Explaining the Predictions of Any Time Series Classifier

Bachelor Thesis

to attain the scientific degree

Bachelor of Science in IT-Systems Engineering

handded in by **Felix Mujkanovic**

Supervisor: Prof. Dr. Tobias Friedrich
Advisors: Vanja Doskoč, Martin Schirneck
Chair: Algorithm Engineering

Potsdam, July 26, 2019

Zusammenfassung

Aktuelle Modelle für maschinelles Lernen, die auf Zeitreihen spezialisiert sind, legen beeindruckende Prognosefähigkeiten an den Tag, bleiben jedoch größtenteils Black Boxes. Gleichzeitig versprechen modellunabhängige, nach dem *Perturbation-Prinzip* arbeitende Erklärsysteme, insbesondere das kürzlich vorgeschlagene SHAP, einzelne Vorhersagen eines beliebigen Modells zu erklären. In dieser These wird demonstriert, wie diese Techniken an Modelle angepasst werden können, die auf Zeitserien arbeiten. Hierfür werden neuartige, sogenannte *mapping functions* vorgestellt, die auf Zeitreihen zugeschnitten sind. Die Aussagekraft sowie die Limitierungen dieser Mappings werden analysiert. Darüber hinaus werden verwenden die vorgeschlagenen Mapping-Techniken in einem groß angelegten Experiment verwendet, um die Entscheidungsfindungsprozesse einer Vielzahl von aktuellen Zeitreihenklassifikatoren zu vergleichen und nach Ähnlichkeiten zwischen scheinbar unterschiedlichen Klassifizierungskonzepten zu suchen.

Explaining the Predictions of Any Time Series Classifier

Felix Mujkanovic [1]

Abstract

State-of-the-art machine learning models specialized on time series display impressive predictive capabilities, yet they mostly remain black boxes. At the same time, model-agnostic, so-called perturbation-based explainers, including recently proposed SHAP, promise to explain individual predictions of any model. In this work, we show how to adapt those techniques to time series predictors. For this, we present novel domain mappings tailored to time series. We analyze the explicative power as well as the limits of our mappings. Additionally, we employ our proposed mapping techniques in a large-scale experiment to compare the decision-making processes of a variety of state-of-the-art time series classifiers and discover similarities between seemingly distinct classification concepts.

1. Introduction

In recent years, AI technology has become ubiquitous in numerous industries. From dynamic pricing and recommender systems to autonomous vehicles and predictive policing, machine learning (ML) models have found their way into our everyday lives. As they are progressively entrusted with more and more responsibility, we should wonder whether we can actually trust them. ML systems must be safe, exhibit predictable and expected behavior, and must not discriminate (Lipton, 2016). Whether an ML system satisfies such fuzzy, yet important criteria cannot be determined by just computing a single “trustworthiness” metric.

To combat this issue, researchers are pushing towards technologies that *explain* the decision-making process of ML models. Through understanding the inner workings and reasons behind the predictions of such complicated estimators, ML engineers are enabled to both spot flaws and build trust in their systems. The movement towards explainable AI (XAI) has been further fueled by a “right to explanation” recently proposed by the European Union (Goodman & Flaxman, 2017).

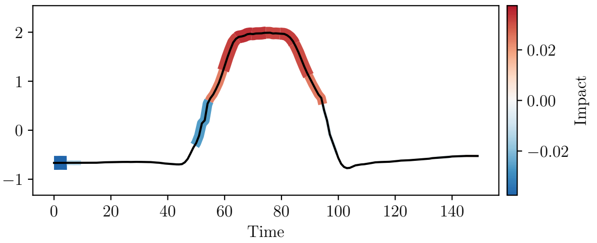

Figure 1. Exemplary explanation depicting which temporal segments of a specimen (time series #76 from the UCR GunPoint test set) are relevant to a residual neural network classifier (ResNet) when asked whether the specimen belongs to its true class. The red areas were identified as supporting the classifier’s decision towards the true class while blue areas oppose it. The explanation was generated using Kernel SHAP with time slice mapping.

However, one’s ability to comprehend the decision-making process of an ML model highly depends on its complexity. While understanding a linear regression is feasible (Lou et al., 2012), grasping the whole of the inner workings of a sufficiently complex neural network with thousands or millions of weights is considered to be impossible at this time (Lipton, 2016). As a possible solution, Alvarez-Melis & Jaakkola (2018b) proposed specialized models which have interpretability built into them. However, explaining other, uninterpretable models post-hoc and at the grand scale still elude us.

Rather than striving to explain an inherently uninterpretable model in its completeness, numerous methods have been proposed that only explain the model’s prediction of an individual input. In what follows, we call that input *specimen*. For example, saliency-based approaches like those of Selvaraju et al. (2017) examine the gradients of neural networks during the prediction of the specimen. However, most of these techniques are highly model-specific. In contrast, we will focus on *model-agnostic*, *perturbation-based* approaches that treat the model as a black box and probe it in the vicinity of the specimen. Exploiting the gained insight, these approaches then infer which features of the specimen are relevant to the model, yielding an *explanation* as the result. Ribeiro et al. (2016) pioneered work in this area with their LIME framework. Multiple extensions were

[1] Hasso Plattner Institute, University of Potsdam, Potsdam, Germany.

subsequently proposed, which have recently been unified by Lundberg & Lee (2017) with their generic SHAP framework and specifically its model-agnostic explanation generator called *Kernel SHAP*.

Perturbation-based approaches require instructions on how to perturb the specimen to yield variants that lie in its vicinity. These instructions are embodied by a *mapping function*. Ribeiro et al. (2016) previously introduced mappings for images, text, and tabular data. In the domain of time series, however, such mappings are still lacking. In this work, as a step towards model-agnostic explanations for predictions on time series, we present two expressive mapping functions for this new domain in Sections 3.1 and 3.2. An exemplary explanation found using one of our mapping functions is exhibited in Figure 1. Additionally, we discuss challenges and pitfalls in designing mapping functions and analyze the limitations of our extensions in Sections 3.3 and 3.4.

