WindowSHAP Stationary | 0.802 | 0.614 | 0.946 | 0.742 | 533.99 |
| WindowSHAP Sliding | 0.909 | 0.542 | 0.975 | 0.694 | 9.32 |
| WindowSHAP Dynamic | 0.534 | 0.534 | 0.694 | 0.601 | 55.91 |
| LIMESegment | 0.267 | 0.144 | 0.378 | 0.208 | 488.13 |
| LEFTIST | 0.623 | 0.325 | 0.838 | 0.467 | 0.09 |
| Feature Ablation | 0.810 | 0.611 | 0.950 | 0.740 | 0.39 |
| Shapley Value Sampling | 0.801 | 0.614 | 0.946 | 0.742 | 7.86 |
| Random | 0.003 | 0.191 | 0.496 | 0.275 | 0.0 |

Table 3 reveals that TSHAP Window, TSHAP ROI, and Sliding WindowSHAP exhibit the highest cosine similarity to ground truth attributions. TSHAP ROI attributions demonstrate being more conservative, achieving high precision but slightly lower recall. Conversely, TSHAP Window, Sliding WindowSHAP, Feature Ablation, and Shapley Value Sampling show higher recall but lower precision, indicating a tendency to overemphasize relevant areas (Figure 4). The similar performance of TSHAP Window and Sliding WindowSHAP stems from their shared sliding window approach, with differences in the details of calculation methods. LIME-based methods perform poorly on all metrics for this dataset. Regarding running time, TSHAP variants are the fastest among SHAP-based methods. It is important to note that TSHAP calculates both TSHAP Window and ROI together, thus the running time of each is actually the total running time for both variants combined. LEFTIST is the overall fastest but with low scores. Feature Ablation is not only the second fastest method but also performs relatively well.

**Regression Attribution Methods.** For the regression problem, we experiment with two different reference values $r = 0$ and $r = 80$. For $r = 0$, the all-zero sample can be used as the background data because its target value is also 0. For $r = 80$, to ensure $E(f(X_b)) \approx 80$, we generate background samples that have the target values in the range of $[75, 85]$.

Table 4 shows the scores of each attribution method for both $r = 0$ and $r = 80$. The results present a similar pattern to the previous experiment (shown in Table 3) where TSHAP ROI stands out as the most precise method with good scores overall across the metrics, for either type of reference value.

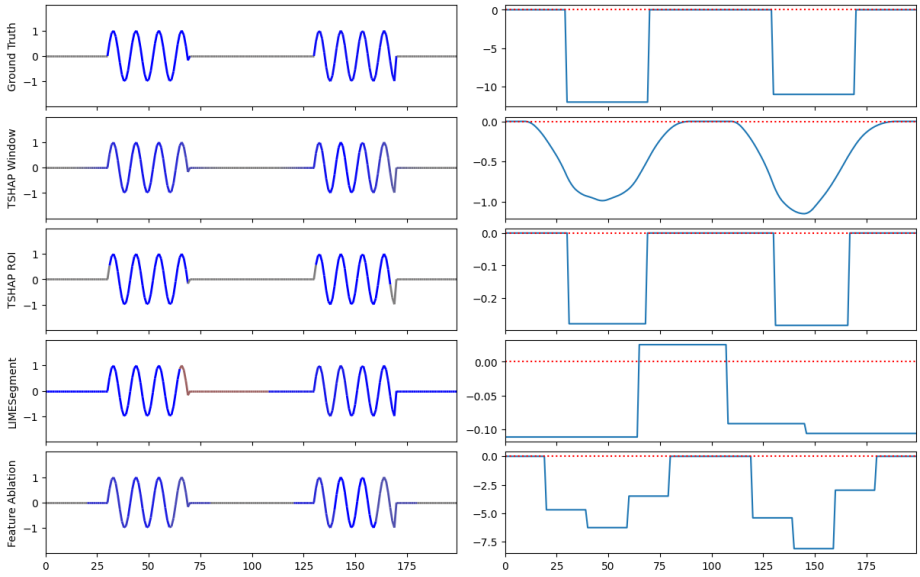


Fig. 4: Synthetic data attributions with different classification attribution methods. The left panel displays the time series, and the right panel shows the attribution profile. The time series is color-coded: red for positive and blue for negative attributions. For this input time series, we have two segments with lower frequency which makes it more likely for the time series to belong to the negative class.

### 4.4 Real Datasets: Faithfulness Analysis

We evaluate the attribution methods using five binary time series classification UCR datasets. These datasets are well-studied in this area and were also studied in [15]. For the classifier we use MiniROCKET [5] due to its efficiency and effectiveness, and for the background we use the training dataset, as we found that the other single-time-series options (zero, centroid, balanced centroid) fail to satisfy the background condition.

