# *Post-hoc* Explainability for Time Series Classification: Toward a Signal Processing Perspective

Rami Mochaourab[1], Arun Venkitaraman[2], Isak Samsten[3]
Panagiotis Papapetrou[3], Cristian R. Rojas[4]

[1]*Digital Systems Division, RISE Research Institutes of Sweden, Sweden*
[2]*School of Electrical Engineering, EPFL, Lausanne, Switzerland*
[3]*Department of Computer and System Sciences, Stockholm University, Sweden*
[4]*Division of Decision and Control Systems, KTH Royal Institute of Technology, Sweden*

## 1 Motivation

Time series data correspond to observations of phenomena that are recorded over time [1]. Such data is encountered regularly in a wide range of applications such as speech and music recognition, monitoring health and medical diagnosis, financial analysis, motion tracking, and shape identification, to name a few. With such a diversity of applications and the large variations in their characteristics, time series classification is a complex and challenging task. One of the fundamental steps in the design of time series classifiers is that of defining or constructing the *discriminant features* that help differentiate between classes. This is typically achieved by designing novel *representation techniques* [2] that transform the raw time series data to a new data domain where subsequently a classifier is trained on the transformed data, such as 1-nearest neighbors [3] or random forests [4]. In recent time series classification approaches, deep neural network models have been employed which are able to jointly learn a representation of time series and perform classification [5]. In many of these sophisticated approaches, the discriminant features tend to be complicated to analyze and interpret, given the high degree of non-linearity.

Explaining machine learning predictions is an important task in *complex applications* [6], and is steadily gaining interest and relevance, given that in many practical applications the designed system needs to satisfy additional criteria such as safety, reliability, fairness and other ethical aspects besides just accuracy. Complex applications represent situations where the decisions following the machine learning predictions have significant consequences on humans. Examples are diagnosis of diseases, criminal sentencing, facial recognition [7], and autonomous driving [8]. In such complex applications, it is desirable and often necessary to consider building interpretable machine learning models without

loss on performance. Given the large diversity of models and approaches, it is usually not straightforward to determine which models are to be used unless the models or their predictions can be explained.

In general three classes of approaches are used for explaining machine learning predictions [9]. The first class relies on using interpretable models [10], such as sparse linear models or decision trees, that are easier to comprehend by humans. The second class, called model-agnostic explainability, assumes that the machine learning models are *a priori* determined and either these are too complex or it is not permitted to access their details. The third class corresponds to model-dependent explainability approaches where model-specific information, other than just the predicted output, is used for building explanations rather than treating the classifiers as complete black boxes. Some of the popular methods within this third class include Integrated Gradients method [11], DeepLIFT [12], and layer-wise relevance propagation [13]. However, these approaches are typically tailored to deep learning or neural networks setting.

As the classification techniques for time-series data get more and more sophisticated, the underlying representation of the time-series that is built by the technique tends to get less and less understandable. The goal of this paper is to study the use of traditional signal processing features for representation of time-series data for classification in the light of post-hoc explainability. We believe that our work serves as a step towards encouraging a structured dialogue on how the understanding from traditional signal processing could be systematically used in promoting explainability of time-series classification.

In particular, the contributions of this paper are as follows:

- We use well-known and interpretable signal processing techniques for time series representation (Section 2) – the frequency and time-frequency domain transforms, to design features for the classifiers.
- We perform experiments on three diverse datasets that have implications on medical diagnosis, security, and food safety.
- We show that these low complexity transforms provide high classification performance that is comparable and sometimes superior to state-of-the-art algorithms (Section 3).
- We interpret the classifications using *counterfactual explanations* that quantify the

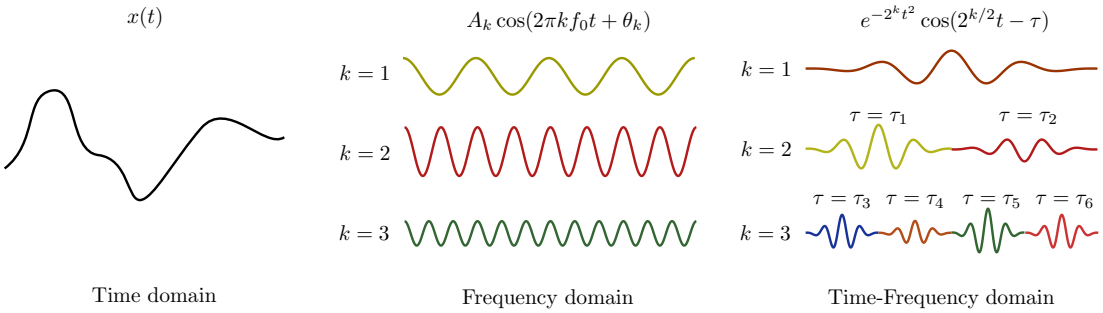



Figure 1: Illustration for a time series features in frequency and time-frequency domains.

necessary changes to an individual data instance such that its classification changes to another class (Section 4), and finally,

- We investigate *feature importance* methods for time series classification and discuss the benefits of using the frequency and time-frequency representations (Section 5). In particular, we perform an empirical study of how the feature importance captures the trade-off between accuracy and explainability.

We wish to emphasize that the focus of this article is not to produce yet another sophisticated technique that outperforms the state-of-the-art approaches; rather our endeavour is directed to bringing together signal processing perspectives and post-hoc explainability.

## 2 Feature Representation

A univariate time series of length $T$ is an ordered sequence of real-value data points $\boldsymbol{x} = (x_0, \ldots, x_{T-1}) \in \mathbb{R}^T$, corresponding to observations of a certain phenomenon. Adjacent observations within a time series are typically dependent, and understanding the nature of this dependence is of importance in order to identify different phenomena when they occur. Time series classification is concerned with identifying such phenomena using supervised learning. The classification accuracy depends on the ability to capture the discriminant features which differentiate between time series classes. These discriminant features are typically concealed within the specifics of the time series temporal correlation. Accordingly, time series feature representations are of paramount importance for the task of learning an accurate classifier.

Let us define the *transform* of a time series of length $T$ to a feature domain $\mathcal{Z}$ as

$$\phi \colon \mathbb{R}^T \to \mathcal{Z}. \tag{1}$$

In this work, we use features obtained from frequency and time-frequency domain transformations [14, 15], namely, the Fourier transform and the Wavelet transform. Our motivation for choosing these features is three-fold: (i) these transforms have been well understood and successfully used for decades, (ii) they are supported by strong physical meaning and intuition, making them naturally suited for our study on explainability, and (iii) they are linear, simple, and efficient to compute – most advanced time-series analysis approaches use these transforms as a building block.

The frequency domain representation of a time series involves its decomposition into a set of sinusoids with different amplitudes and phases. In the time-frequency representation, the time series is decomposed into a basis that is localized in time and frequency. In Figure 1, we illustrate the decomposition of the signal into the bases corresponding to frequency and time-frequency domains. In the frequency domain, we consider the phase and amplitude of the Fourier transform as features for the classifier. In the time-frequency domain, we use the wavelet coefficients as features (they are real-valued since we consider only real-valued wavelets in our work). We next explain these in detail.

