Department of Mathematics
Chair of Mathematical Statistics

Technical University of Munich

Master's Thesis in Mathematics

Explanations from the Latent Space:

The Need for Latent Feature Saliency Detection in Deep Time Series Classification

Maresa Schröder

Examiner:	Prof. Dr. Mathias Drton
Advisor:	Dr. Narges Ahmidi, Dr. Oleksandr Zadorozhnyi, M.Sc. Alireza Zamanian
Submission Date:	14^{th} June 2022

I hereby confirm that this is my own work, and that I used only the cited sources and materials.

Munich, 14^{th} June 2022

Maresa Schröder

Abstract

Deep Leaning models are widely used for time series classification. For understanding the decision-making process of the model and identifying artifacts, explainability methods for these black-box classifiers are necessary. State-of-the-art saliency methods, originally developed for image data, assign importance scores to image pixels, providing visual explainability by highlighting informative regions in images. These methods have also been utilized for time series classification, where they equally highlight informative temporal patterns i.e. shapelets. Nevertheless, in time series data the class label might as well depend on latent information rather than temporal regions, such as a difference in the time series dominant frequencies. In this setting, common explainability methods fail to provide accurate results. We thus identify a need for improvement in explainability methods for time series. To the best of our knowledge, there are no methods currently in the literature that can visually explain how latent-patterned time series are classified. In this thesis, we shed light to this concern by empirically showing the shortcomings of current explainability methods for the mentioned time series scenario. To offer a solution, we propose an extension for existing methods which provides latent saliency results based on time-step-wise importance scores. In order to find the best candidate to augment with our extension, we examine various explainability method-classifier pairs. We restrict our study to Fourier series models and its corresponding frequency, amplitude and phase shift latent parameters, to provide a sensible scope for the thesis. We argue, nevertheless, that the same approach can be used to solve the problem with respect to other latent models. Our main focus throughout this thesis is on a local latent saliency framework, however, we provide primary remarks about obtaining global latent saliency results as well.

Acknowledgements

First, I would like to express my deepest gratitude to my advisors Dr. Narges Ahmidi, Dr. Oleksandr Zadorozhnyi and M.Sc. Alireza Zamanian for their support and guidance throughout this thesis.
I am very grateful to Dr. Narges Ahmidi for giving me the chance to learn and develop while conducting this research at Fraunhofer Institute for Cognitive Systems, Munich. Her senior expertise was extremely valuable for this research project. Dr. Oleksandr Zadorozhnyi provided constant support sharing his mathematical expertise, especially while designing experiments and proofreading. Special thanks to Alireza Zamanian for his invaluable assistance and insights throughout this project. Without his help this thesis would not have been possible in the same way.

My sincere thanks go to Professor Dr. Mathias Drton for making this joint project of Technical University of Munich and Fraunhofer Insitute for Cognitive Systems possible.

Furthermore, I would like to thank the entire team of the Reasoned AI Decisions lab for the fruitful discussions and helpful suggestions.

Lastly, I would like to thank my family for their continuous encouragement, unconditional love and unfailing support throughout my years of study and every day, especially in difficult times. Thank you.

Contents

Chapter 1

Introduction

1.1 Problem definition

Deep neural networks provide state-of-the-art performance on most time series classification problems [40]. Although these networks have become highly popular over the last two decades for various different tasks as image classification, speech recognition or document classification, there is still little insight into their internal decision process. This is not only extremely unsatisfactory from a scientific point of view [104], but also poses a high risk in real world application. Especially in safety critical applications such as health care or autonomous vehicles, interpretable and trustworthy results are essential [37]. This is why the utility of deep learning methods in real world scenarios is highly regulated and limited by the safety regulations. The pressure on developers to equip black-box artificial intelligence (AI) models with explainability and interpretability was increased when the EU General Data Protection Regulation [28] was implemented in 2016, mandating explainability in the context of automated decision making. Nevertheless, the establishment of trust and confidence in AI models is not the only requirement that calls for explainability. During model development, explainability aids to identify prediction biases and failure modes. Models often base their decision process on artifacts (aspects which falsely influence the prediction) in the data. Detection of such *Clever Hans* strategies [74] is indispensable before real-world usage [58]. Moreover, in situations where AI surpasses human capability, explainability serves as a knowledge intermediary allowing humans to extract high-level knowledge from the machines' superior decision making [83]. Due to the high complexity and non-transparency of deep learning methods, the development of explainability methods is a non-trivial task [75]. Until today, explanation and interpretation of black-box AI models, as deep learning based time series classification, remains a mostly unexplored field posing a scientific challenge [37].

With the aim of gaining insights into the relation of observations and prediction outcomes of black-box models, various explainability methods are proposed which assign importance scores to each input feature [37, 42]. To the extend of our knowledge, these importance scores are exclusively based on positional information, i.e. high scores

are assigned to class-distinctive spatial patterns and regions of the feature space. We hypothesize that this is due to the fact that explainability methods were originally mostly developed for image or static data. For image data, assigning feature importance scores to positional information (pixels) is a valid approach for explaining the prediction, since in the image domain the label is often associated with specific input regions. Consequently, heat maps based on these importance scores highlight the important regions in the image, and hence are human-interpretable, as depicted in Figure 1.1.

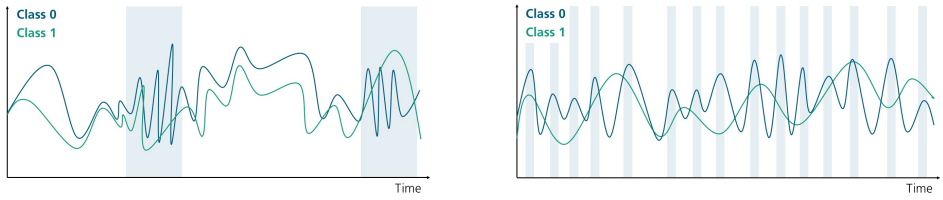
(a) Neuschwanstein castle at dawn classified as 'castle' with corresponding importance heat map.

(b) Reflection of Zugspitze in Eibsee classified as 'dam' (possibly due to reflection in water) with corresponding importance heat map

Figure 1.1: Importance scores visualized as heat maps for image classification. The more intense the red color the higher the importance of the respective pixel.

However, this approach of scoring based on positional information fails to provide interpretability in some time series classification problems. To discuss the point, we draw the readers' attentions to the underlying data generation mechanism in the time series domain. In this domain, the class label may depend on either a class-distinctive positional pattern (a shapelet) or latent features of the time series, such as a dominant frequency, state-space model parameters, or the overall trend of a non-stationary time series. In case of shapelets, assigning importance scores to every input region (i.e. time points) is justified, as the label correlates directly with positional information. In other words, visualization through heat maps is directly interpretable. An example is depicted in Figure 1.2(a), where the presence or absence of a unique pattern distinguishes two classes. In case of the latter, however, positional scores do not immediately imply any importance of the latent features, as the feature space and latent space are different. Figure 1.2(b) outlines a toy example explanation in a scenario in which the label depends on the difference in frequencies between the classes. Although a good guess could lead to the correct interpretation of the label-making feature, in the presented example the label could as well depend on a difference in amplitudes. Finally, we also notice that the independence assumption between neighboring data points which is made in the aforementioned approaches, neglects the time ordering of input features.

This leads to the inability to detect temporal dependencies [62]. As a conclusion, the current approaches which have frequently been prescribed for time series may fall short of explainability and interpretability in many scenarios.



(a) Scenario 1: Label based on the presence of certain shape in the time series

(b) Scenario 2: Label based on the latent feature frequency

Figure 1.2: Toy explainability example showing the two label-making scenarios in the time series domain. Areas of the time series with high importance scores are shaded in grey.

Based on these observations we identify the need for an extension of current explainability methods to latent feature explainability. For including temporal dependencies into the explanation, we consider this step inevitable.

1.2 Goals of the research

Schlegel et al. [82] pose the question of validity of the application of XAI methods designed for different fields to time series and propose a framework for evaluating and verifying the use of feature attribution methods on time series problems [82]. Following the same concern, we question the direct application of XAI techniques for explaining black-box time series classifiers.

Throughout our research project, we focus on explanations for the predictions using the highly popular Long Short Term Memory (LSTM) [39] and Convolutional Neural Network (CNN) [59] classifiers. Comparisons of LSTM and CNN performances as well as explainability on time series prediction tasks are presented in the literature: Suresh et al. [96] evaluate the models performance on prediction of clinical interventions in intensive care units, concluding that the performance of an LSTM is at least as high as the performance of CNNs on the specified task. The authors measure performance based on per-class area under the receiver operating characteristics curve (AUCs) and the average of the per-class AUCs. Furthermore, the authors claim to have success-

fully gained interpretability into both LSTMs and CNNs. Weytjens and Weerdt [100] compare the performance of CNNs and LSTMs as well as of LSTMs with an attention mechanism on the task of predicting the outcome of ongoing processes already early on in the process. The authors find that there does not exist a significant difference in performance of the models, thus recommending the use of CNNs due to their higher computational efficiency. We aim to challenge and extend these results by conducting experiments on synthetic data sets in which we assign class labels based on the presence of specific shapes in the time series as well as on differences in the latent features. To the extend of our knowledge this is the first study investigating explainability on latent features in time series classification. For the scope of this thesis we focus on binary classification of univariate time series.

We find that none of the evaluated explainability methods can provide reasonable results when the label is based on a difference in the latent features. This strongly supports our argument of the need for a latent feature importance detection method. We identify the input-cell attention LSTM model [43] as a good candidate for moving toward latent space explainability. Comparing to standalone LSTMs, this model shows drastic improvement in the binary classification performance, measured by a combination of multiple metrics [20] including AUROC and F1 score. We show that in addition to the improved performance, the importance heat maps provided by the attention mechanism are the most precise across all data sets and evaluated explanation methods. Based on these observations we present an extension of this model to provide a latent feature explainability framework. Our framework is evaluated on various synthetic data sets.

1.3 Contributions

Our main contributions are as follows:

1. We design a study comparing the performance of commonly applied XAI techniques on explaining classification results provided by different LSTM and CNN architectures.

2. We evaluate and extend the results provided by former studies by conducting experiments on synthetic data sets in which we assign class labels based on the presence of specific shapelets in the time series as well as on differences in the latent features.

3. We empirically show that common XAI methods fail to provide accurate explanations when the class-label is based on the latent features of the time series, identifying the need for a latent feature importance detection method.

4. Finally, we present an extension for XAI methods, quantifying to what degree the provided explanation should be interpreted in terms of the time dimension or the latent space of the input time series.

1.4 Notations

Throughout this thesis $X = (x_1, ..., x_T)$ and $Y = (y_1, ..., y_T) \in \mathbb{R}^T$ represent time series where each time step is denoted by x_t or y_t, $t = 1, ..., T$ respectively. The data set D consisting of such time series is described by $D = \{X_i\}_{i=1,...,n}$. All time series are assumed to be of length $T \in \mathbb{N}$. The investigated classification models f commonly consist of weight matrices W and biases b aside further non-linearities. In the experiments and model development, the concept of Fourier series is employed. The Fourier coefficients are presented by the amplitude A, frequency ω and phase shift ϕ parameters. An initial offset of the Fourier series is described by a_0. Importance scores, or saliency scores, of feature x_i are stated as $s(x_i)$. The complete saliency vector of the whole time series X is refered to as $SV(X) = (s(x_1), ..., s(x_T))$.

1.5 Outline

The remainder of this thesis is organized as follows: **Chapter 2** provides an overview of the current state-of-the-art explainability methods for deep learning models with a special focus on the applicability to time series classification. We provide a general categorization of explainability approaches and outline the focus of this project. Then, we explain various methods in detail. Subsequently, common deep learning methods for time series classification are described. In **Chapter 3** the experimental framework is presented. First, the generation mechanism of the synthetic data sets employed in the experiments is described. Then, a detailed overview of the investigated classifier-explainability pairs as well as the implementation details is provided. **Chapter 4** illustrates our new framework for latent feature saliency detection. After introducing the general idea and intuition behind our framework, the calculations and the final algorithm are outlined in detail. Furthermore, a global baseline approach to detecting latent feature importance is presented. Results of the conducted experiments are provided and discussed in **Chapter 5**. The final **Chapter 6** concludes this thesis, summarizing the major results and contribution as well as possible future work.

Chapter 2

Literature review

Explainable artificial intelligence is a continuously growing field in the literature. Before conducting experiments in Chapter 3, we present an overview of explainability methods and commonly used deep learning methods for time series classification. Section 2.1 introduces multiple explainability methods, focusing on ante-hoc techniques in Section 2.1.1 and on post-hoc methods in Section 2.1.2. Subsequently, the need of explainability for Long Short Term Memory networks and Convolutional Neural Networks is discussed in Sections 2.2.1 and 2.2.3. The focus is put on explaining the classification decision of LSTMs which poses a challenge, as Recurrent Neural Networks (RNNs) suffer from the vanishing saliency problem described in Section 2.2.2.

2.1 Explainability methods

Deep learning models are highly complex. The model parameters are in general very abstract and not directly interpretable [17]. Therefore, especially in the field of deep learning, there is an immense need for techniques aiding to explain and interpret the models' decision processes.
Explainability methods can be classified by their scope or type. An explainability method is called *global* if it provides results with regard to the overall decision-process on the entire dataset. A possible application of such global explainability methods in the banking sector could be the description of factors which influence the outcome of a default risk prediction model for rating customers' solvencies. On the other hand, a method is said to be *local* if it provides explanation for the prediction of a single observation [26, 37]. Continuing the previous example, to explain why one certain customer was not provided with a loan based on the decision of the default risk prediction model, a local explainability method need to be applied. For very complex models such as deep neural networks, global methods are difficult to design [4, 81]. Thus, most proposals in the literature are restricted to local explanations.
In this project we focus on local explainability with a special emphasis on post-hoc analysis described in Section 2.1.2. There exist multiple taxonomies of explainability methods in the literature [9, 8, 67]. We suggest and structure this thesis based on the taxonomy of explainability methods presented in Figure 2.1.

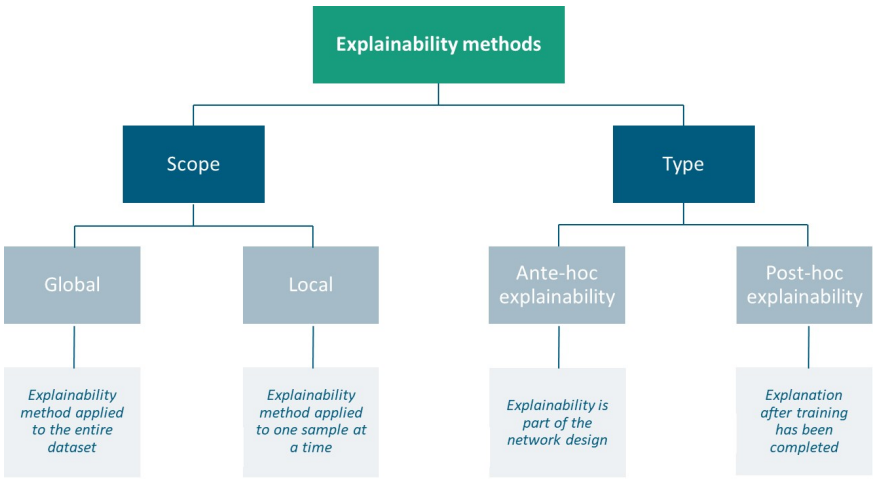


Figure 2.1: Taxonomy of explainability methods

2.1.1 Ante-hoc explainability

Multiple statistical and deep learning prediction models are explainable by default or by design [17]. These models are said to be *ante-hoc* explainable. Decision trees, decision rules, k-nearest neighbors or linear models are examples for naturally explainable machine learning models. In contrast, deep learning models are not naturally explainable and have to be designed to provide ante-hoc explainability. For example including attention mechanisms in the network architecture can provide explanation for prediction outcomes (see chapter 2.1.1). In general, altering the architecture also changes the decision process of the model. Therefore, including explainability designs into the model might pose a trade-off between prediction accuracy and interpretability [6]. Different ways of achieving explainability within deep neural networks are proposed in the literature: Self Explaining Neural Networks (SENN) generalize a simple linear predictor to a complex network structure by retaining the properties explicitness, faithfulness and stability [2]. A class of architectures called Explainable Deep Neural Networks (xDNN) employs prototypes, e.g. data samples which represent local peaks of the empirical data distribution, to create interpretable if-then rules that reflect the internal dynamics of the network [6]. Attention-based deep learning models learn to focus on important aspects of the given input, thus increase the performance and provide insights into the decision process. The employed attention mechanisms [12] are described in more detail in the following paragraph.

Attention mechanisms

First introduced in [12], attention mechanisms brought a revolution not only to ante-hoc XAI methods, but also to model performance, especially in neural language processing. Originally designed for recurrent encoder-decoder structures, attention mechanisms take the hidden states of the encoder and decoder as inputs to provide a context embedding of the hidden states based on the alignment of input and output at a certain point in the sequence [50]. Through drawing parallels to the allocation of human visual attention [76] for interpreting the mechanism, the attention mechanism can help to understand the internal decision process of the network [102].
Let $h_j = f_E(x_t, h_{t-1})$ represent the encoder's hidden state at time j, $j = 1, ..., T$ for input sequence $X = (x_1, ..., x_T)$, and $s_i = f_D(s_{i-1}, y_{i-1}, c_i)$ represent the hidden state of decoder f_D at time i, $i = 1, ..., T$, where y represents the target, f_E and f_D are non-linear functions. For constructing a context vector c_i, an alignment model relating s_{i-1} and h_j is computed as

$$e_{ij} = a(s_{i-1}, h_j),$$

where a is commonly chosen to be a feed-forward neural network. The network parameters are jointly learned with the rest of the systems' parameters. This constitutes a soft-alignment allowing for backpropagation of the gradient. For each annotation h_j, an attention weight

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})}$$

is assigned, representing the importance of h_j for the calculation of the next state s_i and target y_i. The context vector c_i is then calculated as

$$c_i = \sum_{j=1}^{T} \alpha_{ij} h_j,$$

which can be interpreted as the expected encoder hidden state over all possible alignments [12]. Lin et al. introduce self-attention mechanisms for sequential models in [63] which produce an embedding of the hidden layer of a recurrent neural network. To account for the possibility of multiple components composing the overall interpretation of the input, various context vectors focusing on different parts of the input need to be computed. The authors refer to the number of context vectors as *attention hops*. By extracting the attention weights α_{ij} from the model, insights into the inner process of the model can be obtained. Since attention mechanisms are mostly applied after an encoder block, they provide interpretable importance weights for the hidden states [101]. Only few other works such as [43] propose to apply the mechanism at different points in the model architecture as for example directly on the input sample.

Brunner et al. [16] prove that in many applications, self-attention weights are not identifiable. This is implied by the fact that an infinite set of attention weights might yield the same prediction outcome. Since it is proven that not the whole attention weight contains information affecting the model's prediction, the authors propose to focus on effective attention, the part of the attention weights which does influence the prediction, to enhance model explainability which is followed up by [94].