Table 5 shows the faithfulness scores (Equation 11) of all the evaluated attribution methods. The average column shows the average faitfulness scores across the datasets with TSHAP Window as the highest. It is interesting that TSHAP Window falters with the Wine dataset while TSHAP ROI struggles with the Chinatown dataset. It should be noted that Chinatown time series are much shorter than the others (only 24 time steps in total). The methods LEFTIST, Feature Ablation, Shapley Value Sampling achieve concerning negative faithfulness scores ($< 0$), each on one occasion. Regarding runtime, on these real datasets, TSHAP Window and TSHAP ROI combined take 7 minutes in total, slightly less than WindowSHAP Sliding. LEFTIST, WindowSHAP Dynamic, and Feature Ablation are the only methods that are faster (only 16 seconds for LEFTIST) but achieve lower faithfulness scores. Stationary WindowSHAP

Table 4: Evaluating regression attribution methods using a synthetic dataset with a hypothetical model and ground truth attributions. The reference values of the attributions are $r = 0$ and $r = 80$.

| Attribution Methods | Cosine | Precision | Recall | F1 | Runtime (secs) |
|---|---|---|---|---|---|
| r = 0 | | | | | |
| TSHAP window | 0.922 | 0.525 | 0.995 | 0.687 | 0.06 |
| TSHAP ROI | 0.885 | 1.000 | 0.750 | **0.851** | **0.06** |
| WindowSHAP Stationary | 0.792 | 0.607 | 0.896 | 0.723 | 10.89 |
| WindowSHAP Sliding | 0.923 | 0.585 | 0.992 | 0.735 | 0.91 |
| WindowSHAP Dynamic | 0.867 | 0.799 | 0.974 | 0.876 | 1.1 |
| Feature Ablation | 0.653 | 0.539 | 0.717 | 0.613 | 0.01 |
| Shapley Value Sampling | 0.769 | 0.604 | 0.892 | 0.720 | 0.25 |
| Random | 0.015 | 0.205 | 0.512 | 0.293 | 0.0 |
| r = 80 | | | | | |
| TSHAP window | 0.941 | 0.508 | 0.989 | 0.668 | 0.55 |
| TSHAP ROI | 0.916 | 0.968 | 0.810 | **0.877** | **0.55** |
| WindowSHAP Stationary | 0.840 | 0.639 | 0.979 | 0.771 | 103.08 |
| WindowSHAP Sliding | 0.944 | 0.568 | 0.990 | 0.719 | 3.05 |
| WindowSHAP Dynamic | 0.874 | 0.748 | 0.983 | 0.841 | 19.98 |
| Feature Ablation | 0.844 | 0.648 | 0.988 | 0.780 | 0.11 |
| Shapley Value Sampling | 0.841 | 0.636 | 0.975 | 0.767 | 2.09 |
| Random | -0.019 | 0.192 | 0.496 | 0.276 | 0.0 |

Table 5: Faithfulness of classification attribution methods on UCR TSC datasets.

| Attribution Methods | Coffee | Wine | BirdChicken | ECG200 | Chinatown | Average |
|---|---|---|---|---|---|---|
| TSHAP Window | 0.424 | 0.059 | 0.155 | 0.195 | 0.230 | **0.213** |
| TSHAP ROI | 0.241 | 0.338 | 0.146 | 0.180 | 0.008 | 0.183 |
| WindowSHAP Stationary | 0.034 | 0.206 | 0.106 | 0.209 | 0.210 | 0.153 |
| WindowSHAP Sliding | 0.412 | 0.068 | 0.157 | 0.159 | 0.174 | 0.194 |
| WindowSHAP Dynamic | 0.156 | 0.355 | 0.085 | 0.067 | 0.215 | 0.176 |
| LIMESegment | 0.262 | 0.197 | 0.110 | 0.123 | 0.288 | 0.196 |
| LEFTIST | 0.184 | -0.012 | 0.055 | 0.190 | 0.131 | 0.110 |
| Feature Ablation | -0.009 | 0.107 | 0.109 | 0.155 | 0.254 | 0.123 |
| Shapley Value Sampling | 0.028 | -0.037 | 0.118 | 0.114 | 0.234 | 0.091 |
| Random | -0.000 | 0.154 | 0.000 | -0.003 | -0.013 | 0.027 |

is again the slowest method with more than 5 hours of runtime. All methods surpass the faithfulness of random attributions on average.