### 2.1 Frequency Domain Representation

By assuming that the time series $\boldsymbol{x} = (x_0, \ldots, x_{T-1})$ is periodic, each data point $x_t$, can be expressed using the equation [14]

$$x_t = \sum\nolimits_{k=0}^{T-1} X_k e^{j2\pi kt/T}, \tag{2}$$

where $X_k \in \mathbb{C}, k = 0, \ldots, T-1$, are the Fourier coefficients. In Eq. (2), each data point $x_t$ of the time series is decomposed into $T$ frequency components $e^{j2\pi kt/T}, k = 0, \ldots, T-1$, which are weighted with the Fourier coefficients. Using the discrete Fourier transform (DFT), the time series $\boldsymbol{x}$ is represented through the Fourier coefficients $X_k = \frac{1}{T}\boldsymbol{v}_k^* \boldsymbol{x}, k = 0, \ldots, T-1$, where $\boldsymbol{v}_k := [1, e^{j2\pi k(1)/T}, e^{j2\pi k(2)/T}, \ldots, e^{j2\pi k(T-1)/T}]$. The set of vectors $\{\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{T-1}\}$ constitutes an orthogonal basis for the $T$-dimensional complex vector space. Accordingly, each Fourier coefficient can be considered as an independent

description of a sub-component of the whole time series. This transformation of the time series to the frequency domain can be done efficiently using the Fast Fourier Transform (FFT) algorithm, and is often powerful enough to capture the time series discriminant features for accurate classification, as we will see later. Observe that we can transform from the frequency domain back to the time domain using the inverse DFT, as in Eq. (2).

It is typically sufficient to use the first $L \leq T$ Fourier coefficients as features since the higher frequency components usually represent observation noise. Since the coefficients are complex valued, we need to extract their real-valued representation. From each coefficient $X_k$, we have the alternatives of using the phase, i.e., $z_k = \arg(X_k)$, or the amplitude, i.e., $z_k = |X_k|$, or both quantities. Consequently, we generate a features vector $\boldsymbol{z} = [z_1, \ldots, z_L]$ where the dimension $L$ depends on the choice of features. Let $\phi_L^{\text{DFT}}: \mathbb{R}^T \to \mathbb{R}^L$ be the transform of the time series to the frequency domain.

### 2.2 Time-Frequency Domain Representation

Features designed purely on the basis of the Fourier transform tend to lose important time-localization information. This is because, by definition, the Fourier transform does not retain any temporal information. In many time series applications, it may be important to retain both temporal and frequency domain information in designing features. This is particularly so in the case of non-stationary signals where the frequency components vary with time, as in music or speech. This motivates the use of time-frequency representations (TFRs) that express a time series $\boldsymbol{x}$ in terms of a set of basis functions that span the two-dimensional time-frequency space. The $t$-th data point of $\boldsymbol{x}$ when expressed in a time-frequency basis $\{\psi_{\tau_i,s_j}\}_{i,j}$ is written as [15]

$$x_t = \sum_{i=1}^{T} \sum_{j=1}^{J} X_{i,j}\, \psi_{\tau_i,s_j}(t), \tag{3}$$

where $\psi_{\tau_i,s_j}(t)$ is a function of $t$ localized in time around $\tau_i$ and with dominant frequencies centered around $\omega = \omega_{s_i}$, and $X_{i,j} \in \mathbb{C}$ is the TFR coefficient given by the convolution of $x_t$ with $\psi^*_{\tau_i,s_j}(t)$ [15]:

$$X_{i,j} = x_t \star \psi^*_{\tau_i,s_j}(t) = \sum_{t=0}^{T-1} x_t\, \psi^*_{\tau_i,s_j}(-t), \tag{4}$$

where $*$ denotes the complex conjugate, and $\star$ denotes the convolution in time. Intuitively, the coefficient $X_{i,j}$ indicates the strength of the component of $\boldsymbol{x}$ lying in the $j$-th frequency band at time $\tau_i$. Thus, TFR generates a description of the signal in terms of its frequency components as a function of time. Popular TFRs include the Gabor transform, Wigner-Ville transforms, Hilbert-Huang transforms, and wavelet transform.

In this paper, we select the TFR to be the wavelet transform (WT) which uses a time-frequency basis $\psi_{\tau_i,s_j}(t)$ formed by scaling (compression and dilation) and temporal translations of a function $\psi(t)$ known as the "mother-wavelet" [15]. The wavelet transform basis is then given by $\psi_{\tau_i,s_j}(t) = 2^{-s_j/2}\psi(2^{s_j}\,t - \tau_i)$. The basis is evaluated not at different frequencies, but at different scales $j = 1, \cdots, J$ – each scale corresponds to a frequency band. The WT coefficients at smaller scales represent coarse information or approximation of the signal, whereas those at higher scales represent increasing levels of details. Thus, the feature $X_{i,j}$ represents the information in $\boldsymbol{x}$ as a two-dimensional feature. This corresponds to what is known as the continuous wavelet transform, as the WT coefficients are computed over all times and scales. This is akin to computing the discrete-time FT (DTFT) for a discrete time-signal.

Similar to the DTFT, the CWT is also a redundant representation of the signal. However, if we consider the class of wavelets that generate an orthogonal basis and form a multi-resolution [15], one arrives at what is known as the discrete wavelet transform (DWT). The DWT is not computed at all the scales as with the CWT but at discrete scales in powers of 2 such that $s_j = 2^j$. Furthermore, at each scale, a down-sampling by 2 is made in the time-domain to reduce redundancy in time. As a consequence of considering all the redundancies, the DWT of $\boldsymbol{x}$ has the same length as that of the signal, and takes the form [15] $\boldsymbol{X}^{\mathrm{DWT}} := [a_J, d_1, d_2, \ldots, d_J] \in \mathbb{R}^T$, where $a_J$ contains the $J$-th level approximation coefficients of the signal that give a smooth approximation of the signal, and $d_j$ represents the higher-order information or detail at the $j$-th level. Depending on the level $J$ used, the DWT reveals the information in the signal at different resolutions. Nevertheless, the DWT always has the same dimension as that of the time signal. Just like the DFT, the DWT can also be computed efficiently.

Analogous to the frequency domain representation, we could limit ourselves to using only the first $L$ coefficients of $\boldsymbol{X}^{\mathrm{DWT}}$. Let $\phi_L^{\mathrm{DWT}} \colon \mathbb{R}^T \to \mathbb{R}^L$ be the mapping that transforms a time series to the time-frequency domain using the DWT. In the case of using

the CWT as the feature transform, we use a feature matrix $\boldsymbol{Z} \in \mathbb{R}^{T \times J}$ based on the CWT coefficients $X_{i,s}$. Let $\phi^{\text{CWT}} \colon \mathbb{R}^T \to \mathbb{R}^{T \times J}$ be the transform of the time series to the time-frequency domain representation using the continuous wavelet transform.