Whether attention weights should be interpreted as explanation is an ongoing discussion. Many are of the opinion that if attention can be seen as explanation depends on a faithful assignment of importance scores to the input features [45, 101]. They conclude that attention must not be interpreted as explanation, since in many cases attention weights are uncorrelated with feature importance scores obtained from post-hoc explainability methods, as in [45]. Nevertheless, this can be misleading, since explanations in general are not mutually exclusive [101]. Bastings and Filippova [14] argue that saliency methods constitute better explanations than attention mechanisms, since the latter only provide importance weights for a representation of the input at a certain point in the computation path of the prediction model, whereas saliency methods assign importance scores based on the complete model. Various other extensions of the attention mechanism to improve plausibility and faithfulness when interpreting attention weights as explanation exist [21, 22, 56]. A different field of study focuses on including theory from learning with rationales into XAI methods. Human domain knowledge and annotations explaining a samples label are referred to as *rationales* [92]. Several studies on this supervised form of attention demonstrate the superiority of attention mechanisms trained with human rationals in terms of explainability and classification accuracy [7, 48, 92].
With our experiments, described in chapter 3, we hope to contribute to this discussion, showing that in our specific scenario attention does provide explanation.

2.1.2 Post-hoc explainability

Given a trained black-box model, *post-hoc* explainability methods aim to describe the reason for the prediction outcome of a specific sample. One class of post-hoc explainability methods provides attribution scores, i.e. relevance scores, for each input feature. These scores relate to the effect of the respective feature on the prediction. Feature attribution values are often visualized in form of a heat map. Methods which do not need access to model parameters are called *model-agnostic*. All other methods which need information about the internals of the prediction model are refered to as *model-specific* [78].

Model-specific feature attribution methods

Model-specific post-hoc explainability methods, also introduced in the literature as back-propagation-based methods, propagate the gradient from the output of a prediction model back to the input instance [20]. This type of feature attribution method is also refered to as white-box approach, since access to the internals of the to-be-explained model is necessary [73]. Gradient-based methods assume a direct relationship between the magnitude of the gradient of the output with respect to a certain input feature and the importance of the respective feature for the prediction. Since they only require a forward and a backward pass through a model, these methods are especially computationally efficient [75]. The attribution method Saliency [89] directly employs gradients to generate saliency maps. Simonyan, Vedaldi, and Zisserman [89] define saliency as part of a linear approximation to a prediction model f

$$f(X) \approx s^T X + b,$$

where b represents a bias and the saliency score s for feature x_i of input X is calculated as

$$s = \frac{\partial f}{\partial X}\bigg|_{x_i}.$$

These scores are only precise in a very restricted neighborhood around the input [4]. Bastings and Filippova [14] and Ancona et al. [4] argue that raw gradients only express sensitivity, whereas multiplying the gradient with the input [88] represents the marginal effect of the respective input feature to the prediction outcome, thus expressing saliency. Thus, saliency s of input feature x_i for a prediction $y = f(X)$ should be defined as

$$s(x_i) = x_i \cdot \frac{\partial f(X)}{\partial x_i}.$$

This definition of saliency directly represents the contribution of each input feature to the output of the model f [4]. Furthermore, Shrikumar et al. [88] note that multiplying the gradient with the input significantly improves saliency maps for visualizing pixel-wise importance scores in image classification tasks.
Multiple extensions and adaptions of the basic approaches of Saliency and Gradient $\times$ Input have been proposed since. For reducing the noise in the gradients, DeCov-Net [104], Guided Backpropagation [91] and SmoothGrad [90] introduce adaptions of backpropagation rules [31].
Activation functions employed in the networks might challenge the usage of gradient-based approaches. Rectified Linear Units (ReLUs) might lead to a zero gradient although information might be conveyed through the activation function. Sigmoid or tanh activations as commonly applied in RNNs, see section 2.2, produce near-to-zero gradient for low and high inputs, therefore not reflecting a possible high

importance of these inputs [88]. Deep-Lift [87] overcomes this problem by using a neuron attribution-based difference-from-reverence approach to assign attribution scores. The reference-based method of Integrated Gradients (IG) determines saliency scores by calculating the path integral from a non-informative baseline input to the respective input feature [95], thus tackling the problem of saturation of the gradient [14]. In comparison to most other attribution methods, IG fulfills the three desiderata of *Sensitivity*, *Implementation Invariance* and *Completeness*. Sensitivity in an attribution method is achieved if for every two inputs differing in one feature but assigned a different class label by the prediction model, the respective feature is assigned an importance value different from zero. Implementation invariance is achieved if for two prediction models with equal prediction outcomes for all inputs, attribution methods assign equal importance scores to the input. The completeness desideratum is satisfied when the sum of all attribution values of an input equals the difference between the prediction and the prediction of the baseline input [95].
Only applicable to CNNs (see section 2.2.3), Class Activation Mapping (CAM) [107] and GradCAM [83] visualize attribution scores as a weighted sum of feature maps. Guided Grad-CAM [83] provides high-resolution attribution scores combining Guided Backpropagation [91] and GradCAM.

Relevance-based methods such as Layer-wise Relevance Propagation (LRP) [11] and Deep Taylor Decomposition [68] determine final attribution scores by propagating relevance scores from the output backward through the network via various designed propagation rules. These methods thus differ from gradient-based feature attribution methods in the way of decomposing the relevance scores at each layer. Montavon et al. propose to propagate relevance via deep Taylor decomposition. Attribution scores are enforced to be positive only. This implies the strong assumption that only evidences for the predicted outcome are relevant, neglecting possible evidence against the prediction. This is an oversimplifying assumption in many real-world applications [3]. LRP uses the output probability for the target class as the initial relevance score. For propagating the scores through different types of network layers, layer-specific propagation rules are employed. The LRP approach suffers from high noise in the relevance scores and a lack of class-discriminativeness [46]. As improvements, multiple extensions of the method, as contrastive LRP [36], softmax gradient LRP [44] and selective LRP [46], were proposed.

Model-agnostic feature attribution methods

In contrast to model-specific white-box methods, model-agnostic feature attribution methods can be applied to any black-box classifier, even without physical access to its internals [20, 73]. This fact makes them especially popular in settings where the prediction model itself is not directly available. The understanding of feature impor-

tance in this class of attribution methods differs from the definition of importance in the gradient-based or relevance-based methods. Importance is related to the change in output when the respective feature is perturbed.

Many methods employing different perturbation approaches are proposed in the literature. The Occlusion technique [104] assigns saliency scores based on a drop in the class probability when systematically occluding various input regions. The LIME method [77] locally approximates a classifier by fitting an interpretable surrogate model, e.g. a linear model, on randomly drawn samples in a neighborhood of the input sample. Afterwards the original model is explained through suitable features of the interpretable model. The Meaningful Perturbations method [30] combine aspects from gradient-based methods as Saliency and Integrated Gradients together with the underlying idea of local explanations from LIME to formulate explanations as meta predictors. One disadvantage of LIME is that highly non-linear models might not be closely approximated by a linear model. This might prevent the method from capturing important input features [73]. Petsiuk, Das, and Saenko overcome this problem by analysing the effect of random masking of input features on the outcome. Their method Randomized Input Sampling for Explanation (RISE) [73], linearly combines the binary masks used for occluding the input weighted by the output probabilities of the respective masked input to generate the final importance scores.

Other methods are inspired by theorems from the field of game theory [24, 64, 93]. Among all, the application of the Shapley Value [85] has achieved great popularity. This method determines the average marginal contribution of an input feature to the output. Lundberg and Lee define a new class of additive feature importance measures in [66] and prove that there exists a unique importance measure within this class i.e. the Shapley value, which fulfills the desirable properties of local accuracy, consistency and the ability to function in presence of missingness. Examples of explainability methods belonging to this class are LIME, DeepLIFT and Layer-wise relevance propagation. Based on their findings, Lundberg and Lee propose to measure feature-importance as the Shapeley value of a conditional expectation function of the to-be-explained prediction model. They introduce the SHAP values method [66]. For a more detailed description of Shapley values and SHAP refer to Section 3.2.

Since for the perturbation methods, the prediction model must be re-run for every single perturbation of the input sample, these methods are highly computationally expensive. This states a challenging trade-off between better attribution through more perturbations per sample (or more perturbed samples) and computation time [81]. Furthermore, performance decreases in the number of input features [49]. This implies that the model-agnostic feature attribution methods are more applicable in settings with fewer features.

Multiple model-specific as well as the perturbation-based feature attribution methods require the definition of an uninformative reference value. The performance of the methods is known to highly depend on the choice of a suitable value. In the image domain, this is mostly chosen to be an either completely white or black pixel. For the perturbation of time series inputs, in a lot of cases the mean value of the input sample is chosen. A different approach to perturbing a sample is adding noise to the respective features [73]. Sundararajan, Taly, and Yan [95] justify and motivate the use of a baseline input as a human-interpretable attribution allocation method via counterfactual intuition. The idea of counterfactual explanations is discussed below.

Counterfactual explanations

Counterfactuals highlight to what extend the value of a certain input feature would need to be modified to change the classification outcome. This explainable case-based reasoning technique has become particular popular in recent years [54], since counterfactual explanations are argued to be causally informative [53], easily interpretable by humans [18, 47, 81] and legally compliant with the GDPR of the European Union [53]. As for the other classes of post-hoc explainability methods, only few counterfactual methods are designed to explain the outcome of a time series classification problem [25, 38, 51]. Nevertheless, multiple counterfactual generation methods have been adapted to or designed for the time series domain recently. Guidotti et al. propose to explain the decision of a time series classifier by learning a shapelet-based decision tree to construct counterfactual examples, highlighting shapes which must not be present in the time series to receive a certain prediction [38]. Wang et al. extend an approach to generate counterfactuals in latent space representation through auto-encoder models from the image domain to time series classification [98]. Other approaches to the construction of time series counterfactuals include time series tweaking [52], or minimal perturbation towards a nearest unlike neighbor time series using a greedy search algorithm [10]. Delaney, Greene, and Keane as well as Keane and Smyth identify properties which a good counterfactual explanation for time series classification should fulfill [25, 53]. Besides being as close as possible to the to-be-explained instance (to achieve the property of proximity), informative counterfactual methods should also give plausible explanations by generating counterfactuals within the data domain. Furthermore, good counterfactuals need to provide sparse explanations, perturbing only as few features as possible. In [25] a counterfactual generation method fulfilling these properties is proposed. A so-called 'native guide' is determined, which is a sample from the training data set closest to the to-be-explained instance under the dynamic time warping distance with a different class label. Afterwards, the sample is perturbed towards the decision boundary.

As a summary, all of the explainability methods mentioned in this chapter provide

feature-based explanations, where in the setting of univariate time series classification, features correspond to time steps. None of these methods is designed for providing reasonable explanations, when the classification is based on latent features of the time series instead of certain time steps as stated in Chapter 1. Based on this finding, we hypothesise the need for latent feature explainability methods for time series classification.

2.2 Deep Learning methods for time series classification

Deep neural networks are widely used for time series classification and have proven highly effective in many applications over the past years. Two families of neural networks are commonly used for sequential tasks: Recurrent Neural Networks (RNNs) [79] and Convolutional Neural Networks (CNNs) [59]. Recurrent Neural Networks are especially designed to process sequential data. Since they are able to process much longer sequences than other classes of neural networks while keeping some amount of memory, RNNs have become particularly popular for time series classification and prediction tasks. Convolutional Neural Networks, originally designed for the image domain, have become highly popular for time series classification as well. Ismail Fawaz et al. [40] find, that due to their robustness and computational efficiency, CNNs are the most commonly employed deep learning models for time series classification tasks.
In the following section Recurrent Neural Networks, especially the Long Short Term Memory (LSTM) model, and Convolutional Neural Networks are briefly summarized. We then continue with describing the explainability difficulties for these models.

2.2.1 Recurrent Neural Networks and Long Short Term Memory

Recurrent Neural Networks leverage the learned dependencies between data points in the learning process of the model by introducing feed-backs into the network structure [34]. This class of networks defines recurrent functions rf to map an input $x_t \in \mathbb{R}^n$ at time t and hidden state $h_{t-1} \in \mathbb{R}^m$ from the previous iteration to an updated hidden state

$$h_t = \mathrm{rf}(x_t, h_{t-1})$$

[86]. When gradients are propagated back through multiple layers, they tend to either vanish or explode, since weights are exponentially decreasing over the iterations. Thus, long-term dependencies will naturally be assigned very small weights [34]. To overcome the vanishing gradient problem, the cell structure of the Long Short Term Memory system (LSTM) was designed [39], providing a constant error flow. For LSTMs the recurrent function rf changes to

$$h_t = \mathrm{rf}(h_{t-1}, c_{t-1}, x_t),$$

where c_{t-1} defines a cell state calculated as

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh W_{x_c} x_t + W_{h_c} h_{t-1},$$

with input gate i_t, forget gate f_t and output gate o_t controlling the gradient flow by

$$\begin{aligned} i_t &= \sigma(W_{x_i} x_t + W_{h_i} h_{t-1}), \\ f_t &= \sigma(W_{x_f} x_t + W_{h_f} h_{t-1}), \\ o_t &= \sigma(W_{x_o} x_t + W_{h_o} h_{t-1}), \end{aligned}$$

where σ represents the sigmoid function and $\odot$ the element-wise multiplication. All weight matrices $W_x \in \mathbb{R}^{m \times n}$ and $W_h \in \mathbb{R}^{m \times m}$ are learned during training of the network. The gating structure allows the LSTM to decide how much weight to put on the current input through i_t, how much information from previous steps to delete through f_t and how much the current input should impact the output at step t through o_t [43]. The final equation of the recurrent function then assigns a value to the hidden state h_t as

$$h_t = o_t \odot \tanh c_t.$$

When the gradient of the output y_T with respect to the input x_t

$$\frac{\partial y_T}{\partial x_t} = \frac{\partial y_T}{\partial h_T} \left(\prod_{i=T}^{t+1} \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_t}{\partial x_t}$$

is propagated back through the network, the vanishing gradient problem might reduce the partial derivatives of the hidden states to

$$\frac{\partial h_t}{\partial h_{t-1}} \approx o_t \odot (1 - tanh^2(c_t)) f_t.$$

Thus, the forget gate f_t can prevent the gradients from vanishing [43].

Although LSTMs are widely used for time series prediction or classification tasks, it is known that these models still have difficulties learning long-term dependencies [50]. A proposed solution for the memory limitation of LSTMs is the incorporation of attention mechanisms into the network architecture. In most of the proposed architectures in the literature, a self-attention mechanism is applied to the hidden states of an LSTM model. The authors of [43] propose to apply a self-attention mechanism directly on the input before feeding the context vector as an input to the LSTM cell structure. For more details see Section 2.2.2.

The feedback loop makes RNNs extremely non-linear, keeping humans from understanding the complex internal decision process. For increasing the performance of the model, in many cases multiple networks are stacked on top of each other, using the hidden states of one recurrent block as an input to the next RNN, which makes explanation of the complete architecture nearly impossible. Explainability techniques as described in Section 2.1 provide a first step towards understanding and interpreting the models. Nevertheless, explainability methods taking into account the temporal dependencies in time series data which can be applied to RNNs are almost non-existent. Thus, there is a great need for an extension of commonly employed explanation techniques to incorporate importance information of the temporal dynamics represented by the latent features of a time series.

2.2.2 Vanishing saliency problem in RNNs

Recurrent neural networks suffer from the vanishing gradient problem. As explained in Section 2.2.1, the cell structure of the LSTM including the gating mechanisms was designed to diminish this problem. Nevertheless, the authors in [43] show that the gating mechanisms cannot sufficiently reduce the vanishing gradient problem to allow gradient-based explainability methods to correctly identify important intervals at earlier time steps. The amount of reduction in the gradient is controlled by the forget gate, as depicted in Section 2.2.1. For early time steps, the gradient-reducing effect is multiplied many times, leading to a diminishing gradient and thus decreasing relevance of these time steps [43]. This poses a great challenge to the explainability of RNNs. In the following subsection, two approaches addressing this problem are described.

Input cell attention

As an approach to reduce the vanishing saliency problem in LSTMs, the authors in [43] propose a network extension by modifying the input of the LSTM cell. A self-attention mechanism $\mathcal{A}_t$ consisting of a two layer unbiased feed-forward neural network, as proposed in [63], is used to produce a fixed-size weighted embedding vector M_t of the input $X_t \in \mathbb{R}^t$ for every time step t

$$M_t = \mathcal{A}_t X_t$$

where

$$\mathcal{A}_t = \text{softmax}(W_2 \tanh(W_1 X_t^T)).$$

In this formulation, the softmax function is applied to every row of the raw attention matrix separately. By passing the vector M_t as an input to the LSTM cell, the network is forced to learn the weight matrices W_1 and W_2 such that the final attention weights i.e. the entries of the matrix $\mathcal{A}_t$, are representing the importance of the respective time steps t.

Saliency guided training

A different approach to diminishing the vanishing saliency problem is introduced in [41]. Designed to reduce noise in gradients, the training procedure improves the explanations from gradient-based saliency methods. Following the rationale that high gradients of the output with respect to the input represent high importance of the respective input feature, time steps with low respective gradients are iteratively masked by a predefined baseline value. The network is trained to jointly minimize the cross entropy loss function and the empirical point-wise Kullback-Leibler divergence between the outputs corresponding to the original input and the masked input by a gradient-based optimizer such as Adam [55]. By producing sparse and less noisy gradients, the training procedure also reduces the vanishing saliency problem.

2.2.3 Convolutional Neural Networks

Convolutional neural networks are most commonly applied in the computer vision and signal processing domains. One reason for the great performance on various tasks as object recognition, face verification and audio classification is the natural property of CNNs to learn highly complex feature representations [23]. One-dimensional CNNs have become popular for time series classification in the last years. By convolving over the time axis, a convolutional layer can detect local patterns, as distinctive shapes, in the time series. More complex patterns can be learned through various stacked layers [23]. The one-dimensional convolution can also be interpreted as a non-linear transformation of the input time series [40].
Convolutional layers can be subdivided into three subsequent stages. The outputs of the first stage, consisting of the convolution of the input, are fed into non-linear activation function. In the last stage, the pooling stage, a summary statistic of outputs in a certain neighborhood is calculated. The pooling operation is used to make the layer output more invariant towards minor shifts in the input. Local translation invariant networks can be helpful in tasks in which information about the presence of a feature is more important than information about its precise location [34].
Let γ represent a convolutional filter of length $l + 1$, and a and b be a non-linear activation function and bias respectively. Without loss of generality l is considered to be an even number. The result C of a convolution over the univariate time series $X = (x_1, ...x_T)$ is formulated as

$$C_t = a(\gamma * X_{[t-\frac{l}{2}:t+\frac{l}{2}]}(t) + b) \quad \forall t \in [1, T],$$

where $*$ represents the convolution operation and $X_{[a,b]} = (x_a, ..., x_b)$. Due to the convolution of the whole input with the same shared filter γ, these learned filters are time-invariant [40]. Furthermore, the sparsity in weights induced by the filter weight

sharing results in a much higher computational efficiency of CNNs compared to RNNs.

For time series prediction and classification, many extensions of convolutional neural networks are proposed. Wang, Yan, and Oates propose a Fully Convolutional Network (FCN) as a baseline approach and starting point for research on time series classification [99]. Although the architecture is deliberately kept simple, by incorporating only one convolutional layer and batch-normalization as well as a ReLU activation function (which prevents heavy preprocessing or data engineering), the network achieves performance comparable with more complex counterparts. As an extension Karim et al. [50] propose a concatenation of an LSTM with a FCN (LSTM-FCN) as well as an Attention-LSTM-FCN, which achieve significantly better performance than the FCN on its own. To avoid loss of information through feature extraction in a separate preprocessing step, the Multi-scale Convolutional Neural Network (MCNN) [23] combines classification and feature extraction in one model. By employing a smoothing and a down-sampling convolutional operation in addition to an identity mapping, it exploits the property of CNNs to naturally learn feature representations in time domain as well as frequency domain. Other approaches include the Time Le-Net [60], Time-CNN [105] and Temporal Convolutional Network [13], where the latter is described in more detail in the next paragraph. Especially designed for multivariate time series classification, the Multi-Channels Deep Convolutional Network (MC-DCNN) [106] applies convolutional filters to each univariate time series separately, before combining the learned feature representations in the final classification layer.