## 5 Conclusion

This paper presents TSHAP, a novel SHAP-based attribution method designed for interpreting black-box time series classification and regression models. TSHAP has two distinct variants: TSHAP Window, which calculates time step attributions based on sliding window Shapley values, and TSHAP ROI, which leverages sliding window attributions to identify important regions within the time series. We also address the critical issue of background data selection, providing a theoretical and experimental analysis leading to a proposed strategy applicable to various time series attribution methods that require background data. To our best knowledge, our paper is the first to study attribution methods for time series regression problems. For evaluation, we tested our proposed methods on both synthetic and real-world datasets in comparison with other state-of-the-art attribution methods. Evaluation on both synthetic and real-world datasets showcases the robustness of our approach, with TSHAP ROI achieving high precision on synthetic data. On real datasets, TSHAP Window demonstrates high overall faithfulness, while TSHAP ROI exhibits less consistent but still strong performance on various datasets. Both TSHAP variants achieve good trade-offs regarding computation runtime versus the usefulness of attributions (as measured using ground truth attributions and faithfulness scores). We acknowledge that the current scope of regression XAI evaluation is limited to synthetic data and we do not address the topic of XAI actionability in this paper. These limitations highlight important avenues for future research, including the development of robust regression XAI evaluation methodologies for real-world data and the exploration of actionability for XAI in the domain of time series classification and regression.

**Acknowledgments.** This publication has emanated from research conducted with the financial support of Taighde Éireann – Research Ireland under Grant [Insight Centre for Data Analytics 12/RC/2289_P2]. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

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

## References

1. J. Bento, P. Saleiro, A. F. Cruz, M. A. Figueiredo, and P. Bizarro. Timeshap: Explaining recurrent models through sequence perturbations. KDD, Aug. 2021.
2. P. Boniol, M. Meftah, E. Remy, and T. Palpanas. dcam: Dimension-wise class activation map for explaining multivariate data series classification. ICMD, 2022.

3. H. A. Dau, A. J. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. J. Keogh. The UCR Time Series Archive. *CoRR*, abs/1810.07758, 2018.
4. A. Dempster, F. Petitjean, and G. I. Webb. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020.
5. A. Dempster, D. F. Schmidt, and G. I. Webb. Minirocket: A very fast (almost) deterministic transform for time series classification. SIGKDD, 2021.
6. M. Guillemé, V. Masson, L. Rozé, and A. Termier. Agnostic local explanation for time series classification. ICTAI, 2019.
7. T. Le Nguyen, S. Gsponer, I. Ilie, M. O'Reilly, and G. Ifrim. Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations. DAMI, 2019.
8. S. Letzgus, P. Wagner, J. Lederer, W. Samek, K.-R. Müller, and G. Montavon. Toward explainable artificial intelligence for regression models: A methodological perspective. *IEEE Signal Processing Magazine*, 39(4):40–58, 2022.
9. S. M. Lundberg and S.-I. Lee. A Unified Approach to Interpreting Model Predictions. NIPS, 2017.
10. 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 (Apr 2024).
11. C. Molnar. *Interpreting Machine Learning Models With SHAP*. 2023.
12. C. Molnar. *Interpretable Machine Learning (book)*, 3 edn. (2025), *https://christophm.github.io/interpretable-ml-book/*
13. A. Nayebi, S. Tipirneni, C. K. Reddy, B. Foreman, and V. Subbian. Windowshap: An efficient framework for explaining time-series classifiers based on shapley values. *J. of Biomedical Informatics*, 144(C), Aug. 2023.
14. T. T. Nguyen, T. L. Nguyen, and G. Ifrim. Robust explainer recommendation for time series classification. *Data Mining and Knowledge Discovery*, 2024.
15. M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. SIGKDD, 2016.
16. D. Guijo-Rubio, M. Middlehurst, G. Arcencio, D. Furtado Silva, and A. Bagnall. Unsupervised feature based algorithms for time series extrinsic regression. Data Mining and Knowledge Discovery, 1-45, 2023.
17. D. I. Serramazza, T. L. Nguyen, and G. Ifrim. Improving the evaluation and actionability of explanation methods for multivariate time series classification, ECMLP-KDD, 2024.
18. D. I. Serramazza, T. L. Nguyen, and G. Ifrim. A short tutorial for multivariate time series explanation using tscaptum. *Software Impacts*, 22:100723, 2024.
19. T. Sivill and P. Flach. Limesegment: Meaningful, realistic time series explanations. PMLR, 28–30 Mar 2022.
20. F. Spinnato, R. Guidotti, A. Monreale, M. Nanni, D. Pedreschi, and F. Giannotti. Understanding any time series classifier with a subsequence-based explainer. *ACM Trans. Knowl. Discov. Data*, 18(2), Nov. 2023.
21. C. Tan, C. Bergmeir, F. Petitjean, and G. Webb. Time series extrinsic regression. Data Mining and Knowledge Discovery, 35:1032—1060, 2021.
22. E. Strumbelj and I. Kononenko. An efficient explanation of individual classifications using game theory. *J. Mach. Learn. Res.*, 11:1–18, 2010.
23. A. Theissler, F. Spinnato, U. Schlegel, and R. Guidotti. Explainable ai for time series classification: A review, taxonomy and research directions. *IEEE Access*, 10:100700–100724, 2022.

24. H. Turbé, M. Bjelogrlic, C. Lovis, and G. Mengaldo. Evaluation of post-hoc interpretability methods in time-series classification. *Nature Machine Intelligence*, 5(3):250–260, Mar 2023.