## 3 Time Series Classification

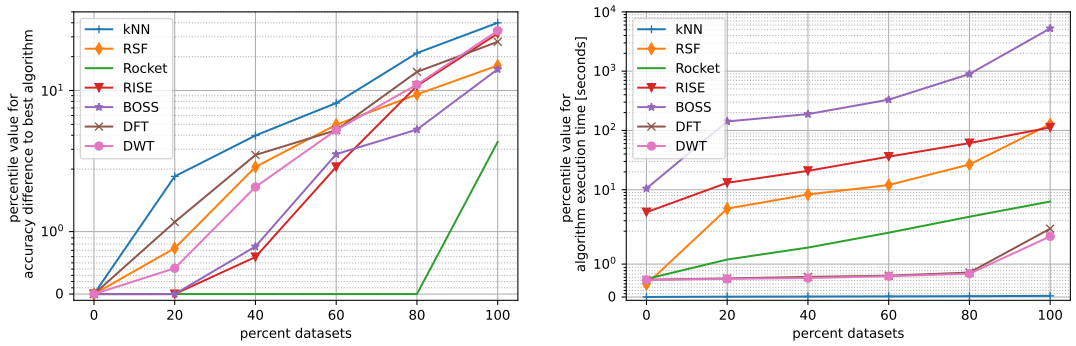
Define a univariate time series dataset $\{(\boldsymbol{x}_n, y_n) \colon n \in \{1, \ldots, N\}\}$ as a collection of $N$ data tuples, where each tuple $(\boldsymbol{x}_n, y_n)$ consists of the time series $\boldsymbol{x}_n \in \mathbb{R}^T$ and its associated class label $y_n \in \mathcal{Y}$. We will consider several time series datasets from the UCR time series repository [16] to compare the performance of different time series classifiers to classifiers which use the frequency and time-frequency domains as features.

For the comparison, we choose five state-of-the-art algorithms (kNN, RSF, ROCKET, RISE, BOSS) that belong to three main approaches for time series classifiers, namely, nearest neighbor-based, shapelet-based, and transformation-based. The reader is referred to [2] for a complete overview of approaches for time series classification.

**k-Nearest Neighbor (kNN)**: The kNN classifier belongs to a baseline family of classifiers that strongly depend on a pair-wise time series distance measure [17], such as the Euclidean distance, or more elastic measures such as dynamic time warping (DTW). We use a 1-NN classifier with Euclidean distance.

**Random Shapelet Forest (RSF)**: The RSF classifier [18] belongs to the class of *shapelet-based approaches* for time series classification. Shapelets are time series subsequences which are generated to have strong similarity with specific classes of time series [19]. The classifier efficiency then relies on how well the shapelets are constructed. The RSF classifier generates the shapelets with different lengths randomly from the time series dataset and constructs a dedicated random forest classifier. As we will see later, the classifications by the shapelet-based classifiers are explainable by relating to the relevant shapelets that the time series share high similarities with.

**BOSS, ROCKET, RISE**: These three classifiers belong to the class of *transformation-based approaches* for time series classification. In this class of approaches, several types of transformations exist: *Dictionary-based approaches* map patterns in a time series to symbolic representations and subsequently build a time series representation composed of sequences of symbols. This technique is useful for time series similarity search through




(a) Accuracy. (b) Complexity.

Figure 2: Comparison of classification performance and complexity of the selected classifiers over 49 datasets from the UCR time series repository [16].

string matching and is utilized in the BOSS classifier [3]. A recent line of research in transformation-based time series classification employs *convolution kernels*, which are used to transform the time series to a very large dimensional space in which efficient separation of the class is possible. The best performing classifier both in terms of training time as well as classification accuracy is ROCKET [20], which generates a large number of random convolution kernels to transform the time series, after which a linear classifier is applied. Other transformation-based approaches correspond to *spectral* transforms where the auto-correlation function and power spectrum are computed on time series intervals as in the RISE classifier [21].

Time series representations in the frequency and time-frequency domains belong to the transformation-based approaches. The DFT or DWT coefficients are used as input features for training a classifier of choice. In order to compare to the classifiers mentioned above, we use the random forest classifier with the DFT and DWT representations. The performance results, given in Figure 2, are obtained using 49 different time series datasets from [16] with sizes ranging between 3 and 1000 samples and time series lengths ranging between 20 and 500 data points. The performance of each classifier on each dataset is averaged over five random splits of the datasets into training and test sets with split ratio 70/30. In addition, for the DFT and DWT approaches, we split the training set into training and validation sets (split ratio of 80/20) for the purpose of finding the best subset of low-frequency coefficients which we use as features for classification. The subsets of low-frequency coefficients correspond to either 10, 25, 50, 75, or 100 percent of all coefficients.

In Figure 2a, we plot the percentile values of the differences in accuracy of each classifier to the most accurate classifier on each dataset. The percentile values illustrate the classifiers' performance over different subsets of all datasets. The ROCKET classifier achieves highest accuracy over at least 80% of the 49 datasets, while RISE and BOSS achieve highest accuracy for about 20% of the datasets. The RISE and BOSS classifiers achieve similar performance over 60% of the datasets, with around 3% and 2% largest difference in accuracy to the best classifier, respectively. However, RISE does not perform as well over all datasets compared to the BOSS classifier. Using the DFT and DWT coefficients with the random forest classifier is shown to perform reasonably well (giving comparable performance to the state-of-the-art on many datasets) despite the simplicity of the feature representation.

The complexity of the algorithms is illustrated in Figure 2b. Here, the execution time of each algorithm includes feature transformation, classifier training (and validation over different subsets of DFT and DWT coefficients), and inference over the test sets. Clearly kNN has the lowest complexity, and is followed by the DFT and DWT approaches.

Motivated by the above observations, we now proceed to consider the DFT and DWT features in more detail on three datasets from [16] with diverse characteristics and related to complex applications. These datasets are:

- **GunPoint (GP)**: Includes two classes of time series where time series of class 0 track the hand movement of a person that draws a gun and class 1 tracks a person's hand movement but with pointing the index finger instead of a gun. The dataset has $N = 200$ time series, each of length $T = 150$. The application of this dataset is complex due to the possibility of identifying a dangerous situation where a person draws a gun.

- **ECG200**: Includes two classes of electrocardiogram (ECG) signals corresponding to normal heartbeat and myocardial infarction. This dataset has $N = 200$ time series, each of length $T = 96$. Misclassification in this dataset could lead to bad consequences for a person through the diagnosis of a heart disease.