Although activation maps of convolutional layers can be employed to provide insights into the internal decision process of the models, these maps only focus on positional information. Hence, the bias of common explainability methods of neglecting non-positional information when assigning importance scores to input features, can also not be mitigated in CNNs.

Temporal Convolutional Network

A temporal convolutional network (TCN), as introduced in [13], aims to integrate benefits of recurrent neural networks in the sequential domain into a convolutional network structure. Based on the structure of a fully convolutional neural network [65] with fixed hidden layer size equal the length of the input and zero-padding of size (convolutional filter $-$ 1), the network provides a mapping from an arbitrary sized input to an output with the same length, in the fashion of RNNs. The complete fully convolutional network can be interpreted as a non-linear filter rather than an arbitrary non-linear function. In many scenarios in the sequential domain, a leakage of future information to the past is undesirable, a property which standard convolutional neural networks cannot guarantee. As a solution [13] introduces causal convolutions to

the network, restricting the convolution to the current and past time steps only. For achieving a history size exponentially increasing with the depth of the network, dilated convolutions are employed at every layer, introducing a step length of dilation factor d between all neighbouring cells in the respective filter. The authors argue that, due to the exponentially increasing receptive field, TCN inherits longer memory compared to RNN architectures.

Chapter 3

Experimental framework

The experiments were conducted in two phases. First, the performance of multiple local explainability methods in combination with each three types of LSTM and CNN classification models was evaluated on various synthetic data sets. Afterwards, the best-performing (classifier, explainability method) combination was extended to provide an importance score for the latent features of the instance of interest.
The generation mechanisms of the employed data sets are described in Section 3.1. In Section 3.2 the classifiers and explainability methods which were tested are described. Section 3.3 presents details about the implementation of the methods.

3.1 Data generation

In order to develop a saliency method for the latent features of a time series in a supervised manner, we generate multiple synthetic data sets according to a number of simulation scenarios. We consider two scenarios where the time series labels depend on either the presence of a specific shape in the series or differences in the underlying latent features as stated in Chapter 1. By comparing the explainability methods on our generated data sets of these scenarios, we argue in Chapter 5 that explainability methods fail in the second scenario. Furthermore, we extend the experiments presented in [41], and investigate the reduction of the vanishing saliency problem in more realistic data sets by allowing the temporal position of the label-informative shape to vary among samples.

For the scope of this project we restrict the analysis to discrete time stationary periodic signals modeled by a Fourier series. Since, by the Fourier theorem, any periodic time series can be uniquely represented as a Fourier series, we consider the generation of Fourier series as a valid approach for our analysis and the development of a latent feature saliency method. Thus, we can assume that for a time series X each time step

$x_t, t = 1, ..., T$ can be represented as

$$\begin{aligned} x_t &= a_0 + \sum_{n=1}^{\infty} a_n \cos(\omega_n t) + \sum_{n=1}^{\infty} b_n \sin(\omega_n t) \\ &= a_0 + \sum_{n=1}^{\infty} A_n \cos(\omega_n t + \phi_n) \\ &= a_0 + \sum_{n=1}^{\infty} A_n \sin(\omega_n t + \phi_n + \frac{\pi}{2}). \end{aligned}$$

Let $\tilde{n}$ represent the number of amplitudes present in the series, i.e. $\forall i > \tilde{n}, A_i = 0$. For the sake of simplicity, only centered stationary periodic time series are considered in the data generation process, i.e. $a_0 = 0$. Then the value at every time step t is calculated as

$$x_t = \sum_{i=1}^{\tilde{n}} A_i \sin(\omega_i t + \phi_i + \frac{\pi}{2}).$$

We refer to the notions amplitude A, frequency ω, phase shift ϕ as *concepts*. The separate Fourier coefficients A_i, ω_i, ϕ_i for $i = 1, ..., \tilde{T}$ are refered to as latent features. The latent features frequency ω_i and phase shift ϕ_i are each sampled from a uniform distribution. The sampling intervals are chosen with respect to the specific intention in the experiment design. To simulate the amplitude parameters A_i, a *dominant amplitude* A_1 is sampled. The next amplitudes are calculated considering an exponential decay with a fixed rate *dec*:

$$A_i = A_1 \exp(-i \cdot dec), \quad i = 1, ..., \tilde{n}.$$

This makes the first frequency i.e. ω_1 to be the dominant frequency of the Fourier series. Throughout the experiments, all time series were generated with an equal length of 300 time steps. i.e. $T = 300$. A detailed overview of the parameters chosen for the simulation can be found in table B.2 in appendix B.

For assigning class labels to the time series samples, we consider the following two scenarios.

Scenario 1: *Label based on the presence of a shapelet*

Analogously to salient region detection in image classification, common explainability methods highlight time steps identified as important for the classification outcome. In general, saliency is defined as the effect each input feature has on the prediction outcome, measured in its simplest form as the gradient of the output with respect to each input feature multiplied by the input feature itself. In image classification,

input features correspond to pixels of the image, whereas in the time series setting, input features correspond to time steps. The saliency assignment of each input feature follows the same logic for both settings. If a certain shape occurring in the time series is responsible for the class label, these location-based explainability methods are expected to be able to correctly identify the time interval in which the shape is occurring. Shapelets [103] represent sub-sequences of time series which maximally explain the split of the data set into two classes, in the sense that the difference of entropy before and after splitting is maximal. The entropy of a discrete random variable U with probability distribution P_U and alphabet $\mathcal{U}$ is defined as

$$H(U) = -\sum_{u \in \mathcal{U}} P_U(u) log(P_U(u)),$$

where $0ln(0) := 0$ [35]. In application to a time series data set D consisting of two classes A and B, the empirical entropy of D is defined as

$$H(D) = -p(A)log(p(A)) - p(B)log(p(B))$$

where $p(A)$ and $p(B)$ represent the proportions of class A and class B in the data set respectively [103]. In a binary classification setting, the data set D is split into two subsets D_0 and D_1, each subset consisting of samples predicted as belonging to the same class. Finding a shapelet can be interpreted as finding an optimal splitting strategy for D into D_0 and D_1. The splitting rule is given by a distance measure d between a specific sub-sequence, also refered to as shapelet candidate Sh and any other sub-sequence of samples in D as well as a splitting threshold d_{th}. The learned parameters maximize the difference in entropy of D and the the combined weighted entropy of $D_{0_{Sh}}$ and $D_{1_{Sh}}$, such that for all sub-sequences $X_{0,i} \in D_{0_{Sh}}$ $d(X_{0,i}, Sh) < d_{th}$ and for all sub-sequences $X_{1,i} \in D_{1_{Sh}}$ $d(X_{1,i}, Sh) \geq d_{th}$. A shapelet then defines the sub-sequence of a sample in D, such that

$$\begin{aligned} &H(D) - (f(D_{0_{\text{shapelet}}})D_{0_{\text{shapelet}}} + f(D_{1_{\text{shapelet}}})D_{1_{\text{shapelet}}}) \\ \geq &H(D) - (p(D_{0_{Sh}})D_{0_{Sh}} + p(D_{1_{Sh}})D_{1_{Sh}}) \end{aligned}$$

for all other sub-sequences Sh in D, where $p(.)$ represents the fraction of samples of D assigned to the respective subset [103].

For assigning shape-based labels to the time series, a shapelet is inserted at a random or fixed position into all time series $X \in D$ belonging to one class. The shapelet is a second simulated Fourier series of length $l \leq T$, where $l = \text{window-ratio} \cdot T$ for a chosen window ratio. We define the sampling intervals for the latent features of the shapelet to be non-intersecting with the sampling intervals of the latent features of the

original time series X. The resulting shapelet replaces the original time series in the interval $[j, j + l]$, where

$$j \sim \mathcal{U}(1, T - l).$$

Scenario 2: *Label based on differences in the latent features*
Following the investigation of the effectiveness of explainability methods for latent features, we introduce a second simulation scenario where the labels depend on a difference in the sampling distribution of latent features of the time series. This scenario highlights the main focus of this project, and represents our novel view of explainability methods for time series. Similar to the first scenario, the time series are sampled as discretized Fourier series with latent variables ω, A and ϕ.

Based on the data generation method described above, we design ten different mechanisms. In four experiments, the label is based on a shapelet at random and fixed positions in the start, middle and end of the time series respectively. Each two data sets are designed such that the label is based on one of the latent Fourier concepts. For the scope of this thesis we design the data sets to only include one label-making feature at a time. An overview of the generated data sets is provided in table 3.1. Details about the parameters of the generated data sets and the label generation per experiment can be found in tables B.1 and B.2 in appendix B.

3.2 Compared classifiers and explainability methods

In Chapter 2, two of the most common families of time series classifiers as well as multiple explainability methods were introduced. During this project, explainability on the following six classifiers was investigated:

- Long Short Term Memory (LSTM):
  The probably most renown deep learning method for time series forecasting and classification is included in the analysis as a baseline classifier. Since the classification performance of this method is not expected to surpass the LSTM-based classifiers listed below, we are mainly interested in the comparison to the standard one-dimensional CNN.

- LSTM trained via Saliency Guided Training (LSTM + SGT):
  Ismail et al. prove that the LSTM as well as other recurrent neural networks suffer from the vanishing saliency problem. One possible approach to addressing this problem is the saliency guided training procedure. In [41] the method is only evaluated on image data treated as multivariate time series, which does

Experiment	Label	Idea
3, 4, 5 & 6	Shapelet	Due to the vanishing saliency problem and the LSTMs' difficulties learning long-term dependencies, we expect the classifiers and explainability methods to achieve different performances depending on the position of the shapelet.
101 & 102	Frequency	The experiments differ only in the number of sines employed for sampling the Fourier series. We aim to test for performance differences of the classifiers and explainability methods on very simple and more complex Fourier series.
103 & 104	Phase shift	As for the frequency experiments, these data sets only differ in the number of sines. The idea behind the generation is the same as above
105 & 106	Amplitude	Following the same idea as in the experiments above, the data sets only differ in the number of sines.

Table 3.1: Description of data sets.

not represent real world time series data. We intend to investigate the saliency and classification performance improvement through this interpretable training procedure on more realistic univariate data sets.

- Input-cell attention LSTM (AttentionLSTM):
  A second way of addressing the vanishing saliency problem is the combination of an input-cell attention mechanism with the recurrent neural network. We intent to investigate the difference in classification and saliency performance between the two solutions proposed in [41] and [43]. By conducting post-hoc explainability experiments on this classifier, we further hope to contribute to the ongoing discussion of whether attention is explanation.

- One-dimensional Convolutional Neural Network (CNN):
  The counterpart to the standard LSTM from the class of convolutional neural networks is evaluated against the generic LSTM regarding classification performance and explainability. It further acts as a baseline for the other investigated convolutional architectures.

- One-dimensional CNN trained via Saliency Guided Training (CNN + SGT):
  The saliency guided training procedure was introduced to provide less noisy gradients and thus improve the saliency performance of gradient-based explainability methods. The interest in including a one-dimensional convolutional neural

network trained via saliency guided training into the analysis lies in the evaluation of the of saliency performance on classifiers different from RNNs employing this procedure.

- Temporal Convolutional Network (TCN):
  The temporal convolutional network was designed to inherit properties from CNNs and RNNs beneficial for sequence modeling. In [13] the authors show that TCNs generally outperform RNNs on sequence modeling tasks and conclude that convolutional neural networks, especially the TCN, should be considered the leading deep network architecture for sequential tasks. We explore the difference in performance of explainability methods on a generic CNN architecture and a temporal convolutional network architecture.

Descriptions of the classifiers were provided in Section 2.2. Architecture details are described in table B.3 in appendix B.
Since ante-hoc explainability methods are integrated in the network itself and thus cannot easily be compared across different network architectures, we focus on post-hoc explainability methods in this project. A single ante-hoc method, the input-cell-attention mechanism, described in Section 2.2.2, is included in the analysis. Specifically, each one gradient based and perturbation based feature attribution method as well as one counterfactual explanation method is chosen to represent different classes of post-hoc explainability methods in the comparison:

- Integrated Gradients (IG):
  The gradient-based attribution method Integrated Gradients, introduced in Section 2.1.2, satisfies the three desiderata of *sensitivity*, *implementation invariance* and *completeness*, as outlined in Section 2.1.2. The Integrated Gradient of an input x defined as

$$IG_i(x) := (x_i - x_i') \times \int_0^1 \frac{\partial f(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha \tag{3.1}$$

  for input dimension i, where $f : \mathbb{R}^n \to [0, 1]$ represents a deep neural network and $x, x' \in \mathbb{R}^n$ represent input and baseline inputs respectively. By fulfilling the stated desiderata, IG is a superior attribution method compared to many other gradient-based methods, while remaining relatively simple and computationally effective, making it our gradient-based attribution method of choice.

- Kernel SHapley Additive exPlanation Values (SHAP):
  Shapley values of the conditional expectation function of the model of interest have been shown to fulfill multiple desirable properties. Since exact computation of these values is challenging, the authors propose an alternative method by combining linear LIME (see section 2.1.2) and the computation of Shapley values

to approximate the Shapley values of the conditional expectation function. Let f represent a prediction model and F the set of all features. The Shapley value v_i for feature x_i of input X is defined as

$$v_{x_i}(f, X) = \sum_{S \subseteq F \setminus \{x_i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{x_i\}}(x_{S \cup \{x_i\}}) - f_S(x_S)].$$

Linear LIME optimizes the loss function $L(f, g, \pi_{x'})$ over all linear explanation models g for f using simplified inputs x', which are mapped to the original inputs by a mapping function $h_x(x') = x$. Let M represent the number of simplified input features. A linear surrogate model g to explain the model f is defined as a mapping of binary variables $z' \in \{0, 1\}^M$ of the form

$$g(z') = v_0 + \sum_{i=1}^{M} v_i z'_i$$

for $v_i \in \mathbb{R}$. The local explainability method linear LIME ensures that $g(z') \approx f(h_x(z'))$ for all binary variables $z' \approx x'$. Let B be the set of non-zero indices of the binary simplified input z' and $h_x(z') = z_B$ the simplified input mapping. By approximating $f(z_B)$ through

$$f_x(z') = f(h_x(z')) := \mathbb{E}[f(z)|z_B]$$

and defining a loss function L for the linear approximation g of prediction model f as

$$L(f, g, \pi) = \sum_{z' \in \{0,1\}^M} [f(h_x^{-1}(z')) - g(z')]^2 \pi(z'),$$

with Shapley weighting kernel

$$\pi(z') = \frac{(M-1)}{\binom{M}{|z'|} |z'| (M - |z'|)},$$

where $|z'|$ describes the number of non-zero values in z', linear LIME minimizes the loss function over all possible linear surrogate models g, calculating the Shapely values via a weighted linear regression.

We employ the presented perturbation-based explainability technique due to its computational superiority compared to many other model-agnostic approaches and its popularity in the literature.

- Native Guide (NG):
  The counterfactual explanation method described in Section 2.1.2, finds the nearest unlike neighbor (NUN) of the time series instance of interest in the training data set, which afterwards is perturbed towards the decision boundary. A NUN is any time series from the database classified as a different class than the instance of interest, which minimizes the distance to the to-be-explained sample based on some pre-defined distance measure. Commonly this measure is chosen to be the Dynamic Time Warping (DTW) distance. Dynamic Time Warping aims to find an optimal alignment of two time series X, Y of lengths L_X, L_Y in the temporal domain which minimizes the accumulated cost

  $$D_{i,j} = f(x_i, y_j) + \min\{D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1}\},$$

  where $i = 1, ..., L_X$, $j = 1, ..., L_Y$ and initialization $D_{0,0} = 0$, $D_{i,j} = \infty$ otherwise. The overall cost D_{L_X,L_Y} is commonly chosen to represent the DTW dissimilarity measure [84]. We employ two different variations for perturbing the NUN towards the to-be-explained instance. First, the saliency output vectors of the two above mentioned feature attribution methods Kernel SHAP and Integrated Gradients are used to perturb only important time steps providing sparse explanations.

  In the second approach, the complete time series is perturbed using weighted dynamic barycenter averaging [32], a special form of DTW barycenter averaging (DBA) [72]. Let $D = \{(Y_1, w_1), ..., (Y_n, w_n)\}$ be a data set of time series Y_i in a dynamic time warping distance induced space Z with corresponding weights w_i, $i = 1, ..., n$. Then, the weighted average time series $\bar{Y}$ is calculated as

  $$\arg\min_{\bar{Y} \in Z} \sum_{i=1}^{n} w_i DTW^2(\bar{Y}, Y_i).$$

  The function above can be minimized with the help of DBA in which a starting average $\bar{Y}$ is iteratively adapted using an expectation-maximization approach. For a detailed description of the DBA and weighted dynamic barycenter averaging algorithm, refer to [32, 72].

  The comparatively simple yet intuitive counterfactual generation method of Native Guide especially suites our purposes, since it allows to combine counterfactual explanations with the other above mentioned post-hoc feature attribution methods, additionally aiding in interpreting the feature attribution results.

3.3 Implementation details

Since the main goal of this project was not to analyse the classification performance of different network architectures, but to investigate the performance of explainability

methods when the class labels depend on latent input features rather than positional information, all classifiers only consisted of one-layer networks. Furthermore, we do not employ dropout or any other form of additional regularization. By keeping the architecture simple, we intend to objectively evaluate and compare the explainability methods for certain architectures without the influence of optional variations, preventing overfitting or boosting the network performance. A detailed description of the network architectures can be found in table B.3 in appendix B.
All algorithms were implemented in the Python programming language. The classifiers were implemented using the deep learning library *PyTorch* [71] with the help of the wrapper *PyTorch Lightning* [29]. The authors of [43] provide a publicly available implementation of the input-cell-attention module which we adapted. Hyperparameter optimization was performed through the library *Optuna* [1]. For the feature attribution techniques, the implementations from the *PyTorch* based model interpretability library *Captum* [57] were employed. The other investigated classifiers and explainability methods had to be manually implemented[1].

[1]All rights for the code of this thesis belong to the Department Reasoned AI Decisions of the Fraunhofer Institute of Cognitive Systems.

Chapter 4

Latent feature saliency

This chapter introduces a new framework for extending current explainability methods providing timestep-wise importance scores to the latent space of a time series. First, the general idea and the intuition behind the approach is presented. Afterwards, the derivation of the performed calculation is stated. In Section 4.3 we present a straightforward statistical baseline approach to global explanation of latent feature saliency.

4.1 Description of the proposed method

Our goal is to develop an extension of those explainability methods which assign importance scores to input time steps. We aim to define a procedure that maps the time-step-wise scores to one overall importance score for the latent Fourier concepts frequency, amplitude and phase shift. The general importance score for the Fourier concepts is motivated by easy interpretability and wide use of Fourier series modelling in time series analysis. We define the mapping such that a high value of the map indicates a high likelihood that the latent Fourier concepts were responsible for the class label. Thus, a high score expresses relative importance of the Fourier concepts. On the other hand, low values signify either the importance of some latent time series features other than the Fourier concepts or the presence of a certain shape in the time series which was responsible for the predicted class label. We formalize importance of the latent Fourier concepts as follows: If we observe a significant discrepancy in distribution of at least one of the latent concepts between the classes, we consider the Fourier concepts to be important. We hypothesize that information about this distribution shift is more valuable in real-world applications than information about single latent features considered important by the method. In case the distributions for the different classes are non-overlapping, importance scores for single latent features could be informative as well. Nevertheless, for universal applicability we decide to only focus on a unified overall importance score for all latent concepts.