In time series classification, a multitude of conceptually different approaches coexists. In Section 4, we first survey the current state-of-the-art models for time series classification. Afterwards, we employ Kernel SHAP combined with our mappings to develop an automated tool which is able to generate a vast amount of explanations for these classifiers on different specimens and then automatically compare these explanations. Our tool in fact reveals surprising similarities between diverse models.

2. Local Explanations and Kernel SHAP

Before we introduce Kernel SHAP, we first need to formalize the black-box ML model we want to interpret. We define the space of all possible inputs for the model as the vector space I with dimension d. The model itself then is the function $f\colon I \to \mathbb{R}$, which takes an input and produces a prediction. In classification, we would employ one such function for each class, which outputs the probability that the input belongs to that class.

For the rest of this paper, we formally denote the specimen as $\boldsymbol{x} \in I$. Intuitively, Kernel SHAP "disables" portions (called *fragments*) of $\boldsymbol{x}$ to yield one of many *perturbed specimens* $\boldsymbol{z} \in I$ and then computes $f(\boldsymbol{z})$. Doing this multiple times with different fragments disabled each time explores the behavior of the prediction function f in the vicinity of the specimen. Using the gained insight, an interpretable linear *explanation model* is built that approximates the original model near the specimen, as illustrated in Figure 2. The interpretable explanation model then allows us to estimate the *impact* that each fragment of $\boldsymbol{x}$ has on the prediction $f(\boldsymbol{x})$ of the model. We will now formalize this process.

First, assume we have a way to divide the specimen $\boldsymbol{x}$ into some number d' of fragments that can be disabled individually. We now introduce the space of so-called *simplified inputs* as the vector space $I' = \{0,1\}^{d'}$ with dimension d'. Let us take a look at any one simplified input $\boldsymbol{z'} \in I'$. Each 1 or 0 in $\boldsymbol{z'}$ hints an activated or deactivated fragment of the specimen $\boldsymbol{x}$, respectively.

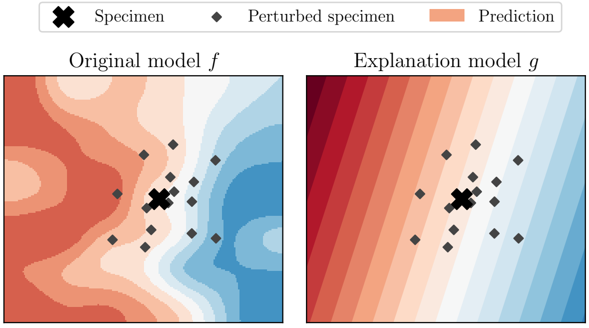

Figure 2. Toy example illustrating the intuition behind Kernel SHAP. The original model f is probed in the vicinity of the specimen. The collected predictions are then used to build a linear regression explanation model g.

The actual act of disabling fragments is performed by a *mapping function* $h_{\boldsymbol{x}}\colon I' \to I$. Upon invocation with any simplified input $\boldsymbol{z'}$, the mapping function disables the fragments of $\boldsymbol{x}$ that are hinted as disabled by $\boldsymbol{z'}$ and yields the resulting perturbed specimen. Take note that for the rest of this paper, symbols with or without a prime are related to simplified or original inputs, respectively.

Let us quickly examine the simplified input $\boldsymbol{x'} = \vec{1} \in I'$. Since $\boldsymbol{x'}$ only contains 1s and thus hints that all fragments are enabled, $\boldsymbol{x'}$ is the simplified representation of the unperturbed specimen $\boldsymbol{x}$. So naturally, $h_{\boldsymbol{x}}(\boldsymbol{x'}) = \boldsymbol{x}$ holds.

With all the groundwork finished, let us now move on to the explanation model.

Definition 1. *An explanation model is a linear regression model $g\colon I' \to \mathbb{R}$ with the* impact vector $\boldsymbol{\phi} \in \mathbb{R}^{d'+1}$ *such that* $\forall \boldsymbol{z'} \in I'$:

$$g(\boldsymbol{z'}) = \phi_0 + \sum_{i=1}^{d'} \phi_i z'_i.$$

The set of all possible explanation models is called G.

The explanation model operates on simplified inputs I' and strives to linearly approximate the behavior of the original model in the close vicinity around $\boldsymbol{x}$. In other words, given a $\boldsymbol{z'} \in I'$ with only a couple of fragments disabled, i.e., only a couple of zeros, $g(\boldsymbol{z'}) \approx f(h_{\boldsymbol{x}}(\boldsymbol{z'}))$ should roughly hold.

The impact vector $\boldsymbol{\phi}$ (also known as SHAP values) now acts as the explanation of the model's decision-making process with respect to $\boldsymbol{x}$. For each $i \in \{1, \ldots, d'\}$, $\boldsymbol{\phi}_i$ expresses which impact activating the corresponding fragment has on the prediction of the model.

In practice, an impact vector $\boldsymbol{\phi}$ is found by probing the model f with a lot of randomly generated perturbed specimens $h_{\boldsymbol{x}}(\boldsymbol{z'})$, $\boldsymbol{z'} \in I'$ and using the results to build an explanation model that approximates the ideal explanation model

$$\xi = \arg\min_{g \in G} \sum_{\boldsymbol{z'} \in I'} \left[f(h_{\boldsymbol{x}}(\boldsymbol{z'})) - g(\boldsymbol{z'})\right]^2 \pi_{\boldsymbol{x'}}(\boldsymbol{z'}).$$

The function $\pi_{\boldsymbol{x'}} \colon I' \to \mathbb{R}^+$ is a distance function such that $\pi_{\boldsymbol{x'}}(\boldsymbol{z'})$ is a distance measure between $\boldsymbol{x'}$ and $\boldsymbol{z'}$. This distance measure reduces the influence of those simplified inputs which have most of their fragments disabled, since the original model's predictions of those are prone to noise. For details regarding $\pi_{\boldsymbol{x'}}$, see Lundberg & Lee (2017).

