



PDF Download  
3627673.3679844.pdf  
29 December 2025  
Total Citations: 2  
Total Downloads: 503

Latest updates: <https://dl.acm.org/doi/10.1145/3627673.3679844>

RESEARCH-ARTICLE

## Time is Not Enough: Time-Frequency based Explanation for Time-Series Black-Box Models

**HYUNSEUNG CHUNG**, Korea Advanced Institute of Science and Technology, Daejeon, South Korea

**SUMIN JO**, Korea Advanced Institute of Science and Technology, Daejeon, South Korea

**YEONSU KWON**, Korea Advanced Institute of Science and Technology, Daejeon, South Korea

**EDWARD CHOI**, Korea Advanced Institute of Science and Technology, Daejeon, South Korea

**Open Access Support** provided by:

**Korea Advanced Institute of Science and Technology**

**Published:** 21 October 2024

[Citation in BibTeX format](#)

CIKM '24: The 33rd ACM International Conference on Information and Knowledge Management  
October 21 - 25, 2024  
ID, Boise, USA

**Conference Sponsors:**  
SIGIR

# Time is Not Enough: Time-Frequency based Explanation for Time-Series Black-Box Models

Hyunseung Chung  
KAIST

Daejeon, Republic of Korea  
hs\_chung@kaist.ac.kr

Sumin Jo  
KAIST

Daejeon, Republic of Korea  
ekrxjwh2009@kaist.ac.kr

Yeonsu Kwon  
KAIST

Daejeon, Republic of Korea  
yeonsu.k@kaist.ac.kr

Edward Choi  
KAIST

Daejeon, Republic of Korea  
edwardchoi@kaist.ac.kr

## Abstract

Despite the massive attention given to time-series explanations due to their extensive applications, a notable limitation in existing approaches is their primary reliance on the time-domain. This overlooks the inherent characteristic of time-series data containing both time and frequency features. In this work, we present Spectral eXplanation (SpectralX), an XAI framework that provides time-frequency explanations for time-series black-box classifiers. This easily adaptable framework enables users to “plug-in” various perturbation-based XAI methods<sup>1</sup> for any pre-trained time-series classification models to assess their impact on the explanation quality without having to modify the framework architecture. Additionally, we introduce Feature Importance Approximations (FIA), a new perturbation-based XAI method. These methods consist of feature *insertion*, *deletion*, and *combination* techniques to enhance computational efficiency and class-specific explanations in time-series classification tasks. We conduct extensive experiments in the generated synthetic dataset and various UCR Time-Series datasets to first compare the explanation performance of FIA and other existing perturbation-based XAI methods in both time-domain and time-frequency domain, and then show the superiority of our FIA in the time-frequency domain with the SpectralX framework. Finally, we conduct a user study to confirm the practicality of our FIA in SpectralX framework for class-specific time-frequency based time-series explanations. The source code is available [here](#).

## CCS Concepts

• **Mathematics of computing** → *Time series analysis*; • **Computing methodologies** → *Neural networks*.

## Keywords

Time Series, Time-Frequency, Explainability

### ACM Reference Format:

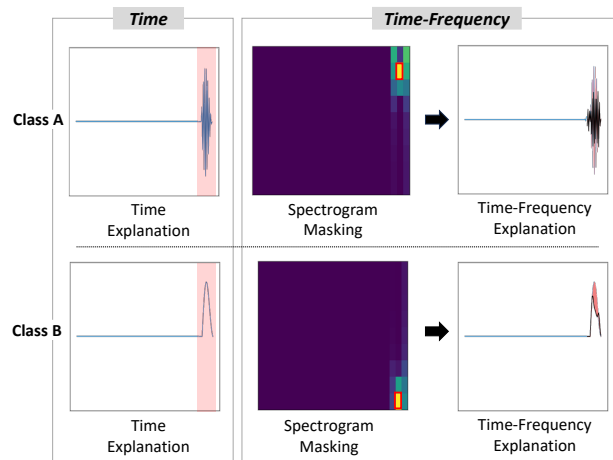
Hyunseung Chung, Sumin Jo, Yeonsu Kwon, and Edward Choi. 2024. Time is Not Enough: Time-Frequency based Explanation for Time-Series Black-Box Models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24)*, October 21–25, 2024, Boise, ID, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3627673.3679844>

<sup>1</sup>In this work, we distinguish between XAI *framework* and XAI *method*.