We assume that the underlying time series generation processes are continuous, periodic and stationary. By the Fourier theorem a trajectory from such processes can be

represented by means of its Fourier series [15] of the form $X = (x_1, ..., x_T)$, where for every $t = 1, ..., T$

$$x_t = a_0 + \sum_{n=1}^{\infty} A_n \cos(\omega_n t + \phi_n).$$

We note that in the experiment setting we consider discrete observations from these trajectories for time $t = 1, ..., T$. As a result, the set of latent features of the time series consist of frequencies ω_n, amplitudes A_n and phase-shifts ϕ_n, $n = 1, ..., \infty$. From a discrete realization of the time series we can numerically recover the latent features for $n = 1, ..., \tilde{T}$ through Fast Fourier Transform (FFT), where $\tilde{T} = \frac{T}{2}$ for T even or $\tilde{T} = \frac{T-1}{2}$ for T odd.
Timestep-wise explainability methods constitute a mapping from each input feature to an importance score. We define a function $\mathcal{I}$ which maps the latent features to the final timestep-wise importance scores through *i)* resampling the time series X by a Fourier transform and *ii)* applying the explanation function to the resampled input. Therefore, the problem of determining a mapping from importance scores provided by some explainability method to a latent feature importance score can be redefined as an explainability problem of function $\mathcal{I}$.

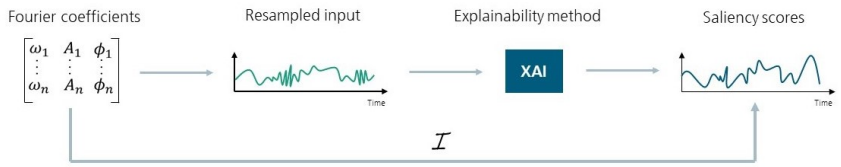


Figure 4.1: Outline of function $\mathcal{I}$.

As stated in Chapter 2.1, several methods exist for assigning input importance scores. In our setting we face a slightly different explainability problem: Instead of assigning saliency scores to multiple input features regarding the prediction of one single outcome, we are interested in quantifying the overall importance of a function of all input features (in our case, the Fourier transform) with regard to multiple related outputs in one score p. Thus it is not possible to directly apply common explainability methods. The input to our saliency method is a vector of T time-step-wise importance scores, which again constitutes a time series. Applying a standard saliency method as Gradient $\times$ Input to this time series with respect to each Fourier coefficient results in one saliency vector per latent feature. We hypothesize that if the original explanation is related to the latent space of a time series, then the importance vector should be highly similar to the saliency vector of the latent concepts, in the sense that the distance between the original importance vector and the saliency vectors of function $\mathcal{I}$ with

respect to one latent concept at a time should be small. Quantifying this distance is a non-trivial task. Various distance and similarity measures are commonly applied in the time series domain [69, 84]. Nevertheless, all of the existing measures suffer from different types of limitations, such that none of the measures on its own is suitable to our problem. One example of a commonly employed similarity measure for time series is the Pearson correlation coefficient. This measure assigns high values to series sharing the same pattern in time. If two highly similar signals are slightly shifted in the time dimension, the measure will fail to assign a high similarity score. Since we are interested in mapping the explanations of one single time series sample at a time to the latent space, it is not possible to perform metric learning or to learn a combination of various difference measures to overcome their respective limitations. Therefore, we need to quantify the distance between the two vectors in a different way.
In a first step, the original importance score vector and all saliency vectors of the latent concepts are transformed by a softmax function. Applying the softmax function to a vector can be interpreted as assigning an importance probability to each entry. From a game-theoretic point of view, the output of the softmax function represents a players mixed strategy for the game [33]. A mixed strategy assigns a probability of being played to each of a players options. Anderson, Goeree, and Holt [5] derive the softmax-based *logit equilibrium* which accounts for random perturbation of the payoff resulting from a players mixed strategy. We hypothesize that this property permits the detection of a possible effect of the phase shift. The problem thus breaks down to comparing the similarity of mixed strategies of two players, where the game is given by the importance allocation task. To investigate the similarity of the strategies, we employ the euclidean distance.

In this thesis an extension of the input-cell attention LSTM is developed. The decision for providing an extension of the adapted LSTM classifier with the attention mechanism as explainability method is based on the convincing performance of the classifier-explainability pair. Results supporting this statement are presented in Section 5. Nevertheless, the provided approach is only a special case. Our framework can easily be adapted to other explainability methods which provide importance scores for every input feature. An overview of our latent saliency framework is provided in Figure 4.2.

4.2 Derivation of calculations

The attention mechanism in [43] represents a multidimensional importance embedding of the input time series. The attention matrix for input $X = (x_1, ..., x_T)$ is calculated

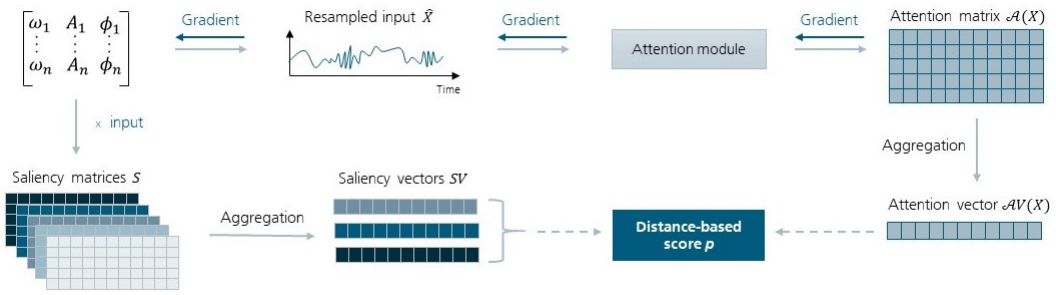


Figure 4.2: Outline of the latent feature explainability framework.

as

$$\mathcal{A} = \text{softmax}(W_2 \tanh(W_1 X^T)),$$

where W_1 and W_2 are weight matrices with dimension $d_a \times N$ and $r \times d_a$ respectively. The number of time steps the network attends to is represented by r, refered to as attention hops, d_a is a hyper-parameter. In the case of univariate time series classification where $N = 1$, the attention matrix has the dimension $r \times T$.

For mapping the time-step-wise attention scores to a latent feature-based importance score, we apply a post-hoc saliency method on the raw attention matrix: calculating the gradient with respect to the latent features multiplied by the feature itself. Although the gradient is often applied as a saliency method on its own, it does not measure importance of an input directly, but the sensitivity of the output with respect to a change in the input. The multiplication of the gradient with the input results in the marginal effect of the input on the outcome, hence representing the importance.

4.2.1 Gradient calculation

For every $i = 1, ..., r$ and $t = 1, ..., T$ the entry $\tilde{a}_{i,t}$ in the raw attention matrix before the softmax is applied is calculated as

$$\begin{aligned}\tilde{a}_{i,t} &= \sum_{k=1}^{d_a} w_{2_{i,k}} \tanh(w_{1_k}[\sum_{n=1}^{\tilde{T}} A_n \cos(w_n t + \phi_n)]) \\ &= \sum_{k=1}^{d_a} w_{2_{i,k}} \tanh(w_{1_k} \hat{x}_t).\end{aligned}$$

Let l represent Fourier coefficient. The gradient from entry $\tilde{a}_{i,t}$ in the raw attention

matrix with respect to l can be calculated as

$$\frac{\partial \tilde{a}_{i,t}}{\partial l} = \frac{\partial \tilde{a}_{i,t}}{\partial \hat{x}_t}\frac{\partial \hat{x}_t}{\partial l} \quad \text{where} \quad \frac{\partial \tilde{a}_{i,t}}{\partial \hat{x}_t} = \sum_{k=1}^{d_a} w_{2_{i,k}} w_{1_k}(1 - \tanh^2(w_{1_k}\hat{x}_t)) =: \Delta_{i,t}.$$

The gradients of the the raw attention values with respect to the latent features A_n, ω_n, ϕ_n for all $n = 1, .., \tilde{T}$ respectively are

$$\begin{aligned}
\frac{\partial \tilde{a}_{i,t}}{\partial A_n} &= \Delta_{i,t} \cdot \cos(\omega_n t + \phi_n), \\
\frac{\partial \tilde{a}_{i,t}}{\partial \omega_n} &= \Delta_{i,t} \cdot A_n t \cdot (-\sin(\omega_n t + \phi_n)), \\
\frac{\partial \tilde{a}_{i,t}}{\partial \phi_n} &= \Delta_{i,t} \cdot A_n \cdot (-\sin(\omega_n t + \phi_n)).
\end{aligned}$$

For the saliency values $s_{\tilde{a}_{i,t}}$ it follows

$$\begin{aligned}
s_{\tilde{a}_{i,t}}(A_n) &= \Delta_{i,t} \cdot A_n \cdot \cos(\omega_n t + \phi_n), \\
s_{\tilde{a}_{i,t}}(\omega_n) &= \Delta_{i,t} \cdot A_n t \cdot \omega_n \cdot (-\sin(\omega_n t + \phi_n)), \\
s_{\tilde{a}_{i,t}}(\phi_n) &= \Delta_{i,t} \cdot A_n \cdot \phi_n \cdot (-\sin(\omega_n t + \phi_n)).
\end{aligned}$$

Since saliency scores are calculated for every latent feature at every time point and for every attention embedding $i \leq r$, the computation results in $3\tilde{T}$ matrices of dimension $r \times T$, where each matrix corresponds to the saliency values with respect to one specific latent feature. We refer to these matrices as saliency matrix $S(l_n)$ for $l \in \{A, \omega, \phi\}$ and $n = 1, ..., \tilde{T}$.

4.2.2 Aggregation of saliency scores

The scores are aggregated in two steps. First, we reduce dimension one of each saliency matrix S. For aggregating the r saliency values per latent variable for every time step $t = 1, .., T$, we adapt the proposed aggregation of the input embedding across the rows of the attention matrix in [63]: Instead of building the column sums and normalizing the resulting row vector, we sum the attention scores across the columns and apply the softmax function. This procedure results in one saliency vector $SV(l_n)$ per latent feature l_n, $l \in \{A, \omega, \phi\}$ and $n = 1, ..., \tilde{T}$

$$SV(l_n) = \text{softmax}((\sum_{i=1}^{r} s_{\tilde{a}_{i,1}}, ..., \sum_{i=1}^{r} s_{\tilde{a}_{i,T}})).$$

The same aggregation is performed on the raw attention matrix, resulting in the raw attention vector $\mathcal{A}V$,

$$\mathcal{A}V = \text{softmax}((\sum_{i=1}^{r} s_{\tilde{a}_{i,1}}, ..., \sum_{i=1}^{r} s_{\tilde{a}_{i,T}})).$$

This attention vector can be interpreted as a vector of importance weights $w_t \in [0, 1]$ for each time step $t = 1, ..., T$, where $\sum_{t=1}^{T} w_t = 1$.
In step two, all saliency vectors belonging to the same concept are averaged, resulting in one overall saliency vector per concept

$$SV(l) = \frac{1}{\tilde{T}} \sum_{n=1}^{\tilde{T}} SV(l_n), \quad l \in \{A, \omega, \phi\},$$

which again can be interpreted as a vector of importance weights $w'_t \in [0, 1]$, where $\sum_{t=1}^{T} w'_t = 1$.

4.2.3 Calculation of distance-based importance score

The aggregation step results in four importance vectors; the attention vector and one saliency vector for each Fourier concept. The overall goal is to quantify the likelihood of the decision for the prediction being based on the Fourier concepts. We choose to assign a score $p \in [0, 1]$ to the importance scores provided by the attention mechanism, relating the attention output to the latent Fourier concepts. A high score implies a high likelihood of the explanation indicating a difference in the sampling distribution of at least one of the Fourier concepts between the two classes, whereas a low score signifies that the reason for the prediction decision cannot be related to the Fourier concepts.
First, the similarity between the importance embedding provided by the attention vector and each embedding presented by the different saliency vectors is quantified in terms of the euclidean distance d,

$$d(l) = \|\mathcal{A}V - SV(l)\|_2, \quad l \in \{A, \omega, \phi\}.$$

Since both vectors sum up to one, for the maximum euclidean distance between the vectors we have

$$d(l) \leq 2 \quad \forall l.$$

Thus, a distance of $d = 2$ implies complete unalignment of the importance embeddings, whereas $d = 0$ signifies that vectors are identical. We argue that the relation between

the euclidean distance and similarity between the two saliency vectors is not linear. Importance embeddings with a distance of one should not be interpreted as similar in the sense that the vectors do not provide approximately the same explanations.
We choose to transform the distances d via a negative exponential transformation, thus penalizing higher distances. This closely matches the intuition of similarity between the importance embeddings. Furthermore, this assumption naturally provides the required limit

$$d(l) = 0 \Rightarrow s(l) := \exp(-d(l)) = 1,$$

stating that the explanation given by the attention mechanism can completely be traced back to the importance of the latent Fourier concept if the embeddings are entirely alike, where $s(l)$ represents the overall *saliency score* for concept l. Nevertheless, the requirement

$$d(l) = 2 \Rightarrow s(l) = 0$$

is not fulfilled by the stated assumption, since $\exp(-2) \neq 0$. Thus, the negative exponential transformation function needs to be adapted to fulfill this requirement as illustrated in Figure 4.3.

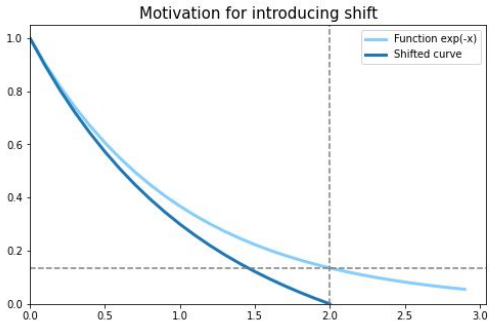


Figure 4.3: Comparison of exponential density function and shifted function $s(l)$.

To ensure $s(2) = 0$, a shift is introduced to the score calculation, resulting in the final formula

$$s(l) := \exp(-d(l)) - \frac{d(l)}{2}\exp(-2).$$

Although the definition of $s(l)$ does not state a probability, the adapted transformation benefits from easy interpretability. To assign an overall score $p \in [0, 1]$ for the importance of the latent Fourier concepts, the three latent saliency scores are averaged

$$p = \frac{1}{3}(s(A) + s(\omega) + s(\phi)). \tag{4.1}$$

The procedure of the method can be summarized as follows.

1. Retrieve Fourier coefficients from the original time series
2. Apply saliency method (Gradient $\times$ Input) to output vector/matrix with respect to all Fourier coefficients.
3. If original explainability output has more than one dimension, perform aggregation step of original explainability output and each latent saliency output separately. In any case, aggregate saliency vectors belonging to the same Fourier concept.
4. Apply softmax function to resulting original explanation vector and the three concept-wise saliency vectors separately.
5. Measure euclidean distance d between the original explanation vector and the concept-wise saliency vectors.
6. Average resulting similarity scores $s(A), s(\omega), s(\phi)$ to obtain final likelihood p of the explanation being based on the latent feature, where

$$s(l) := \exp(-d(l)) - \frac{d(l)}{2}\exp(-2), \quad l \in \{A, \omega, \phi\}.$$

4.3 Global approach: Logistic regression

Designing global saliency methods is far more challenging than the development of a local explainability method. Averaging the explanations provided by local methods for all samples in a data set does not necessarily yield useful and accurate results, due to the underlying structure of the dataset.
As a straightforward global approach, we propose to fit a logistic regression model on the latent features of the test data set to the predicted class label. The resulting coefficients can be interpreted as indicators for the effect each single latent feature had on the classification outcome.

The algorithm for calculating global importance scores based on logistic regression proceeds as follows.

1. Calculate Fourier coefficients for each sample in the test data set.
2. Choose k combinations (A_n, ω_n, ϕ_n), $n = [1, \ldots, \tilde{T}]$ per sample for which A_n belongs to the highest k amplitudes present in the respective sample. The parameter k can be determined through resampling the input time series starting from only one combination of coefficients and iteratively increasing the number of combinations while observing the resampling loss. Choose k as the number of combinations for which the loss curve levels off.

3. Interpreting these k combinations per sample as the predictors, fit logistic regression model with the predicted label $\hat{y}$ as the dependent variable

$$\frac{p(\hat{y}=1)}{p(\hat{y}=0)} = \exp(\beta_0+\beta_1 a_0 + \beta_2 A_1 + \beta_3 \omega_1 + \beta_4 \phi_1 + \\ ... + \beta_{3k-1} A_k + \beta_{3k} \omega_k + \beta_{3k+1} \phi_k)$$

4. The absolute value of the sum of coefficients of $\beta = (\beta_0, ..., \beta_{3k+1})$ belonging to the same concept represents the final importance score. The coefficient β_0 is interpreted as the relevance of aspects which cannot be related to the latent Fourier concepts, as for example the relevance of positional features in the time series.

This approach can be employed as a baseline for comparing our local latent feature explainability method and possible also future methods against. Furthermore, employing a logistic regression approach on the latent features can provide a good starting point for future research of global latent saliency methods. The benefit of a logistic regression model is its easy interpretability and universal applicability.

Chapter 5

Results and Discussion

5.1 Results

The conducted experiments were twofold. In a first step, the performance of various time series classifiers on multiple data sets was assessed. One ante-hoc and three post-hoc explainability methods were employed to explain the classification outcome. The results of the experiments provided in this section underline the need of a latent feature saliency detection method. In the second step, the proposed global and local latent feature saliency methods were applied to extend the the explanations provided by the input-cell attention mechanism to the latent space of the time series. Results of the conducted experiments categorized by classification performance, explainability method evaluation and latent feature saliency detection are presented in the following.

5.1.1 Results of comparison of classification performance

In binary classification, the performance is commonly assessed through a combination of multiple performance measures and metrics. Accuracy, the fraction of the number of correctly classified samples and the sample size, is the most employed performance metric. Although maximizing accuracy is in many cases seen as the primary goal in a classification task, only providing accuracy as a measure of performance can be misleading due to the accuracy paradox, i.e. in the presence of a class-imbalanced data set. High accuracy does not always imply good performance [97]. Thus, it it necessary to determine sufficiently many evaluation metrics suitable to the classification task and optimization goal to objectively evaluate and compare classification performances [19]. Four basic performance measures can be directly inferred from the classification output; the number of correctly positive classified samples (TP), the number of correctly negative classified samples (TN) as well as the number of positive and negative misclassified samples (FP/FN) respectively. Combined, these measures are commonly depicted as a confusion matrix or contingency table [19]. Multiple performance metrics can be directly calculated from these basic measures. The true positive rate (TPR), as well referred to as sensitivity or recall, depicts the ratio of TP to the number of positive samples $P = TP + FN$. In the same manner, the true negative rate (TNR) represents

the ratio of TN to the number of negative samples. Precision states the relation of true positive samples to all samples classified as positive [27]. Which metric should be considered more important strongly depends on the use case. In the medical domain, high recall is often more important than high precision, since falsely classifying a sick person as healthy might lead to life-threatening situations. The F1 score states the harmonic mean of recall and precision, thus being insensitive to TN [19]. In receiver operating characteristic (ROC) analysis the trade-off between TPR and TNR is investigated, which is commonly visualized by plotting TPR against FPR [27] The area under the ROC curve ($AUROC/AUC$) is considered to be the most informative metric to represent the performance in binary classification tasks [19]. The AUC only considers the rank of the scores, ignoring the magnitude [27].