3. Kernel SHAP on Time Series

Kernel SHAP provides a framework to generate explanations for the behavior of a model. However, the interpretability of these explanations highly depends on the choice of a mapping function $h_{\boldsymbol{x}}$, a choice that is left up to the user. The mapping function solely dictates how the specimen is divided into fragments. Those fragments need to be intuitively understandable by the user so he comprehends what it means that a fragment has high or low impact. Additionally, as shown in Section 3.3, employing a mapping function not suitable for the data or model could produce explanations that are misleading. All this renders the choice of $h_{\boldsymbol{x}}$ an important challenge.

Ribeiro et al. (2016) have previously introduced mappings for images, text, and tabular data. We now propose expressive SHAP-style mapping functions for the new domain of time series.

For the rest of this paper, since we focus on time series now, we define the vector space $I = \mathbb{R}^d$ and view it as space of time series with length d. Each vector $\boldsymbol{z} \in I$ thus is a time series, whose value at time $t \in \{1, \ldots, d\}$ is naturally denoted by $\boldsymbol{z}_t$.

3.1. Time Slice Mapping

We first present *time slice mapping*. This technique splits a specimen $\boldsymbol{x}$ into d' equi-length slices along the time axis. Each slice $i \in \{1, \ldots, d'\}$ makes up one fragment whose activation is thus governed by $\boldsymbol{z}'_i$. To disable an individual fragment, i.e., slice, one cannot just remove it or fill it with missing values since most models can neither cope with time series of dynamic length nor with missing values. Instead, the slice is replaced with the corresponding slice from a second replacement time series $\boldsymbol{r} \in I$. Five such replacement series will be presented in the next section.

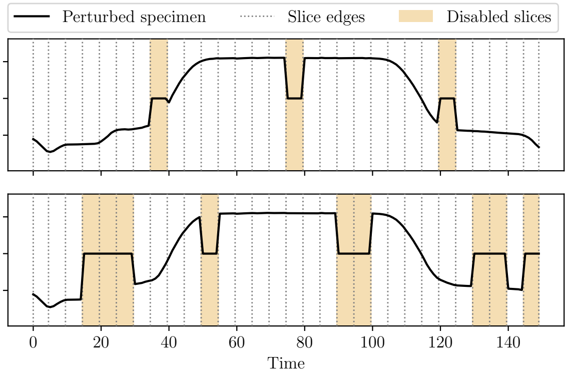

Figure 3. Two different perturbations of the same specimen (time series #7 from the UCR GunPoint test set), yielded by time slice mapping with global mean replacement (see Section 3.1.1). A different set of fragments has been disabled in both plots.

To bring it all together, when the time slice mapping function, which is tailored to the specimen $\boldsymbol{x}$, is queried with any simplified input $\boldsymbol{z'}$, it disables all the slices i of $\boldsymbol{x}$ whose $\boldsymbol{z}'_i$ is 0 and yields the resulting perturbed version of $\boldsymbol{x}$, as demonstrated in Figure 3.

Definition 2. *Let $\boldsymbol{r} \in I$ be a replacement time series. Let $j \colon \{1, \ldots, d\} \to \{1, \ldots, d'\}$ be a function such that $j(t)$ yields the slice at time t.*
For any $\boldsymbol{z'} \in I'$, the time slice mapping function $h^{(1)}_{\boldsymbol{x}} \colon I' \to I$ yields a perturbed time series such that

$$\forall t \in \{1, \ldots, d\} \colon (h^{(1)}_{\boldsymbol{x}}(\boldsymbol{z'}))_t = \begin{cases} \boldsymbol{x}_t, & \textit{for } \boldsymbol{z}'_{j(t)} = 1; \\ \boldsymbol{r}_t, & \textit{for } \boldsymbol{z}'_{j(t)} = 0. \end{cases}$$

Some exemplary explanation found using this time slice mapping technique is exhibited in Figure 1.

To avoid over-specific assumptions, no other, more diverse strategies to splitting a series into slices apart from equi-length slices are considered. Still, we make two assumptions for this technique, which we consider reasonable. First, the explained model bases its decisions on the occurrence of patterns in the time domain (*feature space assumption*). Secondly, temporally neighboring parts of a series, i.e., points from the same slice, have similar impact on the predictions of the model (*temporal coherence assumption*). These assumptions, which have not been discussed in literature previously, are later elaborated in Section 3.3.

3.1.1. Time Domain Replacement

Finding a good replacement time series $\boldsymbol{r} \in I$ is not trivial. Replacing a slice in $\boldsymbol{x}$ with its counterpart from $\boldsymbol{r}$ must remove all structure from the slice so that the model cannot utilize any of the information it previously contained. Conversely, replacing a slice should insert as little accidental additional structure as possible to not distort the explanation. All this requires $\boldsymbol{r}$ to carry as little information as possible.