This work is licensed under a Creative Commons Attribution International 4.0 License.

CIKM '24, October 21–25, 2024, Boise, ID, USA  
© 2024 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-0436-9/24/10  
<https://doi.org/10.1145/3627673.3679844>



**Figure 1: Example of the time-domain explanation, where the important regions are overlapping for both classes. In time-frequency domain explanation, however, zero-masking the red-box, which indicates important region in the time-frequency domain, appears as the red region in original time-domain which explains the most important feature of the class.**

## 1 Introduction

In recent years, the field of Time-Series Classification (TSC) grew rapidly due to the widespread availability of open-source time-series datasets across various domains, and significant advancements in deep learning classification models. These models, particularly Long Short-Term Memory (LSTM) networks [10], Convolutional Neural networks (CNNs) [17], and Transformers [34], have been crucial in effectively capturing intricate temporal features in time-series data. As the complexity of these models increased, their classification accuracy also drastically improved. However, this rapid escalation in complexity overall had a detrimental impact on model interpretability, making these “black box” models less transparent and harder to understand [1].

In light of these complexities, the need for Explainable AI (XAI) has become vital in these models because of the high-stake nature of decisions made in many critical domains of TSC such as healthcare, finance, and climate science. Especially in these cases, explainability of classification models is necessary to understand and validate the rationale behind a model’s decision. Previous methods for XAI on time-series black-box classifiers are categorized into two categories: 1) Adopting representative perturbation-based XAI methods from

other domains (image, text, tabular) to the time-series domain [13, 28, 29]; and 2) Creating novel time-series specific perturbation techniques [8, 20, 22, 25, 31].

To expand on the first category, there were many representative perturbation-based XAI methods in other domains. However, these methods were not originally designed for the time-series domain, and therefore were only applicable to domains such as tabular, image, or text. In order to utilize these methods for time-series model explanations, various adaptation efforts and rigorous evaluations were required because the nature of time-series data was unlike any other data types. The common feature among the previous works that successfully adopted popular methods from other domains to time-series [13, 28, 29] is that they were primarily model-agnostic perturbation-based XAI methods such as RISE [24], LIME [26], or SHAP [18]. Although these methods captured important features in the time-domain and generated heatmaps according to feature importance weights, they did not consider the frequency component of time-series data.

After this phase of adoption, researchers began to introduce various time-series specific perturbation-based XAI methods. These methods focused on considering the distinct characteristics of time-series data such as temporal dependence [8, 22, 31]. For time-series classifier models, these methods provided better explanation performance compared to previous methods. However, explanations were limited to determining important regions in the time-domain based on temporal patterns and trends. They did not consider frequency components of signals, which contained rich information on transient events or periodic patterns that were potentially influential to the model’s decision-making process.

Time-frequency analysis provides a more comprehensive view of the time-series data by capturing both temporal and spectral features. Most real-world time-series datasets contain time-varying spectral properties, which implies the frequency components change over time. In other words, utilizing time-frequency features for model decision explanation generates better time-series explanation than solely relying on time-domain features. An example is shown in Figure 1, which depicts a binary class time-series dataset consisting of same important time-regions but different frequency components for each class. The time-domain explanation highlights the same regions in class A and class B as important, which does not provide valuable class-specific explanation. On the other hand, the time-frequency explanation shows the impact of the most important frequency component for each class by visualizing the difference between with and without the most important frequency component in the time-domain signal. Similar to this example, there are many cases where the frequency band is a crucial feature in distinguishing between classes for a time-series classification model. For example in sleep EEG dataset [15], the sleep stage classes are distinguished by frequency ranges, and in ECG dataset [21], Atrial fibrillation contains different frequency range compared to normal.

In this paper we propose Spectral eXplanation, an XAI framework to provide time-frequency explanations of time-series black-box classifier models. SpectralX enables any perturbation-based XAI method (e.g. LIME, KernelSHAP, RISE) or black-box classifier models (e.g. LSTM, CNN, Transformer) to “plug-and-play” in our framework to determine important time-frequency explanations.