We evaluated the performance of the classifiers on the test data set using the metrics accuracy, precision, recall, F1 score and $AUROC$. Table 5.1 depicts the average performance of the evaluated classifiers across all data sets. A description of the data sets was provided in Section 3.1. More details about the data sets can be found in Section B.1 in Appendix B. The evaluation metric results per experiments are as well provided in Appendix B.

Classifier	Accuracy	Precision	Recall	F1	AUROC
LSTM	0.8398	0.8258	0.8570	0.8364	0.8477
LSTM + SGT	0.8016	0.7854	0.8094	0.7865	0.8243
AttentionLSTM	**0.9383**	0.9103	0.9789	0.9422	0.9616
CNN	0.8332	0.8270	0.9875	0.8823	0.9370
TCN	0.8863	0.8821	**0.9922**	0.9201	0.8948
CNN + SGT	0.9340	**0.9311**	0.9883	**0.9508**	**0.9681**

Table 5.1: Average classification performance on test data across all data sets.

Overall, the CNN combined with SGT achieved the highest performance, whereas the LSTM and the LSTM combined with SGT where not able to compete with the other classifiers. On average, the CNN-based architectures had a stronger performance than the LSTM-based models. The low results of the LSTM might be related to its difficulties learning long-term dependencies as necessary for experiment three (shapelet at random position) and the experiments in which the label was based on a difference in the phase shift. Supporting results can be found in tables B.4, B.10 and B.11. The saliency guided training procedure did not improve the average performance of the LSTM, whereas it consistently improved the average performance of the CNN. In contrast, the input-cell attention mechanism strongly improved the classification performance of the LSTM, making the model competitive with the CNN-based architectures. The TCN, especially designed for application in the time series domain, achieved a heightened performance

in comparison to the standard CNN, even outperforming the other classifiers on the experiments which test the detection of the phase shift as the label-making feature. Details can be found in tables B.10 and B.11 in appendix B.
The classification performance is further visualized in Figure 5.1 depicting the ROC curves as well as *AUROC* values of the different classifiers across all experiments. The experiment numbers are ordered corresponding to the respective label-making feature. In experiments three to six, a shapelet is present in class one whereas it is missing in class zero. The label in experiments 101 and 102 is based on a difference in the sampling intervals of the frequency. In experiments 103 and 104, the phase shift is responsible for the class label and in experiments 105 and 106 the label-making feature is the amplitude. More detailed descriptions about the experiments can be found in table B.2 and B.1 in appendix B.1.

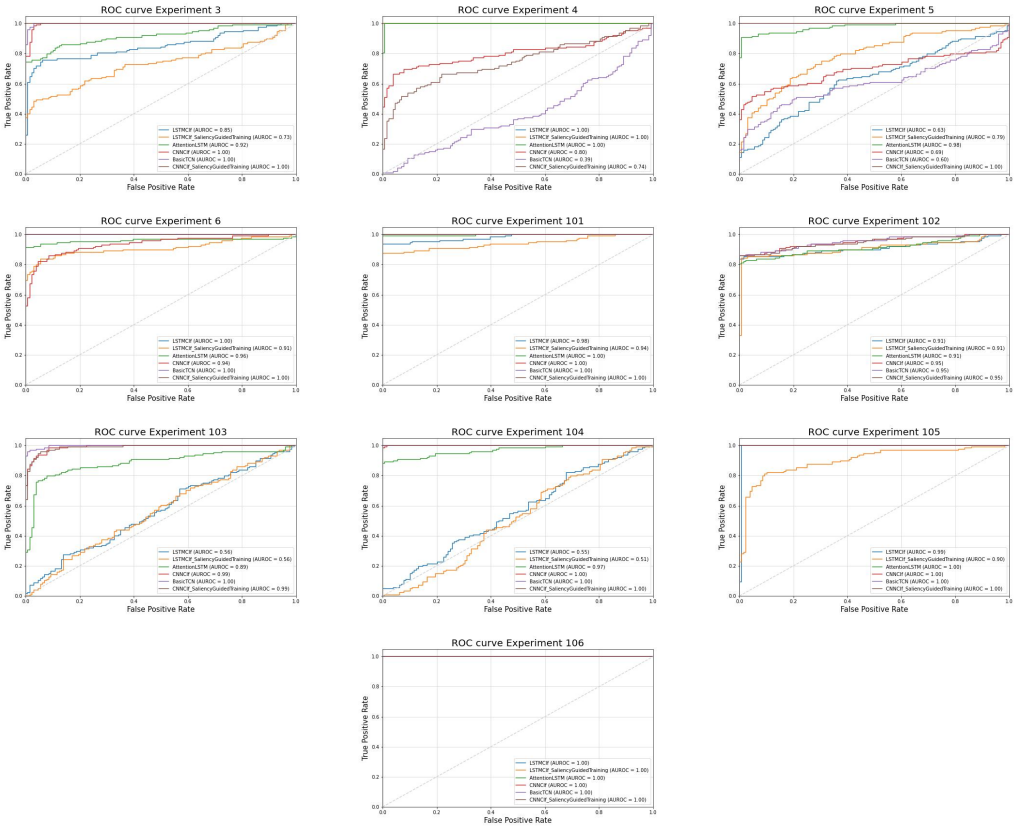


Figure 5.1: ROC curves of evaluated classifiers across all experiments

Overall, the classifier achieved better results on the experiments in which the label was

assigned based on a latent Fourier concept. Scenarios in which the label is based on frequency or amplitude were especially easy to classify. The experiments focusing on the phase shift posed a challenge to the LSTM and the LSTM trained through saliency guided training. All other classifiers could successfully identify the class-differences. Differences in the classification performance of the classifiers depending on the number of sines of the Fourier series could not be observed.
The classifiers showed a very diverse performance on the shapelet experiments. If the shapelet was placed at a random position in the time series, the achievements of the models were similar to performance on the Fourier concept-based data sets. The performance on the data sets, in which the label depends on a shapelet at a fixed position, is very different from all other results as well as inconsistent across the three experiments. In some cases, the standard LSTM model is able to correctly assign class labels to the entire test data set, whereas multiple CNN-based classifiers do not perform better than random guessing, in the sense that these classifiers assign the same label to all samples. We hypothesize, that by imputing a shapelet at a fixed position, a bias was introduced into the data set. As a result, the classifiers learned to focus on the respective time step instead of focusing on the shapelet. For the detailed results of the classification metrics we refer to Appendix B.

5.1.2 Results of comparison of explainability methods

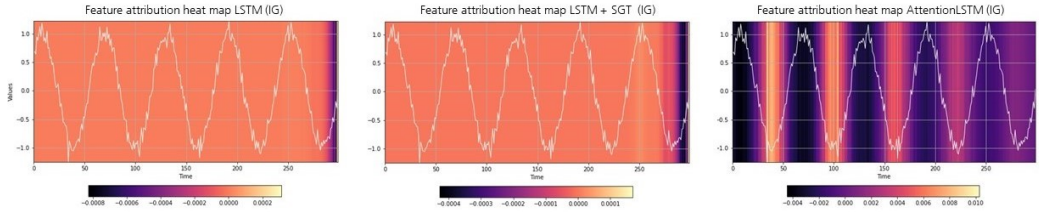
For assessing the performance of an explainability method, no omni-applicable quantitative metric has been introduced in the literature yet. This is partly due to the fact that there does not exist a universal definition of the properties which an explainability technique must fulfill. The most common way of evaluating an explainability method is by measuring the faithfulness of the method. Faithfulness is fulfilled if the saliency scores assigned to each input feature truly indicate importance [61]. This commonly is assessed through the drop or increase in accuracy represented by the area under or over the perturbation curve respectively, after important features are removed from an input sample or important features are added to an uninformative baseline sample [73, 80]. Nevertheless, there exist much criticism of these measures of faithfulness [61].
Since we conduct experiments on synthetic data, we guarantee a supervised evaluation setting for the explainability methods. Therefore, we do not measure the performance of the tested methods based on a quantitative metric, but only visually evaluate the methods performance.

We intend to visually compare the results provided by the post-hoc feature attribution methods IG and SHAP, as well as the attention scores assigned by the input-cell attention mechanism through an importance heat map overlaid by the original time series. This visualization method allows us to directly assess the relevance of each input

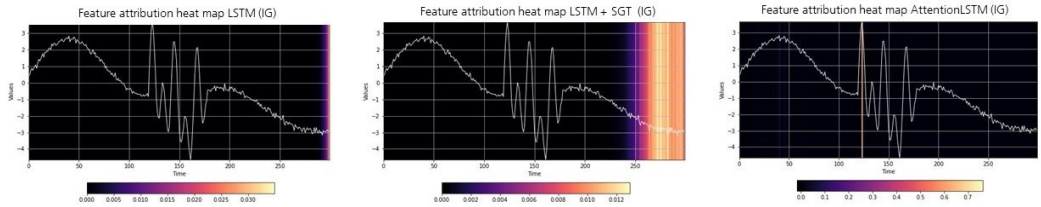
time step for the prediction outcome. If a shapelet is responsible for the class label, we expect the heat map to highlight the time steps in which the shape is occurring. For the experiments which test the detection of the amplitude or the frequency as label-making feature, we expect to see an oscillating heat map focusing either on the peaks or on the valleys of the time series. If the label is based on the phase shift, we expect to either see an emphasis on the beginning of the time series, corresponding to the detection of the importance of the dominant phase shift, or to see a less sparse heat map following the pattern of the peaks or valleys in the input series.
The attention scores provided by the input-cell attention mechanism cannot directly be interpreted as relevance scores per time step, since the mechanism provides r different values for each time step. For aggregating the r attention scores per latent variable for every time step $t = 1, .., T$, we follow the proposed aggregation of the input embedding across the rows of the attention matrix in [63]. After summing the rows of the matrix, the resulting vector is normalized to provide relevance scores in the range of zero to one.
For evaluating the performance of the counterfactual method of Native Guide, the interpretability of the provided explanation is assessed via visual comparison of the instance of interest and the generated counterfactual.

The performance of an explainability method is closely related to the original classification performance of the underlying model. Thus, especially for the LSTM and the LSTM trained via saliency guided training, less informative saliency heat maps or counterfactuals are to be expected. Figure 5.2 depicts examples of the explanations provided by the method Integrated Gradients for the classification by the standard LSTM, the LSTM trained through saliency guided training and the LSTM combined with the input-cell attention mechanism respectively. It is clearly detectable from the explainability heat maps that the LSTM strongly suffers from the vanishing gradient problem. Only few time steps in the end of the time series are assigned an attribution score different from zero. Contrary to the observations in [41], the training procedure did not help to diminish the vanishing saliency problem in the LSTM. In contrast, the input-cell attention mechanism strongly improved the performance of the employed gradient-based saliency method.

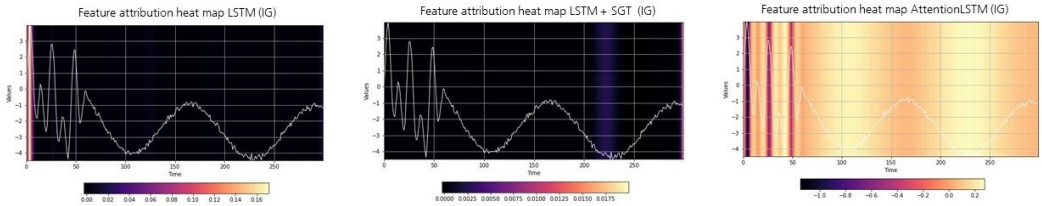
In Figure 5.3, heat maps created based on importance scores assigned by the methods IG and SHAP are compared. The explanations provided by IG align with our expectations and are comparatively easy to interpret. In contrast, the heat map constructed based on importance scores assigned by SHAP do neither align with our expectations nor permit easy interpretability. Thus, we focus on the gradient-based method Integrated Gradients for further comparisons and evaluations. Nevertheless, we note that multiple different explanations can coexist [101]. Claiming the failure of the feature attribution



(a) If the label is based on frequency, only on the AttentionLSTM the method IG can correctly highlight frequency related patterns. SGT does not significantly reduce the vanishing saliency problem.

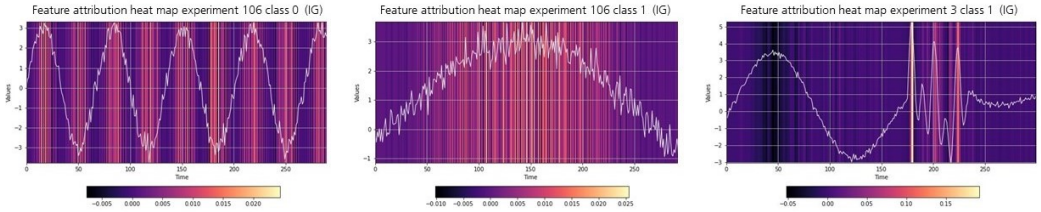


(b) The saliency method IG is not able to highlight the shapelet in the middle of the time series on the LSTM and LSTM+SGT. On the AttentionLSTM the method correctly identifies the start of the shapelet as important.

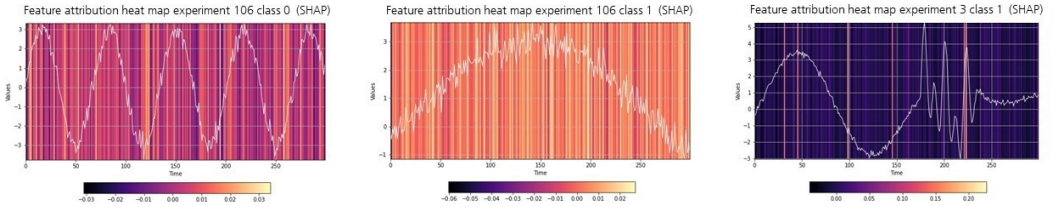


(c) On the LSTM the explainability method successfully highlights the starting point of the shapelet. We hypothesize that the classifier learned to only focus on the first few time steps, due to the fixed position of the shapelet in the beginning of the time series. SGT does not improve the saliency performance.

Figure 5.2: Comparison explanations provided by IG on LSTM, LSTM + SGT and AttentionLSTM.



(a) Explanations provided by IG. In experiment 106 the method clearly focuses on aspects related to the amplitude. In experiment 3 the method identifies the shape possibly through a change in frequency, indicated by the highlighted peaks.

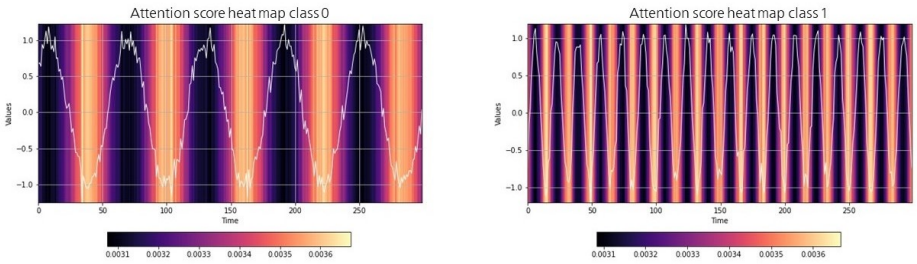


(b) Explanations provided by SHAP. The results for experiment 106 are visually not interpretable and thus not human friendly. In experiment 3 the method highlights time steps outside of the region of the shape.

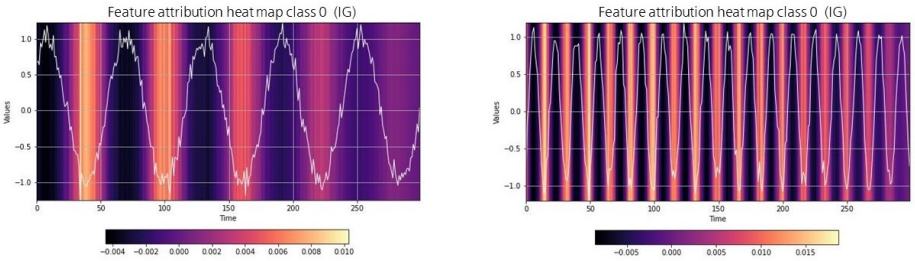
Figure 5.3: Comparison of importance heat maps from feature attribution methods IG and SHAP on experiments 3 and 106 employing CNN and CNN + SGT.

method Kernel-SHAP in our experiments should thus not be directly concluded only from a comparison with other explanations.

After observing the strong classification performance of the LSTM combined with the input-cell attention mechanism in Section 5.1.1, we compare the explanations provided by the attention scores to the explanations provided by the gradient-based saliency method IG. Figure 5.4 depicts heat maps based on IG and on the attention scores for each one input time series of class 0 sampled from low frequencies and class 1 sampled from high frequencies of experiment 102. Both methods focus on the valleys of the time series, thus correctly implicating a transformation of the concept frequency. In the direct comparison of the explainability heat maps, it is observable that the attention-based maps are on one hand more precise than the gradient-based heat maps and on the other hand consistent over time. Due to the superior explanations provided by the attention scores, we decided to develop an extension of the input-cell attention mechanism to the latent space as introduced in Chapter 4. Based on the stated and other similar observations, interpreting attention as explanation is justifiable for the



(a) Background heat map of the attention scores. A lighter color represents a higher score and thus higher importance. The attention mechanism clearly focuses on frequency-related patterns in the time series.



(b) Background heat map of the feature attribution scores assigned by the explainability method Integrated Gradients. The identified importance pattern coincides with the importance pattern identified by the attention scores.

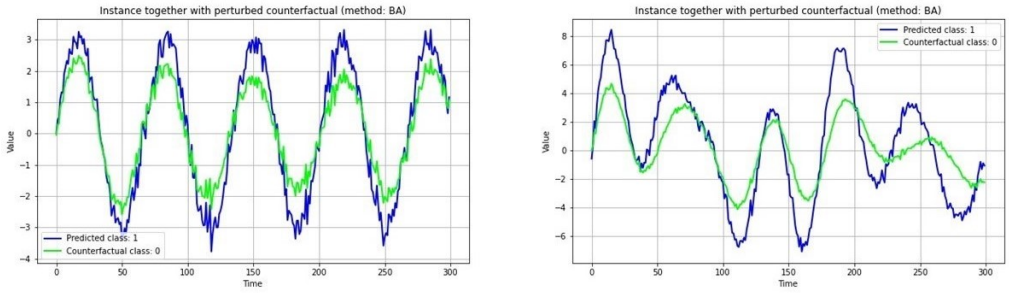
Figure 5.4: Comparison of importance heat maps from attention scores and IG for experiment 102 (important feature = frequency).

conducted experiments.

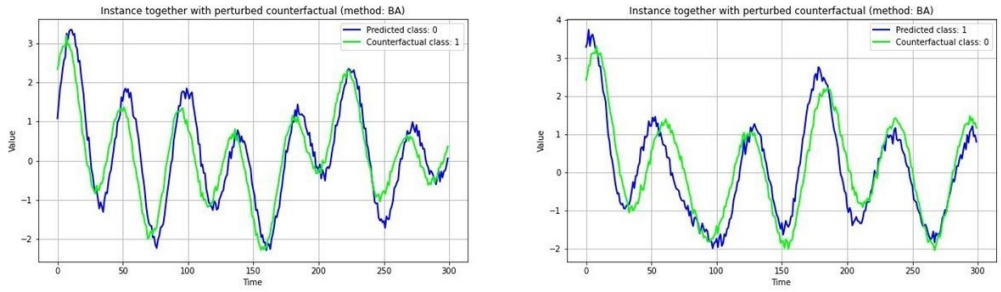
The counterfactual explanation method Native Guide achieves mixed performance on the different data sets as shown in Figure 5.5 and Figure 5.6. This type of explainability method is not visualized through a heat map, but by a direct combined plot of the instance of interest and the counterfactual itself.