We propose the following five options for $\boldsymbol{r}$ with respect to a set of reference time series S, e.g., test data, four of which are illustrated in Figure 4.

$$\begin{aligned}
\text{Zero: } \boldsymbol{r}_t^{(1)} &= 0 \\
\text{Local mean: } \boldsymbol{r}_t^{(2)} &= \frac{1}{|S|}\sum_{\boldsymbol{s}\in S} \boldsymbol{s}_t \\
\text{Global mean: } \boldsymbol{r}_t^{(3)} &= \frac{1}{|S|d}\sum_{\boldsymbol{s}\in S}\sum_{k=1}^{d} s_k \\
\text{Local noise: } \boldsymbol{r}_t^{(4)} &\sim \mathcal{N}(\mu_{(4)}, \sigma_{(4)}^2), \text{ with} \\
\mu_{(4)} &= \boldsymbol{r}_t^{(2)} \text{ (local mean)}, \\
\sigma_{(4)} &= \text{local standard deviation.} \\
\text{Global noise: } \boldsymbol{r}_t^{(5)} &\sim \mathcal{N}(\mu_{(5)}, \sigma_{(5)}^2), \text{ with} \\
\mu_{(5)} &= \boldsymbol{r}_t^{(3)} \text{ (global mean)}, \\
\sigma_{(5)} &= \text{global standard deviation.}
\end{aligned}$$

The local and global standard deviation are computed analogous to the local and global mean.

However, most of replacements violate the requirements outlined beforehand. The two noise replacements insert additional peaks and slopes, which could be considered important information by some models. The two local replacements do not eliminate, but instead preserve the rough path of the specimen and even smuggle in the rough path of the dominant class if there is one. Finally, when a whole dataset is valued at a higher magnitude, zero replacement might introduce large jumps in magnitude.

Intuitively, global mean replacement removes most information while inserting the least accidental structure. It preserves the rough order of magnitude, does not preserve any rough path, and avoids artificial slopes and peaks altogether. It has also lead to satisfactory results in practice. However, the sharp edges it almost always produces (see Figure 3) may distort the explanations of models that are sensitive to those.

3.1.2. Direct Mapping

Returning to the grand view of time slice mapping, a special case arises when one chooses $d' = d$. In this case, each one of the time series' points makes up exactly one fragment and can be disabled individually. That way, the impact of each point is explained individually and isolated from all other points, making the explanation incredibly fine-grained.

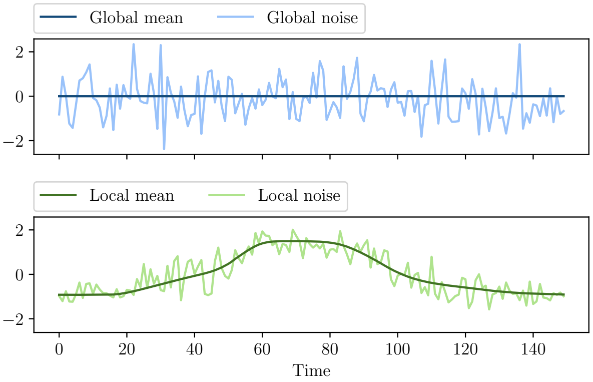

Figure 4. Comparison of four time domain replacement series, generated by four different approaches. The UCR GunPoint test set is employed as reference set S.

This *direct mapping* approach isn't aware that it is operating on time series data since the impact of each vector component of the specimen $\boldsymbol{x}$ is measured directly. Thus, we happily dispose of the temporal coherence assumption. But despite all these merits, direct mapping with its large d' has substantial drawbacks, as discussed later in Section 3.4.

3.2. Frequency Band Mapping

Some models might not only be interested in the temporal structure of a time series, e.g., hills or valleys, but also in whether the series oscillates at particular frequencies. To detect this, they first convert the series from the ordinary time domain to the so-called frequency domain via the Fourier transform. There, the frequencies that the series oscillates at are clearly visible, as illustrated in Figure 5.

While time slice mapping allows us to compute interpretable impacts of temporal slices of a time series, it fails to capture frequencies that are considered impactful by models which operate in the frequency domain. We strive to explain the decisions of a model as completely as possible, thus we need to introduce a mapping function that considers the frequency domain.

Frequency band mapping splits the frequency spectrum of the specimen into d' frequency bands with quadratically increasing bandwidths. We chose a quadratic scaling to provide greater resolution at the information-rich lower frequencies, as opposed to linear scaling, while not loosing too much resolution at higher frequencies, as opposed to logarithmic scaling. Each band $i \in \{1, \ldots, d'\}$ makes up one fragment whose activation is thus governed by $\boldsymbol{z}_i'$. An individual fragment is disabled by cutting the corresponding frequency band from the frequency spectrum of $\boldsymbol{x}$, a task which is performed using a so-called *bandstop filter*.

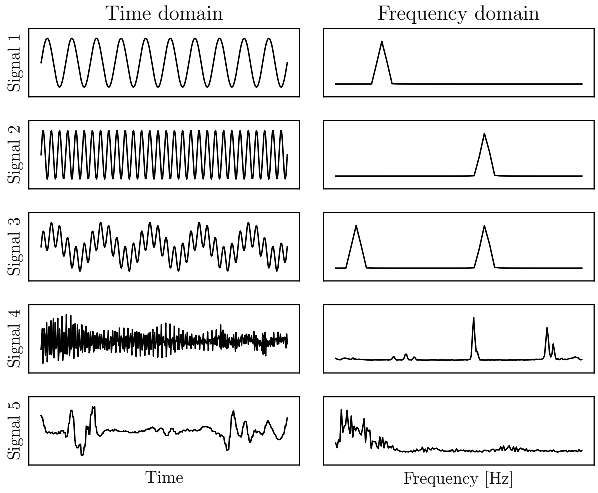

Figure 5. Intuition for the Fourier transformation. On the left, various time series are depicted in the ordinary time domain. Converting them to the frequency domain via the Fourier transform yields the frequency spectra on the right. Signals 1 and 2 demonstrate how oscillations of a single frequency in the time domain are just a single spike at that frequency in the frequency domain. Signal 3 illustrates that two oscillations laid on top of each other result in two spikes at those frequencies. Signal 4 is a complex audio signal which nevertheless has three dominant frequencies. Signal 5 is a real-world time series with a lot of different low frequencies and some high frequencies.

Two such filters will be proposed in the next section.