Through our framework we are able to determine the critical time-series regions by observing the frequency components over the change in time. Given a time-series classification dataset, SpectralX performs a Short-Time Fourier Transform (STFT), search for important class-wise features in the time-frequency domain via feature perturbation, then perform an inverse operation to revert back to the original raw time-series domain to determine class probabilities of class-specific perturbations. Ultimately, SpectralX maintains the black-box classifier model input to be raw time-series signals, and output to represent probability logits of each class of the time-series dataset.

In addition to SpectralX, we propose a set of new perturbation-based XAI methods named Feature Importance Approximations (FIA). The FIA consist of *insertion*, *deletion*, and *combined* methods. The main objective of these methods is to approximate the importance of features in the black-box model’s decision-making process by quantifying the significance of features in contributing to the model’s prediction. In insertion, this is approximated by progressively introducing significant features into a baseline version of the data, and observing the impact on the model’s output. The deletion method progressively removes features from the original data, and highlights features whose absence most negatively affect the model predictions. The combined method takes the absolute difference between both insertion and deletion methods, considering both positive impact when inserted and the negative impact when deleted to determine important features.

Our contributions can be summarized as follows:

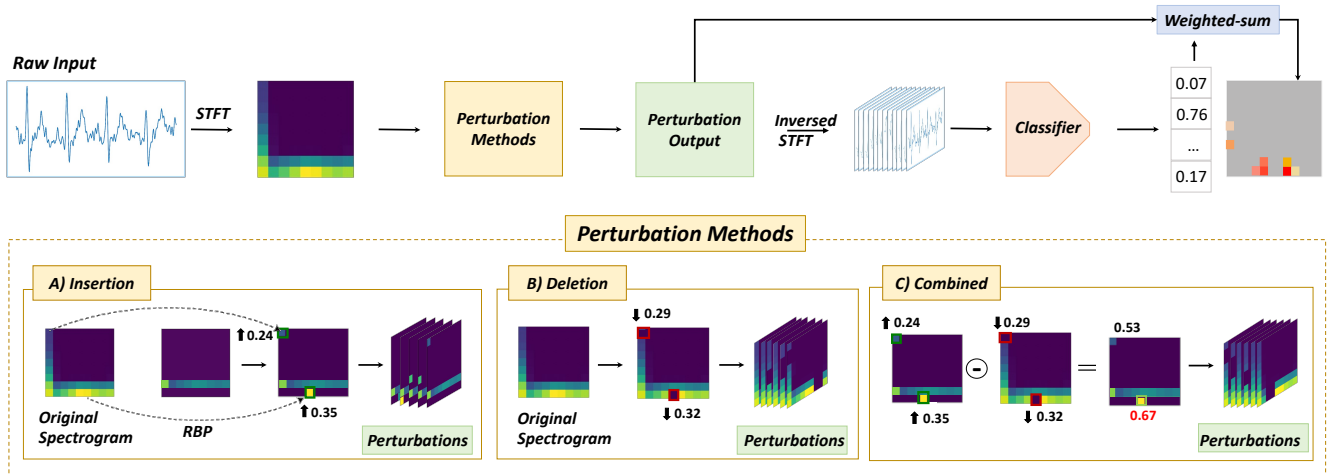
- To the best of our knowledge we propose the first general XAI framework, SpectralX, leveraging both time and frequency information to explain black-box time-series classifiers for various time-series datasets. Our framework facilitates straightforward integration of diverse perturbation-based XAI methods and classifiers to assess spectral explanation performance.
- We introduce FIA, which surpasses existing sample-wise perturbation-based XAI methods in efficiency and effectiveness. The FIA is designed to be class-specific, thus enhancing computational performance with direct approximations of important features in each class.
- We perform extensive experiments across a vast array of time-series datasets to empirically validate the effectiveness of SpectralX and the FIA. These experiments demonstrate significant improvements enabled by our work over existing methods in terms of time-series explanation.
- We conduct a user study to evaluate the practical applicability and user perception of FIA. The results of this study affirm the practicality of our proposed perturbation method, highlighting their potential impact in real-world applications.

check it

## 2 Related Work

XAI methods for time-series models can generally be divided into three different categories: Backpropagation-based methods, attention-based methods, and perturbation-based methods.