- **Beef**: Includes five classes of beef spectrograms, where one class represents pure beef and the other four classes are for beef containing four different adulterants. The dataset includes only $N = 60$ time series where each is of length $T = 470$.

| | | | | | | | $\phi_L^{\text{DFT}}$ | | | | | $\phi_L^{\text{DWT}}$ | | | | $\phi^{\text{CWT}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | kNN | RSF | ROCKET | RISE | BOSS | | DecTree | | RndFrst | | | DecTree | | RndFrst | | CNN |
| | % | % | % | % | % | | % | $L^*$ | % | $L^*$ | | % | $L^*$ | % | $L^*$ | % |
| GP | 89.0 | 99.0 | **100** | 98.7 | **100** | $\theta$: | 78.0 | *7* | 88.7 | *12* | *H.*: | 95.0 | *19* | 99.7 | *76* | 98.7 |
| | | | | | | $A$: | 89.3 | *6* | **97.7** | ***10*** | *D.*: | 94.3 | *60* | **100** | ***90*** | |
| | | | | | | $\theta, A$: | 89.7 | *14* | 97.7 | *38* | | | | | | |
| ECG | 87.7 | 85.7 | **89.0** | **89.0** | 87 | $\theta$: | 82.0 | *13* | 88.7 | *16* | *H.*: | 82.3 | *49* | 87.3 | *58* | 89.3 |
| | | | | | | $A$: | 86.7 | *11* | **90.7** | ***12*** | *D.*: | 82.0 | *81* | **91.0** | ***30*** | |
| | | | | | | $\theta, A$ | 84.7 | *12* | 90.7 | *14* | | | | | | |
| Beef | 46.7 | 58.8 | **83.3** | 74.4 | 71 | $\theta$: | 62.2 | *229* | 80.0 | *158* | *H.*: | 65.6 | *256* | 86.7 | *429* | 74.4 |
| | | | | | | $A$: | 70.0 | *92* | 80.0 | *50* | *D.*: | 74.4 | *192* | **87.8** | ***461*** | |
| | | | | | | $\theta, A$ | 70.0 | *124* | **83.3** | ***148*** | | | | | | |

Table 1: Average classification accuracy on the three datasets. For the frequency-domain representation ($\phi_L^{\text{DFT}}$), the rows correspond to using $\theta$: angle, $A$: amplitude, or both quantities from the DFT coefficients. For the time-frequency domain representation, the rows under $\phi_L^{\text{DWT}}$ correspond to applying $H$: Haar Wavelet, $D$: Daubechies wavelet, and under $\phi_L^{\text{CWT}}$ correspond to Morlet wavelet. The columns $L^*$ give the number of features from the low frequency coefficients that maximize the accuracy on the test data.

The classification in this dataset determines if the beef is safe to eat or not.

We will use the GunPoint and ECG datasets for illustrating the different explanation methods later. The Beef dataset, which has a small number of time series in each class, will only be used to further highlight the suitability of the representations in the frequency and time-frequency domains for time series classification.

In Table 1, we provide average accuracy results for diverse classifiers on the three datasets. As before, the averaging is done over five random splits of the datasets into training and test sets with split ratio 70/30. In $\phi_L^{\text{DFT}}$ and $\phi_L^{\text{DWT}}$, we highlight the power of the representation and the required number of features $L$ by finding the minimum number of features $L^*$ which maximizes the average accuracy on the test data. It is interesting to observe that generally only a few DFT and DWT coefficients relative to the lengths of the time series are needed to provide an accuracy comparable with the state-of-the-art algorithms. For the GP and ECG datasets, the amplitude information alone in the frequency domain representation leads to highest accuracy, while for the Beef dataset, superior performance is achieved with both angle and amplitude quantities. The DWT representation achieves high performance with random forests on all three datasets. The CWT representation achieves higher performance than DWT with decision trees, but generally lower performance than the random forest with DFT and DWT representations.

Comparing DFT to DWT, we observe that DFT generally requires less number of features $L$, however DWT gives higher average accuracy. This shows that DWT provides

a representation with higher capacity than DFT in order to capture the discriminating features more accurately. For CWT, the feature representation is two-dimensional for which we train different convolutional neural networks (CNN) with a few (up to three) hidden layers to perform classification.

## 4 Counterfactual Explanations

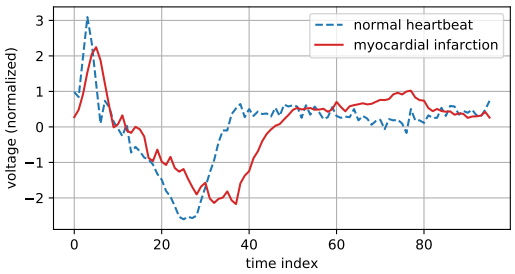
Explainability in time domain has traditionally been offered by interpretable sub-sequences, e.g, shapelets. However, recently several approaches for explaining the decisions by a time series classifier has been proposed using *counterfactual reasoning*, i.e., what perturbations of the input time series must be performed for a classifier in order to change its prediction [22]. Formally, given a classifier $h\colon \mathcal{Z} \to \mathcal{Y}$, a data instance $\boldsymbol{z} \in \mathcal{Z}$ with classification $y = h(\boldsymbol{z})$, and a desired target class $y' \in \mathcal{Y} \setminus \{y\}$, a counterfactual explanation solves the following problem:

$$\underset{\boldsymbol{z}' \in \mathcal{Z}}{\text{minimize}} \quad d(\boldsymbol{z}', \boldsymbol{z}) \quad \text{s.t.} \quad h(\boldsymbol{z}') = y', \tag{5}$$

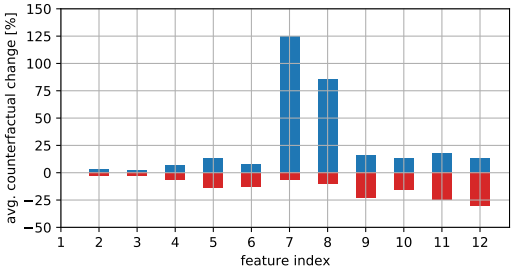
where $d(\boldsymbol{z}', \boldsymbol{z})$ is any distance metric, e.g., Euclidean distance $d(\boldsymbol{z}', \boldsymbol{z}) = ||\boldsymbol{z} - \boldsymbol{z}'||_2$.

Counterfactual explanations (CEs) belong to the class of *post hoc* explainability methods [9] since they require the classifier $h$ to be given. In addition, they provide local explanations according to a given instance $\boldsymbol{z}$. In the frequency and time-frequency domain representations, we are able to utilize approaches to calculate CEs designed for tabular data. One popular model-agnostic algorithm for finding CEs is the growing spheres algorithm [23]. The algorithm relies on searching for CEs by sampling points uniformly at random within a sphere. The sphere's radius is updated iteratively until the nearest point to the instance is found with a different class.

– **Example 1 (ECG):** We plot two example time series from the ECG dataset in Figure 3a. Using a random forest classifier trained on the frequency domain representation of the signals with 12 features (see Table 1), we plot in Figure 3b the average counterfactual changes in the features from all time series of class myocardial infarction to time series with normal heartbeat. Here, we measure the *counterfactual change* from feature vector $\boldsymbol{z}_n$ to the counterfactual feature vector $\boldsymbol{z}'_n$ per feature as $\delta_{n,i} := (z'_{n,i} - z_{n,i})/z_{n,i}$.