Summing up again, when the frequency band mapping function, which is tailored to the specimen $\boldsymbol{x}$, is queried with any simplified input $\boldsymbol{z'}$, it iteratively disables all the frequency bands i of $\boldsymbol{x}$ whose $\boldsymbol{z'_i}$ is 0 one after the other using bandstop filters and yields the resulting perturbed version of $\boldsymbol{x}$, as demonstrated in Figure 6. Let us now formally define this iterative function as a recursive loop.

Definition 3. *For any $\boldsymbol{z'} \in I'$, the frequency band mapping function $h_{\boldsymbol{x}}^{(2)}: I' \to I$ yields a perturbed time series such that*

$$h_{\boldsymbol{x}}^{(2)}(\boldsymbol{z'}) = \Lambda(\boldsymbol{z'}, 1, \boldsymbol{x})$$

with $\Lambda\colon I' \times \{1, \dots, d+1\} \times I \to I$ such that

$$\Lambda(\boldsymbol{z'}, i, \boldsymbol{z}) = \begin{cases} \boldsymbol{z}, & \text{for } i = d' + 1; \\ \Lambda(\boldsymbol{z'}, i+1, \boldsymbol{z}), & \text{for } \boldsymbol{z'_i} = 1; \\ \Lambda(\boldsymbol{z'}, i+1, \lambda(i, \boldsymbol{z})), & \text{for } \boldsymbol{z'_i} = 0; \end{cases}$$

with $\lambda\colon \{1, \dots, d\} \times I \to I$ such that $\lambda(i, \boldsymbol{z})$ cuts the band i from the time series $\boldsymbol{z}$ using a bandstop filter.

Analogous to time slice mapping, the only two assumption made by frequency band mapping are that the model considers frequency information in its decision-making process (*feature space assumption*) and that neighboring frequencies, i.e., those inside the same band, have mostly similar impact on the predictions of the model (*band coherence assumption*). Both assumptions will be elaborated in Section 3.3.

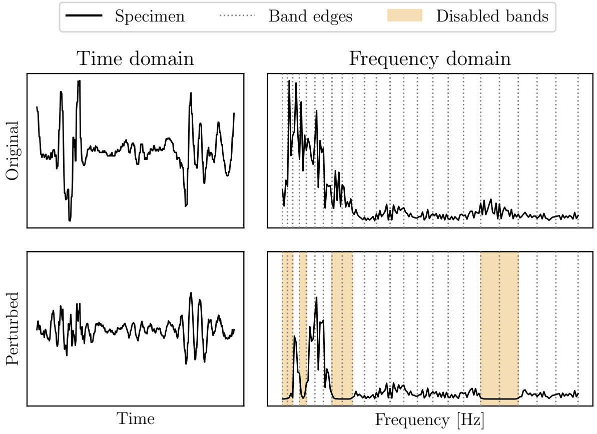

Figure 6. Demonstration of how the perturbation of a specimen (time series #64 from the UCR FaceFour test set) in the frequency domain manifests in both the frequency and time domains. Scales are linear and have been omitted for clarity. Notably, in the bottom right, one can observe how the disabled frequency bands of the specimen are almost completely set to 0 by the bandstop filters. The bottom left shows the effect of this filtering in the time domain.

3.2.1. Bandstop Filters

We propose two bandstop filters that are suitable for frequency band mapping. Exemplary frequency responses of these filters are depicted in Figure 7.

Our first proposal, an elliptic filter (Schlichthärle, 2000), is an infinite impulse response (IIR) filter that is fast in execution and can be designed to feature a frequency response with unparalleled clarity and sharpness at the edges. However, IIR filters are notorious for their unreliability, as shown by Litwin (2000), rendering elliptic filters a clean, but in no way general purpose solution. Without previous examination of how cleanly the filter cuts out the bands $i \in \{1, \dots, d'\}$ from the concrete specimen $\boldsymbol{x}$, its usage thus cannot be recommended.

Instead, we advocate employing an finite impulse response (FIR) filter designed using least-squares linear-phase optimization (FIRLS), as described by Parks & Burrus (1987). While a FIRLS filter lacks clarity and edge sharpness in comparison to the elliptic filter, its reliability makes it perfect to generate explanations on unknown specimens.

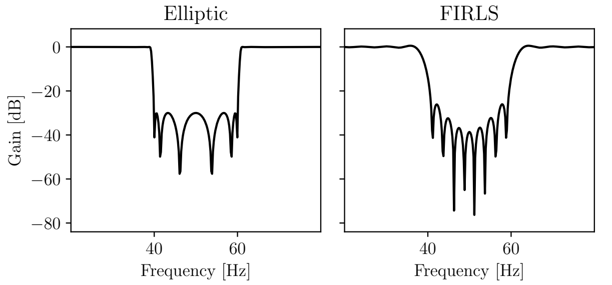

Figure 7. Frequency responses of the the proposed elliptic and FIRLS filters for a bandstop between 40 and 60 Hz. Intuitively, a frequency response shows how the amplitude of each plotted frequency is reduced by the filter, i.e., how much so-called gain is applied to each frequency. As one can see, the elliptic filter has sharp edges, i.e., highly dampens frequencies inside the band and doesn't change the amplitude of frequencies outside the band. The FIRLS filter has blurred edges and more unwanted variety in gain.

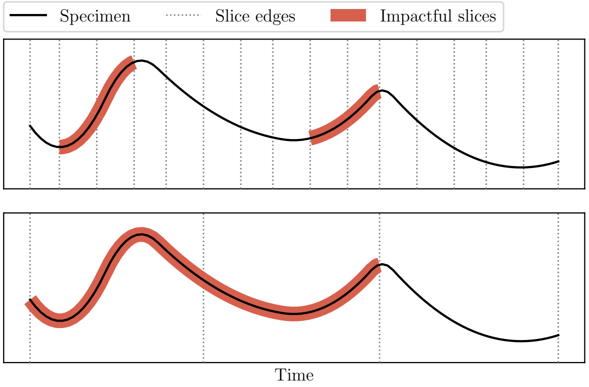

Figure 8. Toy example demonstrating the effect of utilizing too few time slices, implicitly assuming far-reaching coherence. The upper plot assumes only short-reaching coherence and shows in detail that the first and last rising slopes have a high impact on the model's prediction. In contrast, the lower plot assumes far-reaching coherence, thus employs less slices and, as a result, suggests a different and misleading impact distribution.