For the first two categories, there were many notable studies such as Layer-wise Relevance Propagation (LRP) [2], which decomposed a neural network’s output through the layers, Integrated Gradients [32], which quantified importance of model’s prediction



**Figure 2: (Top) SpectralX Framework and Feature Importance Approximations (FIA).** The raw signals are converted to time-frequency representations with Short-Time Fourier Transform (STFT), and a single perturbation-based XAI method is used to generate perturbation outputs. After reverting to perturbed time signal with Inverse STFT, the classifier determines probability scores to determine important class-specific time-frequency features. **(Bottom)** The three perturbation-based XAI methods represent our FIA. The Insertion method introduces significant features into RBP (Realistic Background Perturbation), a global property of the signal, the Deletion method removes significant features from the original signal, and the Combined method merges the two methods. Numeric values represent increase or decrease in class probability for each method.

by integrating the gradients, and Grad-CAM [30] that produced heatmap visualizations with gradients for CNNs. For time-series models, Dynamic Masking [4] used backpropagation to selectively focus on relevant portions of the data, allowing models to adaptively emphasize important temporal patterns. Also, attention was used to interpret a CNN-LSTM hybrid TSC model in [14], and temporal contextual layers were introduced for explanation in TSC [11, 35]. However all these studies require a specific model architecture, or bear a clear constraint (e.g. must be a CNN-based model, cannot support decision tree-based models, etc.). In this work, we aim to develop a model-agnostic XAI framework for time-series classifiers, which naturally sets our focus on perturbation-based methods. Therefore, for the rest of this work, we mainly deal with perturbation-based methods.

For perturbation-based methods, representative methods in the general domain were adapted to the time-series domain. There were numerous studies of perturbation-based methods in the general domain [5, 7, 18, 19, 24, 26, 27]. Among these approaches, methods like LIME [26], SHAP [18], and RISE [24] have been prominently acknowledged for their contributions to the understanding of model explainability.

LIME [26] presents a model-agnostic black-box strategy by creating random samples around an instance to be explained, and construct simpler, linear decision models. This approach aids in clarifying complex models through local interpretability. SHAP [18] merges concepts from cooperative game theory with local explanations. It assigns the prediction output to the contributions of each feature for a given instance, offering a detailed breakdown of feature importance in model predictions. RISE [24] is particularly useful in interpreting predictions of black-box models in visual data. It

works by randomly masking parts of the input image and observing the changes in the model’s output. This results in a saliency map that highlights key regions in the input that significantly influence the model’s decisions.

While these methodologies have provided valuable insights, directly applying them to time series classifiers neglected the unique aspects of time series data, notably their temporal dependence. This risked misleading interpretations of model’s decisions. Recognizing this gap, the focus has recently shifted towards developing perturbation methods that are specifically tailored for time series classifiers. For instance, the study by Neves et al. [22] proposed enhancing time series explainability by integrating raw signals with their derivatives. Also, the authors in LIMESegment [31] adapt LIME for time series analysis, utilizing NNSegment, harmonic perturbation, and Dynamic Time Warping. LEFTIST [8] also presented local time series classifiers, focusing on a systematic approach that includes segmenting components, generating neighbors.

Although most existing research in this field focuses solely on the time-domain, it’s equally important to take into account both time and frequency. The rationale behind this is that frequency, just like time, is a fundamental and crucial property inherent in time series data [23], [36], [38]. Also, most real-world time-series datasets contain dynamic frequency behaviors, which can easily be tracked in the time-frequency domain. This is because dynamic frequency behaviors refer to frequency change over time, which is more easily identifiable in time-frequency domain with observation of transient behaviors and periodic patterns. Therefore, studies have emerged that focus on time-frequency, but the research by Raab et al. [25] is limited to EEG data and has the drawback of requiring separate black-box classifiers trained for temporal and

spatio-temporal domain explanation. Moreover, Mishra et al. [20] focused their research solely on audio data, which hinders the model's adaptability to a wider array of general time-series dataset explanations.

Our main objective of the SpectralX framework is to create a versatile XAI framework that can integrate any TSC model for time-frequency explanations in general time-series datasets. Therefore, as mentioned above, we utilize various perturbation-based methods in our framework to experimentally compare time-frequency explanations in diverse time-series datasets. Furthermore, we present a novel FIA that improves upon other baseline perturbation-based methods.