Generating counterfactuals through weighted dynamic barycenter averaging of the instance of interest and its nearest unlike neighbour when the class label was based on the amplitude or phase shift, provides valuable visualization results, as can be seen in Figure 5.5. The counterfactual correctly depicts a translation in phase or amplitude. Nevertheless, direct interpretability can be difficult, as the results might easily be misleading.

While the method finds the nearest unlike neighbor (NUN) across the whole training data set, there is a possibility that the NUN is a misclassified instance from the same



(a) The counterfactuals have a significantly lower amplitude than the instance of interest (TCN experiment 105).

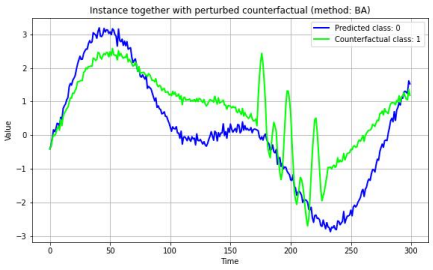


(b) Besides a shift along the time time axis the instance of interest and the counterfactual are very similar (CNN + SGT experiment 104).

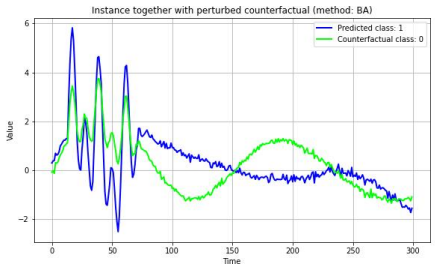
Figure 5.5: Good explanations provided by the counterfactual method Native Guide perturbing the NUN through barycenter averaging when the class label depends on one of the latent features amplitude or phase shift.

class. This can lead to unuseful explanations. An extreme case of this behaviour can be observed in experiment three, in which the label depends on the presence of a shapelet in the time series. Example visualizations can be found in Figure B.1 in Appendix B.3.

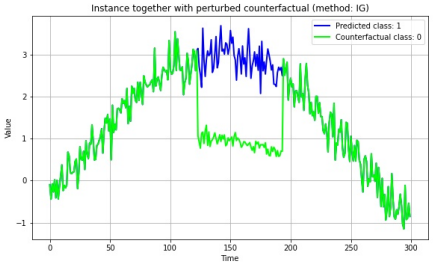
Overall, the counterfactual method Native Guide did not provide satisfactory explanations. In many cases there was no immediate human-readable interpretation associated to the results. This observation opposes the conclusion in [81] which recommends the use of counterfactuals instead of feature attribution methods as explanations, especially for non-experts in the respective domain. The authors propose to a two-step approach to gain insights into the decision process of a model. After the first overview was provided by counterfactuals, in-depth analysis can be performed through feature attribution methods. In our experiments, counterfactual explanations can aid in interpreting the visualizations of the feature-wise importance scores. Even unreasonable results as in



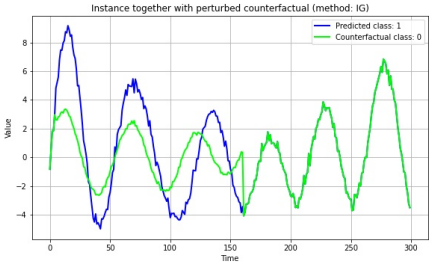
(a) The method correctly inserts a shapelet in the counterfactual. Nevertheless, the explanation provided by the counterfactual is not very precise and not necessarily useful.



(b) Since the counterfactual includes a shapelet as well, an interpretation of the provided explanation is rather difficult.



(c) Perturbation based on feature attribution scores might lead to completely unreasonable explanations (TCN experiment 106).



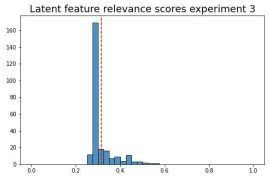
(d) Sparse results provided by perturbation based on feature attribution scores which are not necessarily close to the data distribution (TCN experiment 105).

Figure 5.6: Mixed results for the counterfactual explanation method Native Guide.

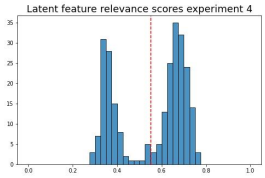
Figure 5.6 (c) can help in differentiating between high feature importance scores due to frequency or amplitude.

5.1.3 Latent feature saliency results

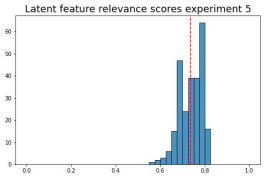
Figure 5.7 depicts histograms of the final scores p derived through equation 4.1 over the complete test data set for each experiment. In Figure 5.7(a), the results for experiment three in which the label was based on a shapelet at a random position in the time series are presented. Our method presented in Chapter 4 correctly assigns low scores, indicating that the explanation provided by the attention mechanism cannot be related



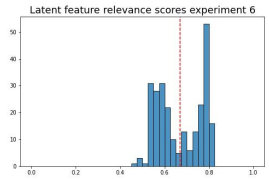
(a) Low scores are assigned, implying that the label does not depend on the Fourier concepts.



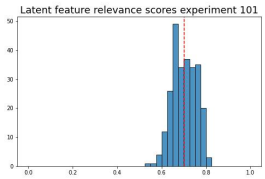
(b) The method does not provide much insights about the importance of the latent space.



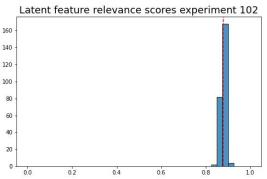
(c) Contrarily to our expectation, the method consistently assigns high scores.



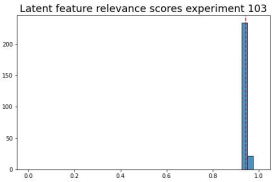
(d) As in experiment 5, the method implies importance of the latent space.



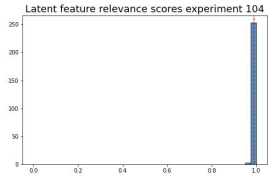
(e) The importance of the latent space can be detected in all samples of the test set.



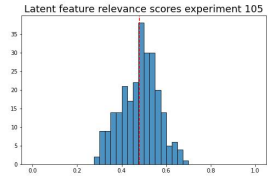
(f) The very high scores across the test data set correctly imply the importance of the Fourier concepts.



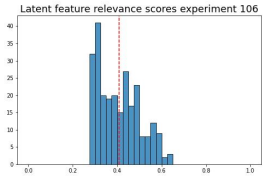
(g) If label depends on the phase shift, the method consistently assigns extremely high scores.



(h) As for experiment 103 the results strongly imply high importance of the Fourier concepts.



(i) If the label is based on the amplitude, the method fails to provide accurate results.

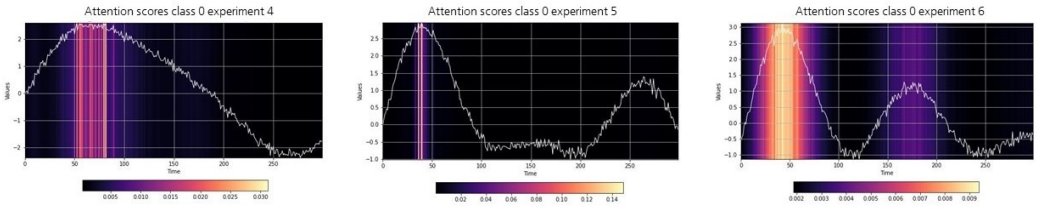


(j) Similar behaviour as in experiment 105 can be observed.

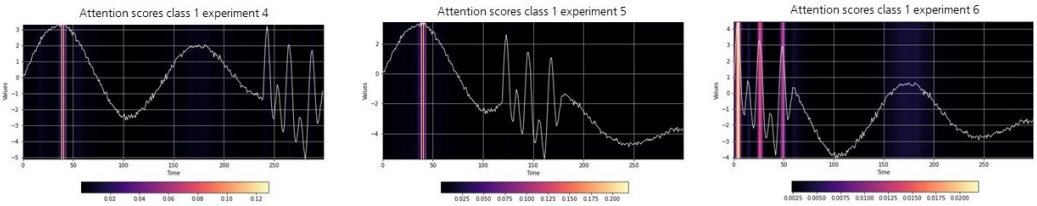
Figure 5.7: Distribution of latent saliency scores p across the test data set per experiment.

to the Fourier latents. The latent saliency results for the experiments in which the label was based on frequency or phase shift are outlined in Figure 5.7(e) to 5.7(h). Our method consistently assigns high scores, implying that the explanation should be interpreted in terms of the Fourier concepts.

If the class label is based on the amplitude, our method is not able to consistently assign high scores, but rather is indifferent about the importance of the Fourier concepts. We hypothesize that this behaviour is due to the direct linear relation of the amplitude with the resampled input time series in contrast to the non-linear indirect relation of frequency and phase shift. For the experiments in which the label is based on the presence of a shapelet in either the beginning, the middle or the end of the time series respectively, our method does not provide the expected results in form of consistently low scores as can be observed in Figure 5.7(b) to 5.7(d). Nevertheless, the unexpected latent saliency scores do not imply the failure of our method, but are caused by the misleading explanations provided by the attention mechanism as presented in Figure 5.8. Instead of highlighting time steps in which the shapelet occurs, the mechanism seems to focus on frequency- or amplitude-related aspects.



(a) If there is no shapelet present in the time series, the provided explanation is slightly similar to the heat map when amplitude or frequency are important.



(b) In experiments four and five, the method does not highlight time steps in which the shape occurs. The explanation provided for experiment six has a slight similarity to heat maps for experiments in which amplitude or frequency are important.

Figure 5.8: The attention mechanism fails to provide useful results on the shapelet experiments four, five and six.

We note that the provided explanations should not be interpreted as a failure of the attention mechanism as an explainability method, since the classifier overall achieves very high performance and even outperforms the other classifiers in experiments five and six as depicted in table B.5 to B.7 in appendix B. Due to the fixed position of the shapelet, the classifier potentially learned a proxy for the shape in terms of frequency. Explainability methods do not necessarily explain the obvious human-interpretable reasons for a classification outcome, but the aspects the decisions of the classification model are based on.

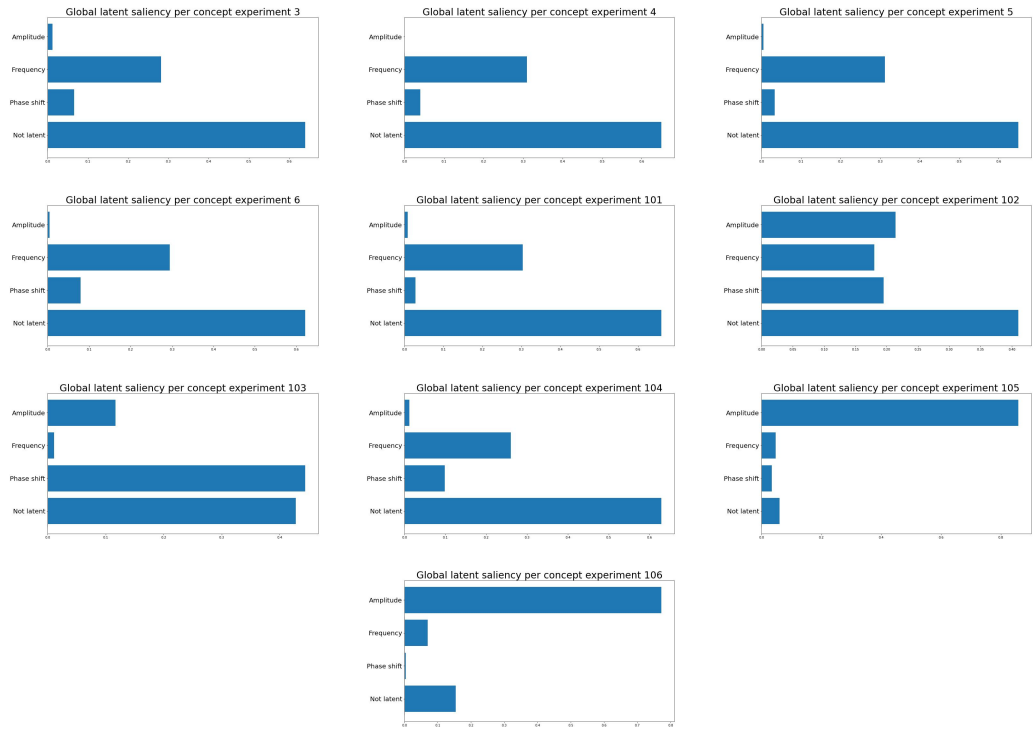
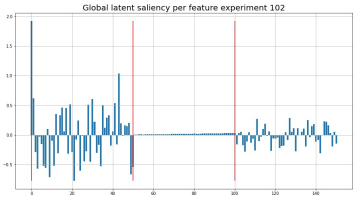


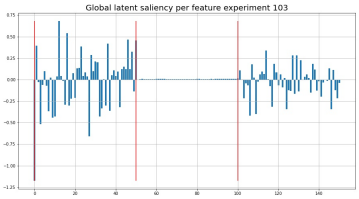
Figure 5.9: Global importance of latent features per experiment.

As a global latent saliency method we proposed to employ a straightforward logistic regression approach on the Fourier coefficients. The resulting regression coefficients corresponding to the same concept amplitude, frequency or phase shift are summed to obtain the final latent saliency score per concept. The logistic regression coefficient of the regression offset is interpreted as an importance score for features different from the three mentioned concepts. This could also be a feature in the time dimension as a class specific shape. Figure 5.9 depicts the global latent saliency results per experiment. The method clearly detects the importance of the amplitude in the case in which the

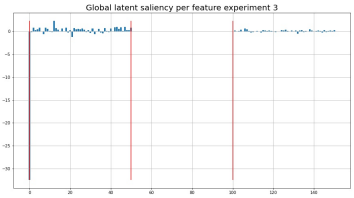
label depends on this latent concept (experiment 105 and 106). All other scenarios for assigning a class label cannot easily be explained by the proposed global method. The aggregation of the regression coefficients to one importance score per concept introduces a huge loss of information as can be seen by comparing Figure 5.9 with Figure 5.10 (a) and (c). Part (c) clearly depicts an important effect which cannot be related to the latent concepts amplitude, frequency and phase shift. After aggregation, as shown in Figure 5.9, the two scenarios in which the label is based on a shape in the time series or on the latent concept frequency cannot be distinguished anymore. Furthermore, the saliency method does not differentiate between importance of frequency and importance of the phase shift.



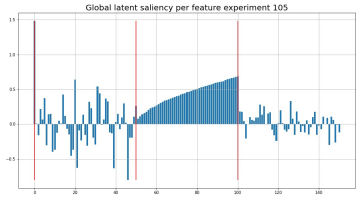
(a) The model is not able to confidently assign high scores to the important concept frequency.



(b) Importance of the phase shift is easily confused with an effect of the frequency.



(c) If the label is based on a shape in the time series, the coefficient of the offset is extremely heightened.



(d) The effect of the amplitude can be easily detected by the method.

Figure 5.10: Coefficients of logistic regression for different experiments. The coefficient for the offset as well as for the latent features frequency, amplitude and phase shift are separated by red lines in the stated order.

5.2 Discussion

In the following section, conclusions and recommendations based on the presented results are stated. Furthermore, the applicability of the proposed latent feature saliency framework and its limitations are discussed.

5.2.1 Classification performance and explainability

From the comparison of the classification performance we observe that the CNN strongly outperforms the LSTM on the presented time series classification tasks. The input-cell attention mechanism strictly improves the classification performance of the LSTM. These observations contrast the results and conclusions in [100]. The authors state that there is no significant difference in the performance of LSTM, CNN and LSTM combined with an attention mechanism on the task of process outcome prediction. In our work we obtained empirical results and examples which illustrate settings where the attention mechanism adds high value to the LSTM for time series classification problems. Although the tasks and architectures are not completely alike, we question the conclusions taken about the attention mechanism, especially due to the high popularity of the latter in the literature. We argue, that it is essential to carry out further rigorous and versatile analysis in order to justify the increased popularity of employing attention mechanisms in the stated tasks.

The runtime of the CNN models is significantly shorter than of the LSTM-based classifiers as shown in table B.14 in Appendix B. In this aspect we agree with the authors of [100], recommending the use of CNNs over the use of LSTMs due to the stated observation. Although the saliency guided training procedure adds high value to the CNN, it strongly increases the computational complexity of the classifier, since a masked input and the respective classification output need to be computed for every input sample and the overall loss function is heightened in complexity as well. The same observation holds true for the LSTM combined with the input-cell attention mechanism. An attention embedding is computed for every input sample before it is passed to the LSTM. We are thus facing a trade-off between classification performance and computational complexity. Solely the TCN is able to improve the classification performance while keeping the same level of complexity. Thus, if run-time is a critical aspect we recommend the use of a temporal convolutional network architecture in terms of classification and explainability performance.

Based on the comparison of the three post-hoc and the attention mechanism as ante-hoc explainability methods, we find that the provided explanations mostly do not match each other. This observation is in accordance with the results in [70]. Nevertheless, since the explanations provided by IG and the attention mechanism are very similar and further correspond to the heat maps we expected to observe in the different scenarios, we argue that these two methods should be considered the strongest in terms of providing interpretable importance scores throughout our experiments. As stated in Section 5.1, we do not intend to condemn the other explainability methods. Since multiple explanations can coexist [101], it might aid to consider various XAI methods to find the correct interpretation of a model. From our experiments it became obvious

that counterfactual explanations, although difficult to interpret on their own, can help to clarify the explanations from heat maps based on feature attribution or attention scores.

5.2.2 Attention as explanation

We have shown that in our setting, attention weights are highly correlated with gradient-based feature attribution scores, contradicting the related finding of [45]. Thus, we conclude that the input-cell attention mechanism can be employed for explanation purposes on time series classification tasks. We note, that the studies in the literature condemning the usage of attention mechanisms as explanations investigate different types of mechanisms. Therefore, the respective conclusions are not mutually exclusive. Nevertheless, we emphasize on the strong explanatory power of the input-cell attention mechanism and recommend to consider this type of attention architecture in the ongoing discussion of whether attention is explanation. To legitimately deduce explanations from attention mechanisms, it is inevitable to define boundaries of the applicability of these mechanisms as XAI methods based on solid theoretical background and domain knowledge.

5.2.3 Applicability of the proposed method and limitations

Our proposed method does not require access to the classification model itself. Therefore, it is highly applicable to many real world scenarios in which the end-user of a classification model is only provided with an interface of the classifier, but not with the model itself. An AI model might run on an external server operated by the IT department of a company. End-users send the to-be-classified samples to the server and are provided with a prediction. For interpretation and explanation of the classification outcome, a model-agnostic explainability method is needed.
Nevertheless, the proposed extension of explainability methods cannot be considered completely model-agnostic, since for calculation of the saliency scores per latent feature access to the internals of the original explainability technique is necessary. Furthermore, for allowing a gradient calculation, the explainability method which should be extended needs to be differentiable as a function of the latent parameters. Since in general, explainability methods do not necessarily need to be differentiable, this poses a strong limitation on the proposed latent feature saliency method. Since our method is based on gradient calculations, it suffers from the same known problems as other gradient based techniques as for example high noise.

For the scope of this project the latent Fourier concepts amplitude, frequency and phase shift were considered. If the class-label of a time series is based on different latent concepts as trend or change points, the proposed method is likely to yield low

scores, implying that the classification decision was not related to the Fourier concepts. Although this implication holds true, it poses strong limits on our framework. An extension of the latent feature saliency framework to other latent is necessary for real-world application. In some cases the framework might yield high results, although the class label depends on a shapelet. If the classifier associated the presence of a shapelet with a change in distribution of the Fourier coefficients, our framework will assign high scores to the latent concepts. This might be an explanation for the unexpected results of experiments four to six.