(a) Two example ECG time series signals.



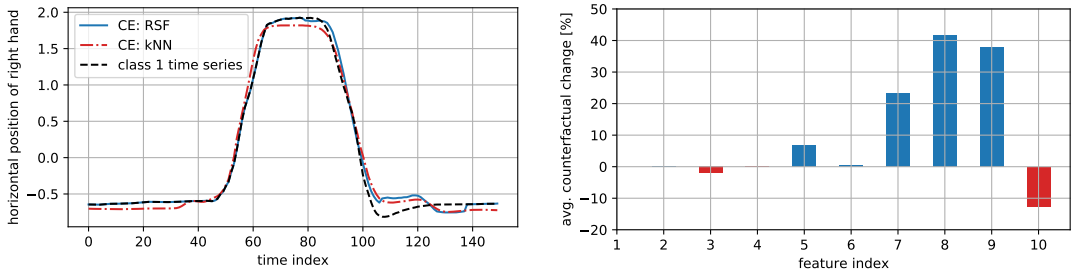

(b) Average changes to the features of ECG signals needed to alter the classifications from myocardial infarction to normal heart beat.

Figure 3: Counterfactual explanations in the frequency domain.

The plot shows that on average, besides small changes in several features, increasing the amplitudes of DFT coefficients 7 and 8 to a relatively large extent is necessary to alter the class of the time series from myocardial infarction to normal heart beats. In other words, the classifier identifies the class of myocardial infarction to be deficient on features 7 and 8, that correspond to two frequency components of the signals. –

There are only a few approaches for finding CEs for time series in the time domain. In [18], two approaches have been proposed for explaining time series classification using nearest neighbour classifiers and forests of shapelet trees. For a trained nearest neighbour classifier, counterfactual explanations correspond to the nearest neighbor to the untransformed time series instance $\boldsymbol{x}$ but of desired label $y'$. For forests of shapelet trees, where each tree recursively partitions time series based on their distance to a shapelet, we can construct a valid counterfactual by fulfilling all tests of a desired prediction path.

– **Example 2 (GunPoint):** In Figure 4, we compare CEs for a GunPoint time series instance of class 1, recording a person pointing the index finger. In Figure 4a, we plot the CEs related to kNN and RSF classifiers [18]. Here, the kNN CE is the nearest neighbor to the instance that has class 0. It can be observed that the main differences in shape of the time series are at the time index intervals $20-50$ and $100-130$ corresponding to the recordings where the person's hand moves from the hip to the level of the shoulder and back again. One can see the difference between the two classes, where the time series with a gun drawn (class 0) seems to be more shaky when it is at the level of the hip. This is due to the weight of the gun. The CE with RSF is able to replace only the necessary parts of the time series with the relevant shapelets in order to change the




(a) Shapelet forest and nearest neighbor.

(b) Frequency domain.

Figure 4: Comparison between different CEs for a GunPoint time series instance.

classification. Notice that the relevant shapelet is the pattern that reflects the shaking due to the weight of the gun.

In Figure 4b, we plot the CEs in the frequency domain. Selecting the best performing case in Table 1, we use the random forest classifier with 10 amplitude features in the frequency domain and apply the growing spheres algorithm to find the CEs. Since the growing spheres method generates points at random in the feature space to successively find the CE nearest to the considered data instance, we plot the average over several counterfactual changes in Figure 4b. From the figure, we see that the amplitudes of the coefficients $7-9$ need to increase to a relatively high extent in order to change the classification. As we will see later, these CEs are very close to the classifier decision boundary, and thus have a probability for classification in class 0 close to 0.5. This will illustrate the power of the frequency domain representation (and analogously the time-frequency representation) for capturing subtle differences in time series.

## 5 Feature Importance

Feature importance approaches assign values to the features that represent their relative contribution to the classifier prediction. A high feature importance value for a specific feature means that the feature contributes greatly to the overall classification, while a low importance value means that the feature has small effect on the classification. We will next describe two approaches for feature importance. One approach is applied to individual predictions to provide local explanations, while the other one explains the classifier model pertaining to global explanations.

### 5.1 Local Explanation using Shapley Values

Consider the individual data instance $\boldsymbol{z}$, and let $p(\boldsymbol{z}) \in [0,1]$ be the probability for its classification $h(\boldsymbol{z})$. Recall that after the time series transformation $\phi$ in (1), $\boldsymbol{z} = \phi(\boldsymbol{x})$ is in the domain $\mathcal{Z}$. Assume that $\mathcal{Z}$ has dimension $L$ as for $\phi_L^{\mathrm{DFT}}$ and $\phi_L^{\mathrm{DWT}}$, and let $\mathcal{L} = \{1, \ldots, L\}$ be the set of all feature indices.

Local feature attribution methods quantify the relative contributions of each feature of the individual data instance $\boldsymbol{z}$ to the prediction $p(\boldsymbol{z})$. It is shown in [24] that the Shapley value from coalitional game theory is a suitable solution for feature attribution for linear predictors. An extension to general nonlinear and individual predictions is done in [25]. The approach for using Shapley values represents the features as players in a cooperative game, and the Shapley values correspond to dividing the worth of the grand coalition (when all players cooperate) among the players. As we will see later, the Shapley value guarantees four desirable properties that are suitable for feature attribution.

To define the Shapley value, we need to determine the value of a coalition of players $\mathcal{S} \subseteq \mathcal{L}$. That is, we need to define the prediction $p(\boldsymbol{z})$ when only specific features $\mathcal{S} \subseteq \mathcal{L}$ contribute to the prediction, while the remaining features $\mathcal{L} \setminus \mathcal{S}$ do not. In [25], this is done by marginalizing over the contributing features $\mathcal{S}$:

$$p_{\mathcal{S}}(\boldsymbol{z}) := \mathbb{E}[p(\boldsymbol{Z}) \mid Z_i = z_i, i \in \mathcal{S}]. \tag{6}$$

We can now define the value of a coalition of features $\mathcal{S}$ for an individual instance $\boldsymbol{z}$ as

$$\nu(\mathcal{S}, \boldsymbol{z}) := p_{\mathcal{S}}(\boldsymbol{z}) - p_{\{\}}(\boldsymbol{z}), \tag{7}$$

where $p_{\{\}}(\boldsymbol{z}) = \mathbb{E}[p(\boldsymbol{Z})]$ corresponds to missing contributions from all features. The Shapley value of feature $i$ is given by

$$\varphi_i(\boldsymbol{z}) = \frac{1}{L} \sum\nolimits_{\mathcal{S} \subseteq \mathcal{L} \setminus \{i\}} \binom{L-1}{|\mathcal{S}|}^{-1} \underbrace{(\nu(\mathcal{S} \cup \{i\}, \boldsymbol{z}) - \nu(\mathcal{S}, \boldsymbol{z}))}_{\text{marginal contribution of } i \text{ to } \mathcal{S}}, \tag{8}$$

which is the weighted average of the feature's marginal contributions over all possible coalitions. The Shapley value is a unique quantity that satisfies several desirable properties for the solution of a cooperative game [25].