3.3. Consequences of Wrong Assumptions

Both time slice mapping and frequency band mapping impose assumptions on the specimen and model. We need to introduce these systematically and explore the severity of false assumptions to assess the merits of any explanation generated through those techniques.

3.3.1. Coherence Assumption

The assumption of coherence is that neighboring values have similar impact on the prediction of the model, be it in the time or frequency domain. Intuitively, this assumption sounds reasonable. A model basing its decisions only on isolated points in time or isolated frequencies, completely ignoring the neighborhoods of those points, is considered a sign of substantial overfitting.

However, increasingly far-reaching coherence, i.e., coherence spanning a longer range of points or frequencies, generally becomes increasingly implausible. Assuming far-reaching coherence where there is none results in misleading explanations, as illustrated in Figure 8. Moreover, one cannot deduce from just looking at one explanation whether the coherence assumption, which is implicitly imposed by the mapping, is correct. Therefore, those who view an impact distribution have to be aware that the truly impactful portions of the specimen might only be short parts lying somewhere inside the fragments that might be misleadingly marked as impactful in their completeness.

To reduce this issue, one is advised to use a high number of time slices or frequency bands, respectively.

3.3.2. Feature Space Assumption

The feature space assumption is that the model bases its decisions on the properties of the specimen in a particular feature space, specifically the time respective frequency domain. Such properties could, for example, be a spike in the time domain or the presence of a particular frequency band.

Let us investigate what happens if one assumes that the model works in the time domain and applies time slice mapping, when in reality, the model works in the frequency domain. When disabling a fragment in the time domain, e.g., removing a spike in time, that removal also affects the frequency domain and presumably lowers the amplitude of some frequencies. If these frequencies are impactful to the model, lowering them affects the confidence of the model's prediction. However, Kernel SHAP only witnesses that removing the spike is affecting the confidence and thus considers the spike to be impactful. Obviously, it is only indirectly impactful.

Those who view impact distributions have to be aware that the fragments which are marked as impactful might not be what the model actually attributes importance to, but instead only artifacts. Unless we are sure which domain the model operates in, we cannot deduce that, e.g., the model considers a specific interval in time important just because it is impactful in a time slice impact distribution.

3.4. Simplified Input Dimension Trade-off

Till now, the choice of the simplified input space dimension d' has been mostly left open. In Section 3.3.1, we have shown that too low d' results in low resolution and potentially misleading impacts. Intuitively, the higher we choose d', the more fine-grained and faithful to the behavior of the model our explanation becomes. Or does it?

In reality, a high-dimensional simplified input space makes the explanation susceptible to noise. The model can only be probed with a tiny portion of the perturbed specimen generated by a huge simplified input space. That is because without access to superior computational resources, it is unfeasible to run a model billions of times in a reasonable amount of time to compute just one explanation. If one ignores this, chooses a high d', probes the model with only a couple of thousands of perturbed specimens, and then fits a linear regression model g with a high number of coefficients ϕ_i and far too few collected samples, the result is noisy and overfitted.

Additionally, Alvarez-Melis & Jaakkola (2018a) have previously observed fluctuating behavior of SHAP and its predecessor LIME caused by slight changes of the specimen. They have concluded that model-agnostic perturbation-based approaches like SHAP and LIME are prone to unstable and noisy behavior. Their results make it even more important for us to choose d' small enough such that enough samples for the linear regression can be collected, reducing instability.

Summing up, both too low and too highly dimensional simplified input spaces lead to severe issues drastically reducing the quality of the explanation. In the following experimental section, we choose d' based on empirical experience. Finding good bounds for d' is left open as possible future work.

4. Classifier Comparison Experiment

We will now employ our proposed techniques to compare the behavior of a set of different time series classifiers on a variety of time series from various datasets. Our goal is to discover similarities in the decision-making process of different classifiers.

4.1. Classifiers

The extent of this experimental analysis is limited by its demand for computational resources. Still, we include at least one classifier from each of the time series classification paradigms outlined by Bagnall et al. (2017).

1. *Whole series techniques* compare the time series pairwise with a distance measure that operates on the whole series and typically allows for some shift in phase. Classification is usually done using 1-nearest-neighbor (1-NN). For our experiment, we choose state-of-the-art elastic ensemble (EE), a 1-NN ensemble of various distance measures including flavors of dynamic time warping (Bagnall et al., 2017). Also, we include a support vector machine with the global alignment kernel (SVM/GAK) that imitates dynamic time warping, as described by Cuturi (2011).

2. *Interval techniques* analyze only training-selected, phase-independent, intervals of a series. This allows to ignore noisy regions and focus on discriminatory ones. We choose the time series forest classifier (TSF) as it is fast and does not perform worse than other, more advanced interval methods (Bagnall et al., 2017).

3. *Shapelet techniques* classify series based on the occurrence of class-specific phase-independent subseries called shapelets, which are found during training. We choose shapelet transform (ST) due to its superior performance over other shapelet methods (Bagnall et al., 2017) and combine it with random forest for classification.

4. *Dictionary-based techniques* not only take into account the presence or absence of subseries, but also their repetition. Histograms are built from these frequency counts and then fed into a classifier. We choose state-of-the-art word extraction for time series classification (WEASEL) and combine it with a logistic regressor, as proposed by Schäfer & Leser (2017).

Additionally, we include random interval spectral ensemble (RISE), a recent classifier proposed by Lines et al. (2018) that extracts features in the frequency domain and utilizes them to learn a time series forest.