The logistic regression approach as a global baseline method can be applied to provide latent feature explainability of any classifier if access to the data set is given. In real world scenarios this might not necessarily be the case. A doctor wondering why a patient was diagnosed with neurological disorder based on classification of his gait via wearable sensor technology most probably does not have access to the AI model itself let alone the data set on which the model was developed.
This limitation in applicability was one reason for focusing on local explanations throughout this thesis. The provided global method in general does not provide satisfactory results. Nevertheless, we argue that it might be worth following the idea of adapting a logistic regression approach for the purposes of global latent feature explainability in future work.

Chapter 6

Conclusion

Explainability of deep time series classification is an uprising and highly important field of research. To aid building trust in AI model as well as to identify artifacts and failure modes, interpretation and explanation of the black-box classifiers is essential. Various XAI methods designed for different domains have been introduced in the literature to explain time series classification. These methods focus on positional information of the input features, providing spatial explanations. In time series data, the class label does not necessarily depend on positional information, but might as well depend on the latent space features of a time series. To the best of our knowledge, there does not exist an XAI method with the ability to explain the importance of latent features in the literature.

In this thesis, we outlined and formalized this problem of current XAI methods for time series classification and developed a framework for extending the interpretation of time-step-wise importance scores to the latent space of the time series. To do so, we compared multiple LSTM and CNN-based classifiers on various synthetic univariate time series data sets. We empirically showed that if the class-label is based on the latent features of the time series instead of the presence of a certain shape, commonly applied XAI methods do not provide accurate or human-interpretable explanations. To tackle this problem, a novel extension of these explainability methods was provided which maps the importance scores per time step to one overall importance score of the Fourier-based latent space of the time series. Our proposed framework was able to correctly identify the importance of the latent Fourier features in almost all conducted experiments. Additionally, we provided a baseline approach for future research on global latent explainability methods for time series classification.

Furthermore, we investigated the interpretive power of different post-hoc explainability methods. We found that heat maps from a gradient-based post-hoc explainability method accurately highlight the informative temporal regions, i.e. a shapelet or a pattern which is implicative of a latent information, providing comparatively easy interpretation. Although counterfactual explanation methods are recommended as a simple interpretable method for non-experts in the literature, our experiments showed

that the provided explanations often are imprecise or far from the ground truth, preventing a direct interpretation. Nevertheless, these explanations can aid in analysing the heat map visualizations of timestep-wise importance scores. The legitimisation of attention mechanisms as ante-hoc XAI methods is a lively discussed topic in the literature. Our experiments provide evidence for interpreting attention as explanation. Attention-based heat maps not only align with gradient-based heat maps, but also present more precise and consistent explanations over time.

As for future work, application of the proposed explainability method extension framework to other explainability techniques can be investigated. An extension of the framework to provide one importance score per latent concept would additionally be of high interest. During this project, the latent features amplitude, frequency and phase shift were considered. Other time series models such as state space or switching point models, with respective parameters such as trend or change points can be considered. An extension of the proposed method to provide importance scores for further latent features is as well left for future work. Moreover, testing our framework on data sets in which a combination of multiple features is responsible for the class label is of great interest.

Overall, we highlighted the problem of common XAI methods lacking the ability to assign importance scores to the latent space of a time series and proposed a novel framework for extending explainability methods to also consider latent features. Due to its comparatively general applicability, the provided framework presents a good baseline for future methods. In general, our work sheds light on the need for further research in the field of latent feature saliency detection for deep time series classification.

Appendix A

Notations

In this appendix, notations and symbols as well as abbreviations employed throughout the thesis are listed.

Abbreviation	Description
AI	Artificial Intelligence
AUROC/AUC	Area under the ROC curve
CNN	Convolutional Neural Network
DBA	DTW Baryventer Averaging
DNN	Deep Neural Network
DTW	Dynamic Time Warping
FCN	Fully Convolutional Neural Network
FFT	Fast fourier transform
FN / FP	False negative / false positive
FNR / FPR	False negative rate / false positive rate
IG	Integrated Gradients (feature attribution method)
LRP	Layer-wise relevance propagation
LSTM	Long Short Term Memory
NUN	Nearest unlike neighbor
RNN	Recurrent Neural Network
ROC	Receiver operating characteristic
SENN	Self Explaining Neural Network
SGT	Saliency guided training
SHAP	Kernel SHapley Additive exPlanation Values
TN / TP	True negative / true positive
TNR / TPR	True negative rate / true positive rate
TCN	Temporal Convolutional Network
XAI	Explainable Artificial Intelligence
xDNN	Explainable Deep Neural Network

Table A.1: List of abbreviations

Symbol	Description
X, Y	Time series vector with values $(x_1, ..., x_T)$ or $(y_1, ..., y_T)$ respectively
$\bar{Y}$	Average time series
T	Length of time series
D	Dataset
A	Amplitude
ω	Frequency
ϕ	Phase shift
a_0	Initial offset in Fourier series
$\tilde{n}$	Number of amplitudes present in the time series
$\tilde{T}$	Maximum number of amplitudes which can be recovered by FFT
f	Classification/ prediction model
σ	Sigmoid function
S	Saliency matrix
SV	Saliency vector
s	Importance score
$\mathcal{A}$	Attention matrix
$a_{i,t}$	Element (i, t) of the attention matrix
$\tilde{a}_{i,t}$	Element (i, t) of the raw attention matrix
$\mathcal{A}V$	Attention vector
$\alpha_{i,t}$	Attention weight (synonym for element (i, t) of the attention matrix used in the literature)
W	Weight matrix
w	Weight
p	Score for likelihood of explanations being related to the latent features
$\odot$	Element-wise multiplication
$*$	Convolution operation

Table A.2: List of symbols and notations

Appendix B

Experimental design and results

B.1 Data generation

All time series were sampled to have equal length of 300 time steps. For training, validating and testing the data set of in total 2560 samples was split into sets of 2048, 256 and 256 samples respectively.

Experiment	Label feature	Description of shapelet
3	Shapelet	Random position, window length of 0.2 $*$ sequence length
4	Shapelet	Fixed position, last 0.2 $*$ sequence length timesteps
5	Shapelet	Fixed position, starting at time step 0.4 $*$ sequence length with window length 0.2 $*$ sequence length
6	Shapelet	Fixed position, first 0.2 $*$ sequence length timesteps
101	Frequency	Overlapping frequency ranges
102	Frequency	Overlapping frequency ranges
103	Phase shift	Non-overlapping phase shift ranges
104	Phase shift	Non-overlapping phase shift ranges
105	Amplitude	Different dominant amplitude
106	Amplitude	Different dominant amplitude

Table B.1: Label-making features per experiment.

Table B.1 lists the parameters and algorithms for assigning labels to each sample. In table B.2 the parameters used for sampling the Fourier series are presented.

Exp.	Number of sines	Freq. low	Freq. high	Phase low	Phase high	Dominant amplitude	Decay rate	Noise ratio
3	10	$\frac{\pi}{300}$	$\frac{\pi}{60}$	$\frac{-\pi}{4}$	$\frac{\pi}{4}$	1	0.3	0.1
4	10	$\frac{\pi}{300}$	$\frac{\pi}{20}$	$\frac{-\pi}{4}$	$\frac{\pi}{4}$	1	0.3	0.1
5	10	$\frac{\pi}{300}$	$\frac{\pi}{20}$	$\frac{-\pi}{4}$	$\frac{\pi}{4}$	1	0.3	0.1
6	10	$\frac{\pi}{300}$	$\frac{\pi}{20}$	$\frac{-\pi}{4}$	$\frac{\pi}{4}$	1	0.3	0.1
101	10/10	$\frac{\pi}{300}/\frac{\pi}{100}$	$\frac{\pi}{20}/\frac{\pi}{2}$	$\frac{-\pi}{4}/\frac{-\pi}{4}$	$\frac{\pi}{4}/\frac{\pi}{4}$	1/1	0.3/0.3	0.1/0.1
102	1/1	$\frac{\pi}{300}/\frac{\pi}{100}$	$\frac{\pi}{20}/\frac{\pi}{2}$	$\frac{-\pi}{4}/\frac{-\pi}{4}$	$\frac{\pi}{4}/\frac{\pi}{4}$	1/1	0.3/0.3	0.1/0.1
103	1/1	$\frac{\pi}{300}/\frac{\pi}{300}$	$\frac{\pi}{20}/\frac{\pi}{20}$	$0/\frac{-\pi}{4}$	$\frac{\pi}{4}/\frac{\pi}{2}$	1/1	0.3/0.3	0.1/0.1
104	10/10	$\frac{\pi}{300}/\frac{\pi}{300}$	$\frac{\pi}{20}/\frac{\pi}{20}$	$0/\frac{-\pi}{4}$	$\frac{\pi}{4}/\frac{\pi}{2}$	1/1	0.3/0.3	0.1/0.1
105	10/10	$\frac{\pi}{300}/\frac{\pi}{300}$	$\frac{\pi}{20}/\frac{\pi}{20}$	$0/\frac{-\pi}{4}$	$\frac{\pi}{4}/\frac{\pi}{4}$	1/3	0.3/0.3	0.1/0.1
106	1/1	$\frac{\pi}{300}/\frac{\pi}{300}$	$\frac{\pi}{20}/\frac{\pi}{20}$	$\frac{-\pi}{4}/\frac{-\pi}{4}$	$\frac{\pi}{4}/\frac{\pi}{4}$	1/3	0.3/0.3	0.1/0.1

Table B.2: Overview of simulation parameters of the Fourier series. If two entries are present in one cell, each the classes were sampled from different distributions. The first entry in each cell corresponds to the sampling parameter of class 0, the second entry to class 1.

B.2 Implementation details

A detailed overview of employed network layers and hyperparameters is provided in table B.3. For the input-cell attention mechanism the attention hops were consistently chosen as $r = 50$. All classifiers only consisted of one layer. The layer-architecture of each network is stated below. For the sake of simplicity no dropout-layers was employed in the networks. The network weights were updated using the Adam optimization algorithm.

Classifier	Architecture	Further parameters
LSTM	LSTM block + Fully Connected + Sigmoid	hidden-dim $\in [4, 64]$
LSTM + SGT	LSTM block + Fully Connected + Sigmoid	hidden-dim $\in [4, 64]$
AttentionLSTM	Input-cell attention + LSTM block + Fully Connected + Sigmoid	hidden-dim $\in [4, 64]$
CNN	$2 \times$ (Conv(5) + ReLU + MaxPool(2)) + Fully Connected + Sigmoid	stride length = 2, num. f. maps = 4, 8
CNN + SGT	$2 \times$ (Conv(5) + ReLU + MaxPool(2)) + Fully Connected + Sigmoid	stride length = 2, num. f. maps = 4, 8
TCN	$2 \times$ (CausalConv(5) + ReLU + MaxPool(2)) + Fully Connected + Sigmoid	dilation = 1, 2, stride(MaxPool) = 2, num. f. maps = 4, 8

Table B.3: Detailed description of the classifier architectures. The kernel size of the convolutional layer and the max-pooling layer are stated in parenthesis following the respective layer. Feature map (f. map) and dilation description correspond to the two convolutional layers separately.

B.3 Results

This section provides supplementary material about the classification performance for all conducted experiments, a comparison of the training time of the investigated classifiers as well as further plots showing explainability results.

B.3.1 Classification metrics per experiment

Table B.4 to table B.13 provide the detailed evaluation results of the classifiers for each experiment. As discussed in Chapter 5, overall the CNN + SGT achieves the highest performance. The AttentionLSTM also significantly improves the performance of the standard LSTM in almost all experiments. The classifiers were trained for 200 epochs each. Hyperparameter optimization was limited to 17 hours. As presented in table B.14, the run-time varies enormously across the classifiers, which as well strongly influences the hyperparameter optimization. Due to the heightened training time of CNN + SGT and especially of the AttentionLSTM, the good performance of these models should even carry more weight in the model evaluation and comparison.

Classifier	Accuracy	Precision	Recall	F1	AUROC
LSTM	0.8281	0.7561	0.9688	0.8493	0.8516
LSTM + SGT	0.6992	0.6474	0.8750	0.7442	0.7272
AttentionLSTM	0.8633	0.7962	0.9766	0.8772	0.9170
CNN	0.9648	0.9612	0.9688	0.9650	0.9951
TCN	0.9766	0.9841	0.9688	0.9764	0.9982
CNN + SGT	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**

Table B.4: Comparison of classification performance Experiment 3

Classifier	Accuracy	Precision	Recall	F1	AUROC
LSTM	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
LSTM + SGT	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
AttentionLSTM	0.9922	0.9922	0.9922	0.9922	0.9985
CNN	0.5000	0.5000	**1.0000**	0.6667	0.8037
TCN	0.5000	0.5000	**1.0000**	0.6667	0.3950
CNN + SGT	0.5000	0.5000	**1.0000**	0.6667	0.7449

Table B.5: Comparison of classification performance Experiment 4

Classifier	Accuracy	Precision	Recall	F1	AUROC
LSTM	0.5859	0.5655	0.7422	0.6420	0.6260
LSTM + SGT	0.7227	0.6887	0.8125	0.7455	0.7864
AttentionLSTM	0.9453	0.9130	0.9844	0.9474	0.9764
CNN	0.5234	0.5120	**1.0000**	0.6773	0.6932
TCN	0.5000	0.5000	**1.0000**	0.6667	0.6046
CNN + SGT	**0.9883**	**0.9771**	**1.0000**	**0.9884**	**1.0000**

Table B.6: Comparison of classification performance Experiment 5

Classifier	Accuracy	Precision	Recall	F1	AUROC
LSTM	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
LSTM + SGT	0.8750	0.8636	0.8906	0.8769	0.9105
AttentionLSTM	0.9219	0.8648	**1.0000**	0.9275	0.9601
CNN	0.5000	0.5000	**1.0000**	0.6667	0.9370
TCN	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
CNN + SGT	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**

Table B.7: Comparison of classification performance Experiment 6

Classifier	Accuracy	Precision	Recall	F1	AUROC
LSTM	0.9688	0.9412	1.0000	0.9697	0.9809
LSTM + SGT	0.9375	0.8889	1.0000	0.9412	0.9407
AttentionLSTM	0.9922	**0.9922**	0.9922	0.9922	0.9973
CNN	**0.9961**	**0.9922**	**1.0000**	**0.9961**	**1.0000**
TCN	**0.9961**	**0.9922**	**1.0000**	**0.9961**	**1.0000**
CNN + SGT	**0.9961**	**0.9922**	**1.0000**	**0.9961**	**1.0000**

Table B.8: Comparison of classification performance Experiment 101

Classifier	Accuracy	Precision	Recall	F1	AUROC
LSTM	0.9180	0.8591	1.0000	0.9242	0.9097
LSTM + SGT	0.8789	0.8593	0.9063	0.8821	0.9083
AttentionLSTM	0.8828	0.8451	0.9375	0.8889	0.9144
CNN	**0.9258**	**0.8759**	**0.9922**	**0.9304**	0.9519
TCN	**0.9258**	**0.8759**	**0.9922**	**0.9304**	**0.9529**
CNN + SGT	**0.9258**	**0.8759**	**0.9922**	**0.9304**	0.9479

Table B.9: Comparison of classification performance Experiment 102

Classifier	Accuracy	Precision	Recall	F1	AUROC
LSTM	0.5703	0.5978	0.4297	0.5000	0.5624
LSTM + SGT	0.5391	0.5472	0.4531	0.4957	0.5583
AttentionLSTM	0.8516	0.8169	0.9063	0.8593	0.8851
CNN	0.9336	0.9440	0.9219	0.9328	0.9897
TCN	**0.9688**	**0.9762**	**0.9609**	**0.9685**	**0.9974**
CNN + SGT	0.9297	0.9661	0.8906	0.9268	0.9881

Table B.10: Comparison of classification performance Experiment 103

Classifier	Accuracy	Precision	Recall	F1	AUROC
LSTM	0.5313	0.5385	0.4375	0.4828	0.5532
LSTM + SGT	0.5313	0.5769	0.2344	0.3333	0.5109
AttentionLSTM	0.9336	0.8828	**1.0000**	0.9377	0.9677
CNN	0.9922	0.9922	0.9922	0.9922	0.9998
TCN	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
CNN + SGT	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**

Table B.11: Comparison of classification performance Experiment 104

Classifier	Accuracy	Precision	Recall	F1	AUROC
LSTM	0.9961	**1.0000**	0.9922	0.9961	0.9929
LSTM + SGT	0.8320	0.7815	0.9219	0.8459	0.9004
AttentionLSTM	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
CNN	0.9961	0.9922	**1.0000**	0.9961	**1.0000**
TCN	0.9961	0.9922	**1.0000**	0.9961	**1.0000**
CNN + SGT	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**

Table B.12: Comparison of classification performance Experiment 105

Classifier	Accuracy	Precision	Recall	F1	AUROC
LSTM	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
LSTM + SGT	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
AttentionLSTM	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
CNN	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
TCN	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
CNN + SGT	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**

Table B.13: Comparison of classification performance Experiment 106

B.3.2 Run-time of classifiers

High computational complexity poses a restriction to the usage of AI models in real world applications. Throughout the thesis we highlight the good classification and explainability performance of the AttentionLSTM. The overall best-performing model is the CNN trained through the saliency guided training procedure. Table B.14 outlines the training times of all classifiers for each one experiment per label-making feature. We can observe that SGT significantly increases the run-time of the models. The AttentionLSTM suffers by far from the highest complexity. Thus, we are facing a strong trade-off between performance and computational effort.

Exp.	LSTM	LSTM+SGT	AttentionLSTM	CNN	TCN	CNN+SGT
3	270.73	2634.76	22756.96	215.13	266.30	560.48
101	220.64	2647.27	10334.24	180.54	232.81	511.42
103	206.66	2272.49	3395.45	184.85	237.60	522.22
105	498.25	2651.30	22703.13	214.24	266.34	504.11

Table B.14: Comparison of training time of classifiers on each one experiment per label-making feature in seconds.

B.3.3 Supplementary plots for explainability method evaluation

The counterfactual method Native Guide provides explanations of mixed quality. One effect strongly increasing the difficulty of interpreting the provided explanations is the search space for finding the nearest unlike neighbor (NUN). Since the NUN is determined through a nearest neighbor approach in the complete training data set, it is likely to be a sample falsely classified as the opposing class, but actually belonging to the same class as the instance of interest. Although explainability methods are not designed to explain the correct decision, but the classifiers actual decision process, the provided results do not provide insights into internals of the classifier. An extreme case of this behaviour is depicted in figure B.1. The classes differ through the presence or absence of a very prominent shape. The provided explanations are highly uninformative or even misleading.

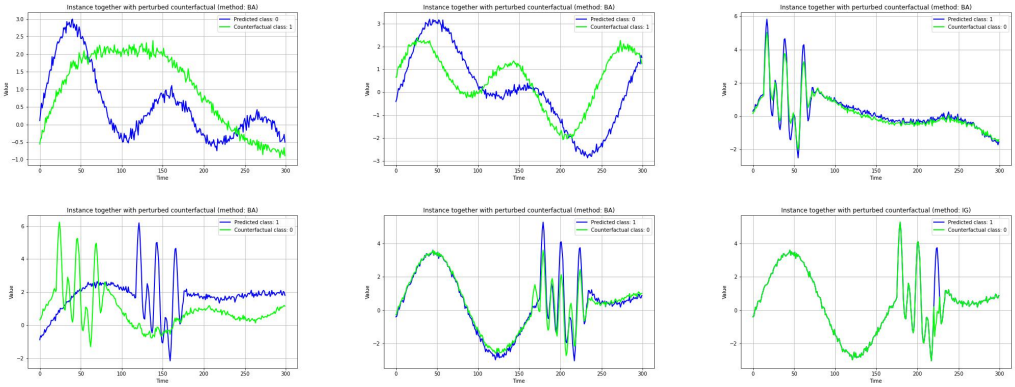


Figure B.1: Results of counterfactual method Native Guide are difficult to interpret, due to misclassified nearest unlike neighbor, when label is based on a shape in the time series.