### 3 SpectralX

The SpectralX framework is designed to provide spectral explanations for time-series classifiers trained on raw signal data. The highly adaptable characteristic of this framework enables users to “plug-in” various perturbation-based XAI methods and pre-trained TSC models for time-frequency explanations. The extensive performance evaluations across diverse perturbation methods<sup>2</sup> and classifier models ensure results are robust and generalizable. The overall framework is shown in Figure 2. In the following sections we will explore the step-by-step process of SpectralX in detail.

#### 3.1 Short-Time Fourier Transform

The dynamic frequency behavior of real-world time-series datasets diversify the frequency component within each time-series data. Therefore, analyzing the change in frequency over time is crucial in determining class-specific explainable characteristics in time-series datasets. Fortunately, the Short-Time Fourier Transform (STFT) captures the time-varying frequency content of time-series signals. In SpectralX, the STFT is employed as a preliminary step before applying a perturbation method. The equation for STFT is given as:

$$S[m, k] = \sum_{n=0}^{N-1} x[n + mH] \cdot w[n] \cdot e^{-j2\pi \frac{k}{N} n} \quad (1)$$

where  $S$  is the total STFT output and  $S \in \mathbb{C}^{M \times K}$ ,  $M$  representing the number of total time-segments and  $K$  representing the total number of frequency bins. Also,  $N$  is the number of samples in the window,  $w[n]$  denotes the window function applied to  $n$ , which is within the denoted portion of the signal. The term  $x[n + mH]$  indicates that for each time segment  $m$ , multiplied by the hop size  $H$ , the window is applied to a different segment of the signal. The exponential term  $e^{-j2\pi nk/N}$  is the Fourier Transform kernel for a discrete frequency  $k$ . Essentially, the STFT output is the frequency content of the signal over time.

#### 3.2 Perturbation-based XAI Methods

As mentioned in the Related Work section, our primary objective of creating a framework that supports any TSC model necessitates the use of perturbation methods out of the three different categories of time-series explanation methods. Therefore, SpectralX framework

is specifically designed to integrate any perturbation method with any TSC model.

Following the transformation of raw time-series data into time-frequency representations via STFT, SpectralX employs a perturbation method to modify time-frequency components of them. The main goal of this process is to assess the impact of these perturbations on particular regions of the STFT output (*i.e.* time-frequency representation), identifying the most influential areas. In random regions in the STFT output, a predetermined number of different perturbations are applied. Afterwards, these perturbations are reverted back to the time-domain to determine the class probability in the pre-trained black-box classifier. The initial probability of the original data is compared to the perturbed probabilities to determine salient features.

#### 3.3 Inverse Short-Time Fourier Transform

The reverse process from time-frequency domain to time domain is accomplished through the inverse Short-Time Fourier Transform (ISTFT). During this process, it is crucial to ensure perfect reconstruction. Otherwise, noise generated during conversion can interfere with explanations. There are three conditions that must be met in order to ensure perfect reconstruction. First, the Overlap-Add (OLA) condition requires 50 percent overlap between window segments for the Hann window [9], which is the smoothing window function used for both STFT and ISTFT. Second, the window function used in the STFT must match the window function used in ISTFT. Finally, both magnitude and phase information must be correctly incorporated as input to the ISTFT. With all these conditions met, our ISTFT produces time-domain signals with no information loss or distortion, except the alterations from time-frequency perturbations. The equation for ISTFT is as follows:

$$\hat{x}[n] = \frac{1}{N} \sum_{m=0}^{M-1} \sum_{k=0}^{N-1} S[m, k] \cdot w[n - mH] \cdot e^{+j2\pi \frac{k}{N} n} \quad (2)$$

Here,  $\hat{x} \in \mathbb{R}^L$ , where  $L$  represents the total one-dimensional length of the reconstructed time-series signal.  $M$  is the total number of windows in the STFT, and all other variables are the same as in STFT. After the ISTFT process, perturbations reverted to time-domain signals are used as inputs to the black-box TSC classifier, to determine important time-frequency features in various ways depending on the perturbation methods.