We are also interested in whether the decision-making process of general purpose classifiers shares any resemblance with those specific to time series. From this category, we choose rotation forest (RotF), as Bagnall et al. (2017) have shown it is leading the field of general purpose classifiers applied to time series. Moreover, we also include a support vector machine with a linear kernel (SVM/Lin). Finally, we employ residual network (ResNet) as the best state-of-the-art representative of deep learning on time series, as shown by Fawaz et al. (2019).

Note that we do neither include the DTW$_F$ ensemble and Flat-COTE (Bagnall et al., 2017) nor more recent HIVE-COTE (Lines et al., 2018) due to their huge training time complexity. They are just ensembles of the above models and thus share their properties.

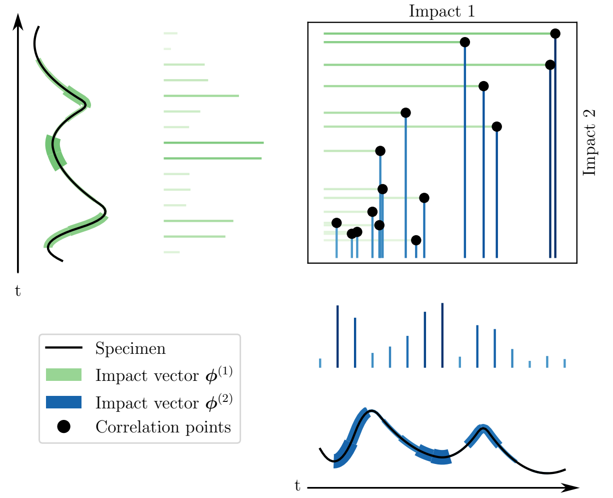

Figure 9. Toy example demonstrating how the impact vectors $\phi^{(1)}$ and $\phi^{(2)}$ of two explanations on the same specimen are compared. For each fragment of the specimen, a correlation point is plotted whose x- and y-coordinates are the impact of that fragment according to $\phi^{(1)}$ and $\phi^{(2)}$, respectively. In other words, the set of correlation points is defined by $\{(\phi_i^{(1)}, \phi_i^{(2)}) \mid i \in \{1, \ldots, d'\}\}$. As one can see, $\phi^{(1)}$ and $\phi^{(2)}$ are highly linearly correlated. Thus, the two explanations are very similar.

4.2. Experiment Setup

We conducted our computations on a server with four Intel® Xeon® Gold 5118 CPUs @ 2.30 GHz, providing 48 physical cores in total, and about 62 GB of RAM. The total wall clock time measuring our utilization of the server to its maximum capacity sums up to about 4 weeks.

Data is drawn from the UCR time series classification archive assembled by Dau et al. (2018). This archive only contains univariate time series. We exclude datasets whose time series contain missing values or are of varying lengths because some models cannot cope with such series.

For each of the remaining 114 datasets, one instance of each classifier is trained. Next, to limit computation time, five specimens are selected from each dataset's test set uniformly at random such that specimen 1 is of class 1, specimen 2 is of class 2, and so on. If a dataset has fewer than five classes, we loop back to the first class.

For each combination of classifier and specimen, we compute two impact vectors via Kernel SHAP, one with time slice mapping and one with frequency band mapping. Each pass probes the classifier with 10,000 random perturbations of the specimen. The parameter d' is chosen based on empirical experience. For time slice mapping, each time slice is configured to consist of about 5 points in time, specifically, $d' = \lfloor {}^{d}\!/\!{}_{5} \rfloor$. Conversely, for frequency band mapping, the number of frequency bands is ${}^{1}\!/\!{}_{15}$ the length of the specimen, i.e., $d' = \lfloor {}^{d}\!/\!{}_{15} \rfloor$.

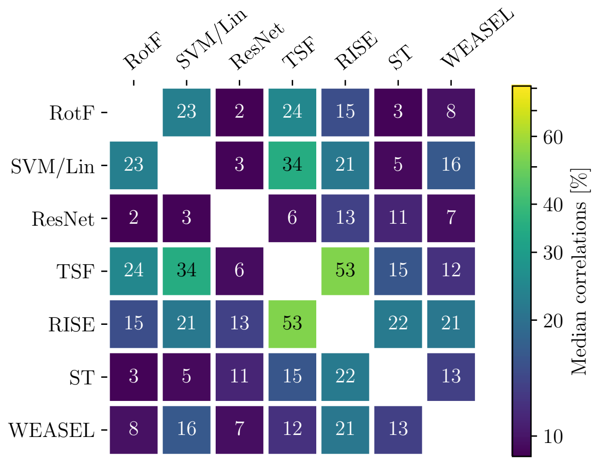

Figure 10. Medians correlations computed between pairs of classifiers using the time slice mapping.

Our final goal now is to investigate how similar the explanations of different classifiers are. For each combination of a specimen and a mapping, we compare the computed impact vectors of all available classifiers pairwise. One such comparison between two impact vectors $\phi^{(1)}$ and $\phi^{(2)}$ from different classifiers is done by computing the Person correlation coefficient between the two, which determines how strong the linear relationship between the two vectors is, as illustrated in Figure 9. We use this correlation as similarity score.

4.3. Results

First, we discuss correlations obtained through time slice mapping. Figure 10 shows the median correlations between all pairs of classifiers. Take note that all medians are positive since negative correlations are, while they do exist, rare. We immediately spot multiple occurrences of relatively high median correlation.