List of Figures

List of Tables

Bibliography

[1] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019.

[2] D. Alvarez-Melis and T. Jaakkola. "Towards Robust Interpretability with Self-Explaining Neural Networks". In: *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*. Montréal, Canada, June 2018.

[3] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross. "Towards better understanding of gradient-based attribution methods for Deep Neural Networks". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. 2018.

[4] M. Ancona, E. Ceolini, C. Öztireli, and M. H. Gross. "Gradient-Based Attribution Methods". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science*. Vol. 11700. Springer, Cham, 2019.

[5] S. P. Anderson, J. K. Goeree, and C. A. Holt. "The Logit Equilibrium: A Perspective on Intuitive Behavioral Anomalies". In: *Southern Economic Journal* 69.1 (2002), pp. 21–47.

[6] P. P. Angelov and E. A. Soares. "Towards Explainable Deep Neural Networks (xDNN)". In: *Neural networks : the official journal of the International Neural Network Society* 130 (2020), pp. 185–194.

[7] I. Arous, L. Dolamic, A. Yang J.and Bhardwaj, G. Cuccu, and P. Cudré-Mauroux. "MARTA: Leveraging Human Rationales for Explainable Text Classification". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 7. 2021, pp. 5868–5876.

[8] A. B. Arrieta, N. D. Rodrıguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcıa, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI". In: *Information Fusion* 58 (June 2020), pp. 82–115.

[9] V. Arya, R. K. E. Bellamy, P. Chen, A. Dhurandhar, M. Hind, S. C. Hoffman, S. Houde, Q. V. Liao, R. Luss, A. Mojsilovic, S. Mourad, P. Pedemonte, R. Raghavendra, J. T. Richards, P. Sattigeri, K. Shanmugam, M. Singh, K. R. Varshney, D. Wei, and Y. Zhang. "One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques". In: *CoRR* abs/1909.03012 (2019).

[10] E. Ates, B. Aksar, V. J. Leung, and A. K. Coskun. "Counterfactual Explanations for Multivariate Time Series". In: *2021 International Conference on Applied Artificial Intelligence (ICAPAI)*. 2021, pp. 1–8.

[11] S. Bach, A. Binder, G. Montavon, F. Klauschen, K. Müller, and S. W. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLoS One* 10.7 (2015).

[12] D. Bahdanau, K. Cho, and Y. Bengio. "Neural machine translation by jointly learning to align and translate". In: *International Conference on Learning Representations*. San Diego, USA, 2015.

[13] S. Bai, J. Z. Kolter, and V. Koltun. *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*. 2018.

[14] J. Bastings and K. Filippova. "The elephant in the interpretability room: Why use attention as explanation when we have saliency methods?" In: *Proceedings of the 2020 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. 2020.

[15] R. N. Bracewell. *The Fourier transform and its applications*. 3rd ed. McGraw-Hill New York, USA, 2000.

[16] G. Brunner, Y. Liu, D. Pascual, O. Richter, and R. Wattenhofer. "On the Validity of Self-Attention as Explanation in Transformer Models". In: (2020).

[17] N. Burkart and M. Huber. "A Survey on the Explainability of Supervised Machine Learning". In: *Journal of Artificial Intelligence Research* 70 (Jan. 2021).

[18] R. M. Byrne. "Counterfactuals in Explainable Artificial Intelligence (XAI): Evidence from Human Reasoning". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*. Aug. 2019, pp. 6276–6282.

[19] G. Canbek, S. Sagiroglu, T. T. Temizel, and N. Baykal. "Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights". In: *2017 International Conference on Computer Science and Engineering (UBMK)*. 2017, pp. 821–826.

[20] A. Carrillo, L. F. Cantú, and A. Noriega. "Individual Explanations in Machine Learning Models: A Survey for Practitioners". In: *ArXiv* abs/2104.04144 (2021).

[21] E. Choi, M. T. Bahadori, J. A. Kulas, A. Schuetz, W. F. Stewart, and J. Sun. "RETAIN: An Interpretable Predictive Model for Healthcare Using Reverse Time Attention Mechanism". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Barcelona, Spain: Curran Associates Inc., 2016, pp. 3512–3520.

[22] G. Correia, V. Niculae, and A. Martins. "Adaptively Sparse Transformers". In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Hong Kong, Aug. 2019.

[23] Z. Cui, W. Chen, and Y. Chen. "Multi-Scale Convolutional Neural Networks for Time Series Classification". In: *ArXiv* abs/1603.06995 (2016).

[24] A. Datta, S. Sen, and Y. Zick. "Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems". In: *2016 IEEE Symposium on Security and Privacy (SP)*. 2016, pp. 598–617.

[25] E. Delaney, D. Greene, and M. T. Keane. "Instance-Based Counterfactual Explanations for Time Series Classification". In: *Case-Based Reasoning Research and Development: 29th International Conference, ICCBR 2021, Salamanca, Spain, September 13–16, 2021, Proceedings*. Salamanca, Spain: Springer-Verlag, 2021, pp. 32–47.

[26] F. Doshi-Velez and B. Kim. "Towards A Rigorous Science of Interpretable Machine Learning". In: *arXiv: Machine Learning* (2017).

[27] *Encyclopedia of Machine Learning*. Boston, USA: Springer, 2010.

[28] C. of European Union. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*. 2016.

[29] W. Falcon. *PyTorch Lightning*. 2019.

[30] R. C. Fong and A. Vedaldi. "Interpretable Explanations of Black Boxes by Meaningful Perturbation". In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 3449–3457.

[31] R. Fong, M. Patrick, and A. Vedaldi. "Understanding Deep Networks via Extremal Perturbations and Smooth Masks". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 2950–2958.

[32] G. Forestier, F. Petitjean, H. A. Dau, G. I. Webb, and E. Keogh. "Generating Synthetic Time Series to Augment Sparse Datasets". In: *2017 IEEE International Conference on Data Mining (ICDM)*. 2017, pp. 865–870.

[33] B. Gao and L. Pavel. "On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning". In: *arXiv e-prints* (Apr. 2017).

[34] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[35] R. M. Gray. "Entropy". In: *Entropy and Information Theory*. Boston, USA: Springer, 2011.

[36] J. Gu, Y. Yang, and V. Tresp. "Understanding Individual Decisions of CNNs via Contrastive Backpropagation". In: *Computer Vision – ACCV 2018*. Springer, 2018, pp. 119–134.

[37] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. "A Survey of Methods for Explaining Black Box Models". In: *ACM Comput. Surv.* 51.5 (Aug. 2018).

[38] R. Guidotti, A. Monreale, F. Spinnato, D. Pedreschi, and F. Giannotti. "Explaining Any Time Series Classifier". In: *2020 IEEE Second International Conference on Cognitive Machine Intelligence (CogMI)*. 2020, pp. 167–176.

[39] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Sept. 1997), pp. 1735–1780.

[40] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. "Deep Learning for Time Series Classification: A Review". In: *Data Mining and Knowledge Discovery* 33.4 (July 2019), pp. 917–963. ISSN: 1384-5810.

[41] A. A. Ismail, H. Corrada Bravo, and S. Feizi. "Improving Deep Learning Interpretability by Saliency Guided Training". In: Sydney, Australia, 2021.

[42] A. A. Ismail, M. K. Gunady, H. Corrada Bravo, and S. Feizi. "Benchmarking Deep Learning Interpretability in Time Series Predictions". In: *34th Conference on Neural Information Processing Systems*. Vancouver, Canada, 2020.

[43] A. A. Ismail, M. Gunady, L. Pessoa, H. Corrada Bravo, and S. Feizi. "Input-Cell Attention Reduces Vanishing Saliency of Recurrent Neural Networks". In: Vancouver, Canada, 2019.

[44] B. K. Iwana, R. Kuroki, and S. Uchida. "Explaining Convolutional Neural Networks using Softmax Gradient Layer-wise Relevance Propagation". In: 2019, pp. 4176–4185.

[45] S. Jain and B. C. Wallace. "Attention is not Explanation". In: *2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Minneapolis, USA, 2019.

[46] Y.-J. Jung, S.-H. Han, and H.-J. Choi. "Explaining CNN and RNN Using Selective Layer-Wise Relevance Propagation". In: *IEEE Access* 9 (2021), pp. 18670–18681.

[47] K. Kanamori, T. Takagi, K. Kobayashi, and H. Arimura. "DACE: Distribution-Aware Counterfactual Explanation by Mixed-Integer Linear Optimization". In: Yokohama, Oct. 2020.

[48] T. Kanchinadam, K. Westpfahl, Q. You, and G. M. Fung. "Rationale-based Human-in-the-Loop via Supervised Attention". In: *Workshop on Data Science with Human-in-the-loop (DaSH) @ KDD 2020*. 2020.

[49] A. Kapishnikov, T. Bolukbasi, F. Viegas, and M. Terry. "XRAI: Better Attributions Through Regions". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019, pp. 4947–4956.

[50] F. Karim, S. Majumdar, H. Darabi, and S. Chen. "LSTM Fully Convolutional Networks for Time Series Classification". In: *IEEE Access* 6 (2018), pp. 1662–1669.

[51] I. Karlsson, J. Rebane, P. Papapetrou, and A. Gionis. "Explainable Time Series Tweaking via Irreversible and Reversible Temporal Transformations". In: *2018 IEEE International Conference on Data Mining (ICDM)*. Los Alamitos, CA, USA: IEEE Computer Society, 2018, pp. 207–216.

[52] I. Karlsson, J. Rebane, P. Papapetrou, and A. Gionis. "Locally and Globally Explainable Time Series Tweaking". In: *Knowledge Information Systems* 62.5 (2020), pp. 1671–1700.

[53] M. T. Keane and B. Smyth. "Good Counterfactuals and Where to Find Them: A Case-Based Technique for Generating Counterfactuals for Explainable AI (XAI)". In: *Case-Based Reasoning Research and Development: 28th International Conference, ICCBR 2020, Salamanca, Spain, June 8–12, 2020, Proceedings*. Salamanca, Spain: Springer-Verlag, 2020, pp. 163–178.

[54] M. Keane, E. Kenny, E. Delaney, and B. Smyth. "If Only We Had Better Counterfactual Explanations: Five Key Deficits to Rectify in the Evaluation of Counterfactual XAI Techniques". In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*. 2021.

[55] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2015.

[56] G. Kobayashi, T. Kuribayashi, S. Yokoi, and K. Inui. "Attention is Not Only a Weight: Analyzing Transformers with Vector Norms". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Nov. 2020, pp. 7057–7075.

[57] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson. *Captum: A unified and generic model interpretability library for PyTorch*. 2020. arXiv: `2009.07896 [cs.LG]`.

[58] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. "Unmasking Clever Hans Predictors and Assessing What Machines Really Learn". In: *Nature Communications* 10 (Mar. 2019).

[59] Y. Le Cun, L. Jackel, B. Boser, J. Denker, H. Graf, I. Guyon, D. Henderson, R. Howard, and W. Hubbard. "Handwritten digit recognition: applications of neural network chips and automatic learning". In: *IEEE Communications Magazine* 27.11 (1989), pp. 41–46.

[60] A. Le Guennec, S. Malinowski, and R. Tavenard. "Data Augmentation for Time Series Classification using Convolutional Neural Networks". In: *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*. Riva Del Garda, Italy, Sept. 2016.

[61] X.-H. Li, Y. Shi, H. Li, W. Bai, C. C. Cao, and L. Chen. "An Experimental Study of Quantitative Evaluations on Saliency Methods". In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '21. Virtual Event, Singapore: Association for Computing Machinery, 2021, pp. 3200–3208.

[62] B. Lim, S. Arik, N. Loeff, and T. Pfister. "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting". In: *International Journal of Forecasting* 37 (4 2021), pp. 1748–1764.

[63] Z. Lin, M. Feng, C. Nogueira dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. "A Structured Self-attentive Sentence Embedding". In: Toulon, 2017.

[64] S. Lipovetsky and M. Conklin. "Analysis of regression in game theory approach". In: *Applied Stochastic Models in Business and Industry* 17 (2001), pp. 319–330.

[65] J. Long, E. Shelhamer, and T. Darrell. "Fully convolutional networks for semantic segmentation". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3431–3440.

[66] S. M. Lundberg and S.-I. Lee. "A Unified Approach to Interpreting Model Predictions". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 4768–4777.

[67] C. Molnar. *Interpretable Machine Learning*. 2022.

[68] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller. "Explaining Nonlinear Classification Decisions with Deep Taylor Decomposition". In: *Pattern Recognition* 65.C (May 2017), pp. 211–222.

[69] U. Mori, A. Mendiburu, and J. Lozano. “Distance Measures for Time Series in R: The TSdist Package”. In: *The R Journal* 8 (Aug. 2016).

[70] M. Neely, S. F. Schouten, M. J. R. Bleeker, and A. Lucic. “Order in the Court: Explainable AI Methods Prone to Disagreement”. In: *ICML Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI*. 2021.

[71] A. Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[72] F. Petitjean, A. Ketterlin, and P. Gançarski. “A Global Averaging Method for Dynamic Time Warping, with Applications to Clustering”. In: *Pattern Recognition* 44.3 (2011), pp. 678–693.

[73] V. Petsiuk, A. Das, and K. Saenko. “RISE: Randomized Input Sampling for Explanation of Black-box Models”. In: (June 2018).

[74] O. Pfungst and C. L. Rahn. *Clever Hans: (the horse of Mr. Von Osten.) a contribution to experimental animal and human psychologyf*. New York, USA: Holt, Rinehart and Winston, 1911.

[75] S.-A. Rebuffi, R. Fong, X. Ji, and A. Vedaldi. “There and Back Again: Revisiting Backpropagation Saliency Methods”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 8836–8845.

[76] R. A. Rensink. “The Dynamic Representation of Scenes”. In: *Visual Cognition* 7 (2000), pp. 17–42.

[77] 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*. KDD ’16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1135–1144.

[78] M. T. Ribeiro, S. Singh, and C. Guestrin. ““Why Should I Trust You?” Explaining the Predictions of Any Classier”. In: *KDD ’16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016.

[79] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning Representations by Back-Propagating Errors”. In: vol. 323. 1986, pp. 533–536.

[80] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller. “Evaluating the Visualization of What a Deep Neural Network Has Learned”. In: *IEEE Transactions on Neural Networks and Learning Systems* 28 (Nov. 2017), pp. 2660–2673.

[81] U. Schlegel and D. A. Keim. "Time Series Model Attribution Visualizations as Explanations". In: *2021 IEEE Workshop on TRust and EXpertise in Visual Analytics (TREX)*. New Orleans, LA, USA: IEEE, Oct. 2021, pp. 27–31.

[82] U. Schlegel, D. Oelke, D. A. Keim, and M. El-Assady. "An Empirical Study of Explainable AI Techniques on Deep Learning Models For Time Series Tasks". In: (2020).

[83] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, Oct. 2017, pp. 618–626.

[84] J. Serrà and J. Lluis Arcos. "An empirical evaluation of similarity measures for time series classification". In: *Knowledge Based Systems* 67 (2014), pp. 305–314.

[85] L. S. Shapley. "A value for n-person games". In: (1953), pp. 307–317.

[86] S.-Y. Shih, F.-K. Sun, and H.-y. Lee. "Temporal Pattern Attention for Multivariate Time Series Forecasting". In: *Machine Learning* 108 (Sept. 2019), pp. 1421–1441.

[87] A. Shrikumar, P. Greenside, and A. Kundaje. "Learning Important Features Through Propagating Activation Differences". In: *ICML'17: Proceedings of the 34th International Conference on Machine Learning - Volume 70*. Sydney, NSW, Australia: JMLR.org, 2017.

[88] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje. "Not Just a Black Box: Learning Important Features Through Propagating Activation Differences". In: *Proceedings of the 33 rd International Conference on Machine Learning*. Vol. 48. New York, USA, 2016.

[89] K. Simonyan, A. Vedaldi, and A. Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *CoRR* abs/1312.6034 (2014).

[90] D. Smilkov, N. Thorat, B. Kim, B. Kim, F. B. Viégas, and M. Wattenberg. "SmoothGrad: removing noise by adding noise". In: *CoRR* abs/1706.03825 (2017).

[91] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. "Striving for Simplicity: The All Convolutional Net". In: *CoRR* abs/1412.6806 (2015).

[92] J. Strout, Y. Zhang, and R. Mooney. "Do Human Rationales Improve Machine Explanations?" In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 56–62.

[93] E. Štrumbelj and I. Kononenko. "Explaining Prediction Models and Individual Predictions with Feature Contributions". In: *Knowledge and Information Systems* 41.3 (Dec. 2014), pp. 647–665.

[94] K. Sun and A. Marasović. "Effective Attention Sheds Light On Interpretability". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, 2021, pp. 4126–4135.

[95] M. Sundararajan, A. Taly, and Q. Yan. "Axiomatic Attribution for Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. Sydney, NSW, Australia: JMLR.org, 2017, pp. 3319–3328.

[96] H. Suresh, N. Hunt, A. E. W. Johnson, L. A. Celi, P. Szolovits, and M. Ghassemi. "Clinical Intervention Prediction and Understanding with Deep Neural Networks". In: *Machine Learning for Healthcare Conference*. 2017.

[97] F. J. Valverde-Albacete and C. Peláez-Moreno. "100% Classification Accuracy Considered Harmful: The Normalized Information Transfer Factor Explains the Accuracy Paradox". In: *PloS one* 9 (Jan. 2014).

[98] Z. Wang, I. Samsten, R. Mochaourab, and P. Papapetrou. "Learning Time Series Counterfactuals via Latent Space Representations". In: *International Conference on Discovery Science*. Oct. 2021, pp. 369–384.

[99] Z. Wang, W. Yan, and T. Oates. "Time series classification from scratch with deep neural networks: A strong baseline". In: 2017, pp. 1578–1585.

[100] H. Weytjens and J. Weerdt. "Process Outcome Prediction: CNN vs. LSTM (with Attention)". In: *Lecture Notes in Business Information Processing*. Springer Nature, Jan. 2020, pp. 321–333. ISBN: 978-3-030-66497-8.

[101] S. Wiegreffe and Y. Pinter. "Attention is not not Explanation". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 11–20.

[102] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning*. Vol. 37. ICML'15. Lille, France: JMLR.org, 2015, pp. 2048–2057.

[103] L. Ye and E. Keogh. "Time Series Shapelets: A New Primitive for Data Mining". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. Paris, France: Association for Computing Machinery, 2009, pp. 947–956.

[104] M. D. Zeiler and R. Fergus. "Visualizing and understanding convolutional networks". In: *In Computer Vision–ECCV 2014*. Springer, 2014, pp. 818–833.

[105] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu. "Convolutional neural networks for time series classification". In: *Journal of Systems Engineering and Electronics* 28 (Feb. 2017), pp. 162–169.

[106] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. Zhao. "Exploiting multi-channels deep convolutional neural networks for multivariate time series classification". In: *Frontiers of Computer Science* 10 (2016), pp. 96–112.

[107] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. "Learning Deep Features for Discriminative Localization". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2921–2929.