### 4 Feature Importance Approximation

The impact of feature insertion and deletion towards the change in class probability served as a key indicator of the quality of explanations [12]. Several evaluation metrics for explanation were proposed based on these probability variations [6, 24, 33]. We propose Feature Importance Approximations (FIA) based on this concept of feature insertion and deletion, where steep rise in class probability after insertion and sharp drop in class probability after deletion indicate important class-specific features and good explanation. Additionally, we propose a combination between insertion and deletion methods, where the weighted absolute difference between insertion and deletion probability changes are selected. The FIA is

<sup>2</sup>We use interchangeably, perturbation methods, perturbation-based methods, and perturbation-based XAI methods.



**Table 1: Average Faithfulness comparison of various methods in Time and Time-Frequency Domain for all nine UCR datasets. Mean and standard deviation are shown across three classifiers. The boldface highlights the best performance in each domain.**

Average Faithfulness		
Methods	Time	Time-Frequency
<i>Baselines</i>		
LIME	0.139 ± 0.05	0.146 ± 0.06
KernelSHAP	0.131 ± 0.02	0.128 ± 0.06
RISE	0.135 ± 0.04	0.143 ± 0.05
LIMESegment	<b>0.148 ± 0.01</b>	–
<i>FLA (ours)</i>		
Insertion	0.107 ± 0.07	0.115 ± 0.08
Deletion	<b>0.148 ± 0.04</b>	0.149 ± 0.08
Combined	0.147 ± 0.04	<b>0.154 ± 0.08</b>

**Table 2: Average Robustness comparison of various methods for top-8 features in Time and Time-Frequency Domain for all nine UCR datasets. Mean and standard deviation are shown across three classifiers. The boldface highlights the best performance in each domain.**

Average Robustness		
Methods	Time	Time-Frequency
<i>Baselines</i>		
LIME	0.58 ± 0.12	0.69 ± 0.14
KernelSHAP	0.59 ± 0.13	0.70 ± 0.13
RISE	0.56 ± 0.11	0.68 ± 0.16
LIMESegment	<b>0.60 ± 0.15</b>	–
<i>FLA (ours)</i>		
Insertion	0.56 ± 0.18	0.70 ± 0.14
Deletion	0.58 ± 0.16	0.69 ± 0.18
Combined	0.57 ± 0.11	<b>0.72 ± 0.12</b>

conducted for all samples in the test set that is in the target explanation class, and is visualized in Figure 2. We will describe in-detail the insertion, deletion, and combined methods.

#### 4.1 Insertion

The insertion method is designed to determine the significance of time-frequency features in enhancing the prediction accuracy of a target class by iteratively inserting features. This method begins with the initial state where the original signal is converted to the baseline representation, which is Realistic Background Perturbation (RBP) [31]. The RBP represents a global property of the signal in the time-frequency representation, and is generated by identifying the frequency band with highest representation and lowest variance. The dimension of the RBP output is the same as the STFT output, which is  $M \times K$ . This reflects the true background content of time-series signals, enforcing meaningful perturbations. We denote the function as  $RBP(\cdot)$ , which is the RBP function to generate baseline representations.

During the insertion phase, we generate diverse perturbations by selecting random features. The selected features remain consistent across all samples within the batch to apply batch parallel processing. This allows for more efficient computation compared

to other methods such as LIME [26], SHAP [18], or RISE [24] that either perturb or select different set of random features for each sample. After applying ISTFT, the probability of the target class is calculated using the black-box classifier  $C$ . The notation and steps for insertion are as follows:

$F$  is the set of all features (i.e., entries) in  $S$ , and  $F_{\text{selected}}$  is the set of features selected. Initially, we apply the  $RBP$  function to the original signal  $x$  to obtain  $RBP(x)$  which is  $S_{\text{RBP}}$ . Then, for each iteration  $i$  where  $i = 1, 2, \dots, F$ , we generate  $P$  different random binary perturbation masks,  $U^{(p)}$ , for  $p = 1, 2, \dots, P$ . Each binary perturbation mask has  $R$  time-frequency regions unmasked (set to 1) while the rest are masked (set to 0). We apply each binary perturbation mask to the RBP representation to create perturbed versions,  $S_{\text{ins-per}}^{(i,p)}$ , for each perturbation  $p$  at iteration  $i$ . The formula is as follows:

$$S_{\text{ins-per}}^{(i,p)} = S \odot U^{(i,p)} + S_{\text{RBP}} \odot (1 - U^{(i,p)}) \quad (3)$$