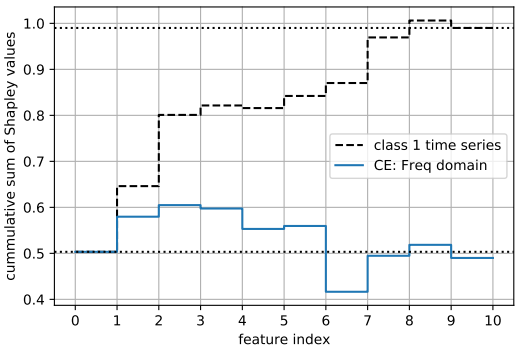
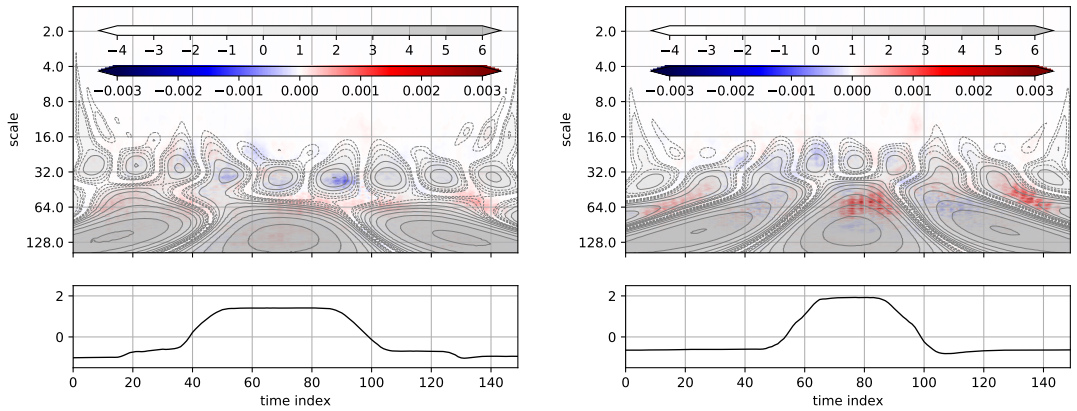



Figure 5: Cumulative sum of Shapley values for 10 features corresponding to the amplitudes of the DFT coefficients. The time series is the same as the one in Figure 4.

The calculation of the Shapley value in (8) requires calculating the valuations in Eq. (7) for each possible subset of features $\mathcal{L}$, which is only computationally feasible for a small number of features. Therefore, low complexity algorithms are needed that provide good approximations for the Shapley values [25, 26]. In [25], a low complexity algorithm is proposed that is based on random sampling, and in [26] an approximation method called kernelSHAP is proposed that relies on the idea from [27] of finding a linear explanation model which approximates the predictor $p$ locally at the given individual instance $\boldsymbol{z}$. In [27], a set of points around $\boldsymbol{z}$ are sampled uniformly at random and used for fitting the explanation model through weighted least squares. The weight for each sampled point corresponds to the proximity of the point from the instance $\boldsymbol{z}$, and is determined using a kernel function. The kernel function which leads to an approximation for the Shapley values is found analytically in [26, Theorem 2].

– **Example 2 (continued):** In Figure 5, we plot the feature importance of the GunPoint time series used in Figure 4 for frequency domain features with a random forest classifier (see Table 1). In Figure 5, the cumulative sum of the features' Shapley values is plotted for the 10 features (amplitude of DFT coefficients), and feature 0 corresponds to missing contributions from all features. The sum of all Shapley values gives the prediction probability for the classification, which is in this case close to 1. It can be seen that features 2, 3, and 8 have contributed the most to the prediction, i.e., their Shapley values are largest, while features 5 and 10 have small negative Shapley values, meaning that their values have slight negative impact on the prediction. Clearly, having such a small number of features is convenient for explainability. Through the transformation




(a) "Gun draw" time series (class 0). (b) "Point" time series (class 1).

Figure 6: Feature representation in the time-frequency domain. The SHAP values [26] represent the feature importance of each pixel in the images.

to the frequency domain, the feature importance translates to the importance of the frequency component in the individual time series. Comparing to the feature importance of the respective counterfactual explanation (CE), shown in Figure 4b, we see that most of the features have smaller Shapley values. In particular, the value of feature 7 in the CE contributes negatively to a great extent to the prediction. The sum of all Shapley values for the CE is close to 0.5, i.e., close to the classifier's decision boundary. This is in accordance with the CE problem definition in (5).

In Figure 6, we use kernelSHAP [26]. We plot the features in the time-frequency domain for the same GunPoint time series, also plotted below the 2D plot. We also compute the Shapley values for each feature value and highlight them in the 2D plot. Even visually, it is evident that the contours generated by CWT for the two classes are distinct. We observe that the Shapley values are higher around the samples in time in the intervals $(10, 30)$ and $(120, 140)$. We note that these intervals correspond to the regions that distinguish the two classes: In class 0, which corresponds to the person having a gun, we observe from Figure 6a that there are two smaller 'bumps' or jumps around $(10, 30)$ and $(120, 140)$. These jumps are on the other hand missing in the same intervals in the time-signal from class 1 as seen in Figure 6b. This might explain why the Shapley values are high in these intervals – since they clearly distinguish the two classes, even visually from the time series. We can also see that the contour of the CWT and the high Shapley areas are distinctly different for the two classes in these intervals. We also observe that around scales $(16, 32)$ (that correspond to the high-frequency or

jump regions corresponding to the transitions), the Shapley values are negative in the intervals $(40, 60)$ and $(80, 100)$, even thought the CWT values are high in these regions. We believe this is due to the nature of the signal in the two classes in these intervals. In both classes, these intervals correspond to similar transitions, and given their similarity, they are perhaps less relevant in the decision making process for the classifier. Thus, we see that the explainability offered by the Shapley values agrees with what one expects from intuition. It is also interesting to see that the Shapley values indicate that the transition interval $(40, 60)$ is less important ('more blue') for class 1, whereas that from $(80, 100)$ is less relevant for class 0. This indicates that the transitions of importance also vary from one class to the other − perhaps indicative of the effect of difference in inertia of carrying a weapon in one case, and not carrying one in the other. −

### 5.2 Global Explanation using Sobol' Indices

The classifier $h\colon \mathcal{Z} \to \mathcal{Y}$ can be written as the composition of two functions, $h(\boldsymbol{z}) = g(f(\boldsymbol{z}))$, where $\boldsymbol{z} \in \mathcal{Z}$, $f\colon \mathcal{Z} \to \mathcal{V}$ and $g\colon \mathcal{V} \to \mathcal{Y}$, with $\mathcal{V}$ being a vector space; this applies to many classification methods, such as thresholded polynomials, support vector machines and deep neural networks [28].