- With a median correlation of 53 %, RISE is notably similar to TSF. This is surprising since TSF is a time-based interval technique while RISE only operates in the frequency domain. Apart from TSF, RISE displays relatively high median correlation with other conceptually different classifiers too, namely ST (22 %) and SVM/Lin (21 %).
- TSF also shares a relatively high median correlation of 34 % with the primitive SVM/Lin classifier. Con-

sidering that SVM/Lin views a time series only as a collection of points and has no knowledge of their true sequential structure, while TSF has such insight, this is astounding. Moving on, TSF has a median correlation of 24 % with RotF, again a classifier that is not specialized on time series. Between RotF and SVM/Lin, a median correlation of 23 % completes this triangle of similarity between TSF and two non-specialized classifiers.

- WEASEL shares a considerable median similarity of 23 % with RISE. Both classifiers consider the concept of frequency and repetition in their predictions.
- The neural network ResNet does not share a single notable similarity with any other one of the compared classifiers. This suggests that the decision-making process of ResNet is unique and substantially distinct from the other tested classifiers.

Due to computation time exceeding the deadline for this thesis, we were not able to include the EE and SVM/GAK classifiers in this comparison. That experiment is left to future work.

Conversely, the medians of pairwise similarities computed via frequency band mapping are shown in Figure 11. Interestingly, these median similarities are substantially higher overall compared to the time domain. To clarify this curious observation, we have conducted extensive manual spot checks. From examining many time series spectra, we conjecture that in most UCR datasets, most of the information of a time series is contained in its lowermost frequencies. Moreover, when surveying many impact distributions and their correlations, we witnessed that the number of bands in these lower frequencies seems to be too low to engender meaningful differences. Consequently, the impact distributions all look the same in these lower bands. We are presumably experiencing a violation of the coherence assumption in the frequency domain, as illustrated in Section 3.3.1. Follow-up experiments with smaller bands and thus higher resolution are left to future work.

5. Conclusion

The increasing adoption of ML systems brings the need to explain them. Kernel SHAP, the basis for our work, is a model-agnostic framework that explains the decision-making process behind the individual predictions of any model operating on images, texts, or tabular rows.

With time slice mapping and frequency band mapping, we presented two mapping functions that transfer Kernel SHAP to the domain of time series. They facilitate the explanation of individual predictions of any time series classifier. Additionally, we carefully analyzed the assumptions made by our mapping functions and studied the behaviors users must be aware of when interpreting the explanations to properly derive insight from them. We found this rigor to be crucial when working on the sensitive field of explainable AI.

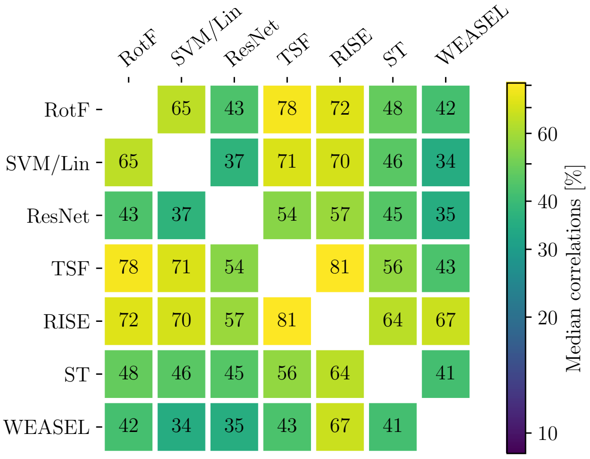

Figure 11. Medians correlations computed between pairs of classifiers using the frequency band mapping.

Finally, we presented an experimental comparison of explanations generated for a variety of different time series classifiers. Our analysis of the results surfaced surprising similarities between pairs of models which are seemingly very different. Promising future work could drill down on these similarities and maybe discover hidden parallels between these different categories of models.

Acknowledgements

We would like to thank Fabian Geier, Emmanuel Müller, and Erik Scharwächter for first posing the research question regarding how LIME and by extension SHAP could be applied to time series data, as well as Jonas Hering and Emanuel Metzenthin who independently of us came up with the basic idea for time slice mapping during in a seminar in 2017. Additionally, we would like to thank Jannes Münchmeyer, Mario Sänger, and Patrick Schäfer for many interesting discussions without whose this work would not have been possible. Finally, we thank Jorin Heide, Linus Heinzl, Nicolas Klodt, Lars Seifert, and Arthur Zahn for proofreading our drafts.

References

Alvarez-Melis, D. and Jaakkola, T. S. On the robustness of interpretability methods. In *Proceedings of the 2018 ICML Workshop in Human Interpretability in Machine Learning (WHI)*, pp. 66–71, 2018a.

Alvarez-Melis, D. and Jaakkola, T. S. Towards robust interpretability with self-explaining neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS)*, pp. 7786–7795, 2018b.

Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660, 2017.

Cuturi, M. Fast global alignment kernels. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pp. 929–936, 2011.

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

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4): 917–963, 2019.

Goodman, B. and Flaxman, S. European Union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57, 2017.

Lines, J., Taylor, S., and Bagnall, A. Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions on Knowledge Discovery from Data*, 12(5):52, 2018.

Lipton, Z. C. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

Litwin, L. FIR and IIR digital filters. *IEEE Potentials*, 19 (4):28–31, 2000.

Lou, Y., Caruana, R., and Gehrke, J. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 150–158, 2012.

Lundberg, S. M. and Lee, S. I. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, pp. 4765–4774, 2017.

Parks, T. W. and Burrus, C. S. *Digital Filter Design*, pp. 54–83. John Wiley and Sons, New York, NY, USA, 1987.

Ribeiro, M. T., Singh, S., and Guestrin, C. 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 (KDD)*, pp. 1135–1144, 2016.

Schlichthärle, D. *Digital Filters*, pp. 27–33. Springer, Berlin and Heidelberg, Germany, 2000.

Schäfer, P. and Leser, U. Fast and accurate time series classification with WEASEL. In *Proceedings of the 26th ACM on Conference on Information and Knowledge Management (CIKM)*, pp. 637–646, 2017.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

Independence Declaration

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references.

Potsdam, July 26, 2019

Felix Mujkanovic