Afterwards, we convert each perturbation back to the time domain using ISTFT to get  $\hat{x}_{\text{ins-per}}^{(i,p)}$ . We pass each perturbed signal through the black-box classifier  $C$  to obtain class probability scores. The target class probability score of the original signal is denoted as  $P_{\text{class}}^{\text{original}}$ , and the target class probability score of the perturbed signal is  $C(\hat{x}_{\text{ins-per}}^{(i,p)})$  which is represented as  $P_{\text{class}}^{\text{perturbed}}$ . The features that are not already selected, represented by  $f \in F \setminus F_{\text{selected}}$ , are the features considered for importance scoring in iteration  $i$ . We calculate the final score based on the influence on the target class probability scores, which is:

$$\text{Score}_{\text{ins-f}}^{(i)} = \frac{\sum_{p=1}^P \delta(U^{(i,p)}, f) \cdot (P_{\text{class}}^{\text{perturbed},f} - P_{\text{class}}^{\text{original}})}{\sum_{p=1}^P \delta(U^{(i,p)}, f)} \quad (4)$$

where  $\delta(U^{(i,p)}, f)$  is an indicator function that equals 1 if the feature is unmasked in the  $p$ -th perturbation, and 0 otherwise. Finally, we select the feature with the highest score:

$$f_{\text{max}}^{(i)} = \arg \max_{f \in F \setminus F_{\text{selected}}} \text{Score}_{\text{ins-f}}^{(i)} \quad (5)$$

and update  $F_{\text{selected}}$  and  $F$  for the next iteration as follows:

$$F_{\text{selected}}^{(i+1)} = F_{\text{selected}}^{(i)} \cup \{f_{\text{max}}^{(i)}\}, \quad F^{(i+1)} = F^{(i)} \setminus \{f_{\text{max}}^{(i)}\} \quad (6)$$

#### 4.2 Deletion

The deletion method focuses on assessing the impact of removing time-frequency features. It emphasizes the importance of features whose absence significantly degrades model performance. Instead of starting with RBPs, the deletion method starts with time-frequency representations of original signals. Also, instead of unmasking specific features, the deletion method masks  $R$  different regions to create  $P$  different perturbations. Apart from this, all other notations and steps are similar to the insertion method. The steps are as follows:

We start with the STFT output of the signal, denoted as  $S$ . For each iteration  $i$  (where  $i = 1, 2, \dots, F$ ), we generate  $P$  different random masked features,  $M^{(p)}$ , for  $p = 1, 2, \dots, P$ . Each perturbation

**Table 3: Rank-Biased Overlap@k and Area Under Curves comparison of various methods in the Time-Frequency Domain for the Synthetic Dataset. Mean and standard deviation are shown across three classifiers. The boldface highlights the best performance in each @k, Area Under Precision (AUP), and Area Under Recall (AUR).**

	Rank-Biased Overlap@k					Area Under Curves	
	@1	@2	@4	@6	@8	AUP	AUR
<i>Baselines</i>							
LIME	0.311 ± 0.12	0.311 ± 0.13	0.471 ± 0.10	0.515 ± 0.10	0.545 ± 0.11	0.489 ± 0.05	0.431 ± 0.06
KernelSHAP	<b>0.422 ± 0.19</b>	0.367 ± 0.14	0.445 ± 0.02	0.465 ± 0.04	0.466 ± 0.08	0.444 ± 0.06	0.433 ± 0.07
RISE	0.2 ± 0.03	0.228 ± 0.05	0.357 ± 0.13	0.416 ± 0.12	0.447 ± 0.11	0.391 ± 0.11	0.330 ± 0.07
<i>FIA (Ours)</i>							
Insertion	0.225 ± 0.05	0.083 ± 0.08	0.206 ± 0.12	0.318 ± 0.13	0.407 ± 0.12	0.293 ± 0.08	0.248 ± 0.06
Deletion	0.333 ± 0.07	0.406 ± 0.05	0.433 ± 0.05	0.474 ± 0.11	0.491 ± 0.09	0.467 ± 0.07	0.427 ± 0.12
Combined	0.333 ± 0.03	<b>0.444 ± 0.12</b>	<b>0.527 ± 0.12</b>	<b>0.574 ± 0.12</b>	<b>0.580 ± 0.08</b>	<b>0.553 ± 0.10</b>	<b>0.491 ± 0.04</b>