Assume, without loss of generality, that $\mathcal{Z} = [0, 1]^L$, i.e., each feature takes values between 0 and 1. This can be achieved through normalization, such as dividing each feature value by the data vector's Euclidean norm to get $||\boldsymbol{z}|| = 1$. Taking a random features vector $\boldsymbol{Z}$, one can write the *functional ANOVA decomposition* [29] of $f$ as $f(\boldsymbol{Z}) = \sum_{\mathcal{S}\subseteq\mathcal{L}} f_{\mathcal{S}}(\boldsymbol{Z})$, where $\mathcal{S}$ runs over all subsets of the set of features $\mathcal{L} := \{1, \ldots, L\}$, and $f_{\mathcal{S}}$ only depends on the variables contained in $\mathcal{S}$. In case these variables are independent and uniformly distributed in $[0, 1]$, the functions $f_{\mathcal{S}}$ satisfy a nice orthogonality property: $\mathrm{cov}(f_{\mathcal{S}}(\boldsymbol{Z}), f_{\mathcal{Q}}(\boldsymbol{Z})) = 0$ if $\mathcal{S} \neq \mathcal{Q}$. Thus, if one defines $\sigma^2_{\mathcal{S}} := \mathrm{var}(f_{\mathcal{S}}(\boldsymbol{Z}))$, one has that

$$\mathrm{var}(f(\boldsymbol{Z})) = \sum\nolimits_{\mathcal{S}\subseteq\mathcal{L}} \sigma^2_{\mathcal{S}}. \tag{9}$$

In words, the contribution of the variables to the variance of $f(\boldsymbol{Z})$ can be decomposed into the contribution of each subset $\mathcal{S}$ of those variables. In particular, one can define

the *Sobol' indices* [30]

$$\underline{\tau}_{\mathcal{S}}(f) := \sum\nolimits_{\mathcal{Q} \subseteq \mathcal{S}} \sigma^2_{\mathcal{Q}}, \quad \overline{\tau}_{\mathcal{S}}(f) := \sum\nolimits_{\mathcal{Q}:\ \mathcal{Q} \cap \mathcal{S} \neq \emptyset} \sigma^2_{\mathcal{Q}}. \tag{10}$$

The lower index $\underline{\tau}_{\mathcal{S}}$ quantifies the importance of the variables in $\mathcal{S}$ through all main effects and interactions in $\mathcal{S}$, while the upper index $\overline{\tau}_{\mathcal{S}}$ includes any interaction to which one or more of the variables in $\mathcal{S}$ contribute. The functional ANOVA decomposition and related indices can be used to determine which variables are the most relevant for a specific prediction of a classifier $h$, in the sense that changing slightly one of them can affect the output of the classifier.

As is done in the previous section, by appealing to concepts from cooperative game theory, it is possible to arrive in an axiomatic manner at a single index, the Shapley value, that quantifies the contribution of each feature $i$ to the function $f$. Here, the value of a coalition $\mathcal{S}$, similar to the one defined in (7), can be chosen as the lower index in (10), and consequently the Shapley values can be calculated. In this case, it is shown in [30] that the lower and upper Sobol' indices for each singleton variable $i$, $\underline{\tau}_{\{i\}}(f) = \sigma^2_{\{i\}}$ and $\overline{\tau}_{\{i\}}(f) = \sum_{\mathcal{Q}:\ i \in \mathcal{Q}} \sigma^2_{\mathcal{Q}}$, respectively lower and upper bound the Shapley value for $i$.

The Sobol' indices $\underline{\tau}_{\{i\}}(f)$ and $\overline{\tau}_{\{i\}}(f)$ can be regarded as lower complexity alternatives to the Shapley value [30]. High values for $\underline{\tau}_{\{i\}}(f)$ mean that $i$ contributes highly to the variance of $f$, while low values of $\overline{\tau}_{\{i\}}(f)$ mean that $i$ does not contribute much to $f$ since its interaction with any other set of variables does not bring much effects.

– **Example 1 (continued):** In Figure 7, we consider the ECG dataset with feature representation in the frequency domain, where the 12 features correspond to the amplitudes of the DFT coefficients. We can use the summary statistics of the Shapley values from the previous section to provide global explanations about the classifier. It can be seen that feature 7 has highest importance in this case. The global explanation using the Sobol' indices approach does not use the data instances as in the local explanation approach of the previous section, but randomly samples feature values from the unit interval. This is particularly useful in cases where we do not have access to many data instances to perform a sufficient number of local explanations. Contrary to the local approach, it can be observed that the global approach identifies feature 4 to be as important as feature 7. These important features, corresponding to two frequencies in the

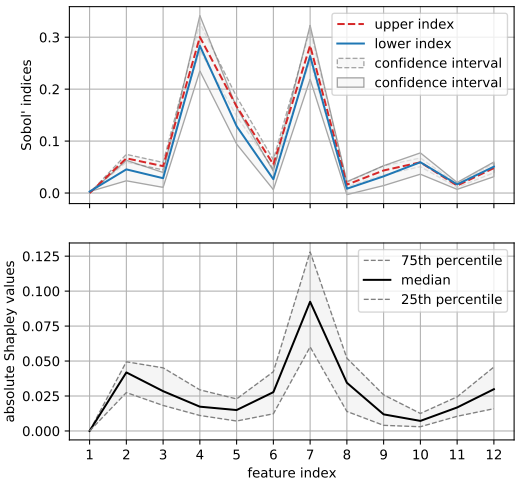
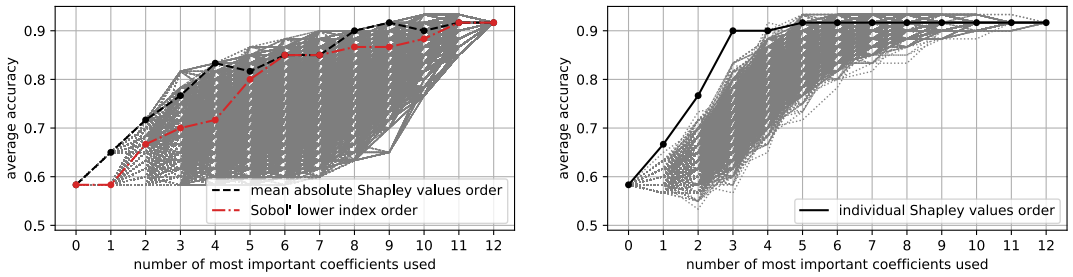



Figure 7: Global feature importance for the ECG dataset using the random forest classifier. The amplitudes of 12 DFT coefficients are used as features.




(a) Using global feature importance.

(b) Using local feature importance.

Figure 8: Effect of using a subset of the most important features in the ECG dataset, according to (a) global feature importance and (b) local feature importance, on the accuracy of the classifier.

time series, contribute greatly to the classifier value. If we remove the contributions from these particular features, then the classifier performance would degrade significantly.

In Figure 8, we illustrate the tradeoff between the classifier accuracy on the test set and the number of most important features used. The important features are ranked according to the Shapley values (global and local feature importance) or Sobol' indices (global feature importance). We compute the classifier accuracy where the input is set to the mean feature values over all the test data except for the subset of features we choose to show the influence of. This subset is obtained by taking the highest $k$ important features, where $k$ is between 1 and $L^* = 12$. We report this in Figure 8a, where we have used the global feature importance values shown in Figure 7. Selecting for example the two most important features means that the two features with the highest feature importance (features 7 and 2 according to Shapley values and 4 and 7 for the

Sobol' indices) are assigned their true values from the test set while the remaining eight features are assigned the same mean feature value. It can be observed that for both feature importance approaches, the accuracy of the classifier increases rapidly with the first few most important features. Generally, we see that the accuracy using Shapley values is higher than that of Sobol' indices, probably due to the fact that the Shapley values are calculated using the test data while the Sobol' indices are agnostic to the actual data. To highlight the importance of the order of the features, we plotted the accuracy for $10^4$ random feature permutations in grey lines. The large accuracy cloud shows that order of the features is significance and that the ordering given by the Shapley values and Sobol' indices give a very good approximation, especially when the information for the classifier comes from only a small number of features.

We repeat the experiment for local feature importance using Shapley values and plot the accuracy in Figure 8b. While the plot in Figure 8a uses the global feature importance which is applied on all test data, in Figure 8b we use for each data sample the feature importance order according to its individual Shapley values. It can be observed that just the first three most important features bring more than 30% accuracy increase. In addition, there is a large gap compared to the best random feature orders which emphasizes the suitability of Shapley values for feature importance. –

The explainability aspects and tradeoffs we reported in the two examples above extend analogously to other datasets. In order to avoid repetition and in the interest of space, we restrict the discussion to these datasets.[1]

## 6 Concluding Remarks

We promote the use of frequency and time-frequency domain representations for explainable time series classification. These representations are efficient in the following aspects: (i) the algorithms applied for the transforms have low complexity, such as the FFT, (ii) we are able to classify time series using simple classifiers for tabular data, and (iii) we can achieve comparable classification accuracy to state-of-the-art time series classifiers. We observe that these transforms as building blocks for classifiers help in their *post-hoc* explainability both for feature importance and counterfactual explainability, and in terms

[1]The code for reproducing the figures in the paper can be found at `https://github.com/rami-mochaourab/TSC-Explainability`.

of strong physical intuition and understanding. Our experiments also indicate that it could often be sufficient to consider only a few features in the frequency/time-frequency domain to achieve surprisingly competitive accuracy results while being intuitive. In particular, our explainability analysis in particular on the GunPoint and ECG datasets highlights the potential and relevance of using traditional signal processing concepts in making classifiers more explainable, especially in critical and complex applications such as health and safety. All these aspects show that the signal-processing perspective could be of great potential and particularly relevant since the lack of explainability is of the most severe roadblocks in the widespread application and acceptance of machine learning approaches in complex applications today.

## References


[1] P. Esling and C. Agon, "Time-series data mining," *ACM Computing Surveys*, vol. 45, no. 1, p. 12, 2012.

[2] A. P. Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall, "The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Min. Knowl. Discov.*, vol. 35, no. 2, pp. 401–449, Mar. 2021.

[3] P. Schäfer, "The BOSS is concerned with time series classification in the presence of noise," *Data Min. Knowl. Discov.*, vol. 29, no. 6, pp. 1505–1530, 2015.

[4] I. Karlsson, P. Papapetrou, and H. Boström, "Generalized random shapelet forests," *Data Min. Knowl. Discov.*, vol. 30, no. 5, pp. 1053–1085, Sep. 2016.

[5] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Min. Knowl. Discov.*, vol. 33, no. 4, pp. 917–963, 2019.

[6] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *CoRR*, vol. abs/1702.08608, 2017.

[7] T. Sixta, J. C. S. Jacques Junior, P. Buch-Cardona, E. Vazquez, and S. Escalera, "Fairface challenge at ECCV 2020: Analyzing bias in face recognition," in *Computer Vision – ECCV 2020 Workshops*, 2020, pp. 463–481.

[8] E. Awad, S. Dsouza, R. Kim, J. Schulz, J. Henrich, A. Shariff *et al.*, "The moral machine experiment," *Nature*, vol. 563, no. 7729, pp. 59–64, 2018.

[9] C. Molnar, "Interpretable machine learning: A guide for making black box models explainable," https://christophm.github.io/interpretable-ml-book/, ver. 2020-03-16.

[10] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong, "Interpretable machine learning: Fundamental principles and 10 grand challenges," *CoRR*, vol. abs/2103.11251, 2021.

[11] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 3319–3328.

[12] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, Aug. 2017, pp. 3145–3153.

[13] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digit. Signal Process.*, vol. 73, pp. 1–15, 2018.

[14] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd ed. Prentice-Hall, 1996.

[15] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*, 3rd ed. Academic Press, 2008.

[16] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, and Hexagon-ML, "The UCR time series classification archive," Oct. 2018, https://www.cs.ucr.edu/∼eamonn/time_series_data_2018/.

[17] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: Experimental comparison of representations and distance measures," *Proc. of VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[18] I. Karlsson, J. Rebane, P. Papapetrou, and A. Gionis, "Locally and globally explainable time series tweaking," *Knowl. Inf. Syst.*, vol. 62, no. 5, pp. 1671–1700, 2019.

[19] L. Ye and E. Keogh, "Time series shapelets: a novel technique that allows accurate, interpretable and fast classification," *Data Min. Knowl. Discov.*, vol. 22, no. 1, pp. 149–182, 2011.

[20] A. Dempster, F. Petitjean, and G. I. Webb, "ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Min. Knowl. Discov.*, vol. 34, no. 5, pp. 1454–1495, 2020.

[21] J. Lines, S. Taylor, and A. Bagnall, "Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles," *ACM Trans. Knowl. Discov. Data*, vol. 12, no. 5, 2018.

[22] C. R. Sandra Wachter, Brent Mittelstadt, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harvard Journal of Law & Technology, forthcoming*, vol. 31, no. 2, 2018.

[23] T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, and M. Detyniecki, "Comparison-based inverse classification for interpretability in machine learning," in *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations*, J. M. et al., Ed., vol. 853, 2018, pp. 100–111.

[24] S. Lipovetsky and M. Conklin, "Analysis of regression in game theory approach," *Appl. Stoch. Models Bus. Ind.*, vol. 17, no. 4, pp. 319–330, 2001.

[25] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowl. Inf. Syst.*, vol. 41, no. 3, pp. 647–665, Dec. 2014.

[26] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017.

[27] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should i trust you?" Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[28] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge Uni. Press, 2014.

[29] G. Hooker, "Discovering ANOVA structures in black box functions," in *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[30] A. B. Owen, "Sobol' indices and Shapley value," *SIAM/ASA J. Uncertainty Quantification*, vol. 2, no. 1, pp. 245–251, Jun. 2014.