has  $R$  time-frequency regions masked (set to 0) while the rest are unmasked (set to 1). Then, for each mask  $p$  at iteration  $i$ :

$$S_{\text{del-per}}^{(i,p)} = S \odot (1 - M^{(i,p)}) + S_{\text{RBP}} \odot M^{(i,p)} \quad (7)$$

The time-frequency perturbation is reverted back to the time-domain signal  $x_{\text{del-per}}^{(i,p)}$  with ISTFT. We pass each perturbed signal through the black-box classifier to obtain target class probability scores,  $P_{\text{class}}^{(i,p)}$ , for each perturbation. The change in probability scores, compared to the original signal's target class probability score,  $P_{\text{class}}^{\text{original}}$ , indicates the importance of the masked regions. The final score for each feature masked in iteration  $i$  is calculated as follows:

$$\text{Score}_{\text{del-f}}^{(i)} = \frac{\sum_{p=1}^P \delta(M^{(i,p)}, f) \cdot (P_{\text{class}}^{\text{perturbed}, f} - P_{\text{class}}^{\text{original}})}{\sum_{p=1}^P \delta(M^{(i,p)}, f)} \quad (8)$$

where  $\delta(M^{(i,p)}, f)$  is an indicator function that equals 1 if the feature is unmasked, and 0 otherwise in the  $p$ -th perturbation. We select the feature with the lowest score:

$$f_{\min}^{(i)} = \arg \min_{f \in F \setminus F_{\text{selected}}} \text{Score}_{\text{del-f}}^{(i)} \quad (9)$$

and update  $F_{\text{selected}}$  and  $F$  for the next iteration as follows:

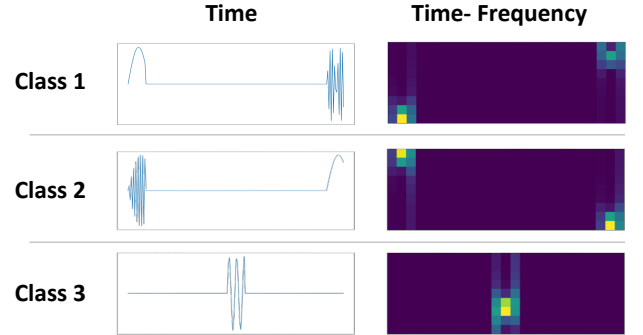
$$F_{\text{selected}}^{(i+1)} = F_{\text{selected}}^{(i)} \cup \{f_{\min}^{(i)}\}, \quad F^{(i+1)} = F^{(i)} \setminus \{f_{\min}^{(i)}\} \quad (10)$$

### 4.3 Combined

The combined method integrates both insertion and deletion methods. It first runs insertion and deletion method in parallel, going through the ISTFT, perturbation process, and importance scoring calculation in both methods. Then, to determine the final score in the combined method for each iteration, the weighted absolute difference between the final scores from insertion and deletion methods are calculated. The time-frequency feature that produces the greatest weighted absolute difference score is the selected feature for the iteration  $i$ . The formula is as follows:

$$\text{Score}_{\text{comb-f}}^{(i)} = \left| \alpha \cdot \text{Score}_{\text{ins-f}}^{(i)} - (1 - \alpha) \cdot \text{Score}_{\text{del-f}}^{(i)} \right| \quad (11)$$

This combined score effectively merges the impact of both insertion and deletion methods in determining the most important feature. By leveraging the  $\alpha$  value as a hyperparameter, we are able to determine the optimal balance between the two methods for the best explanation performance. Insertion and deletion methods are ran in parallel utilizing this final combined score for both methods in the subsequent steps.

**Figure 3: Samples from synthetic dataset for each class. Time and Time-Frequency representation of each sample.**

## 5 Experiments

In this section, we describe the experimental settings and datasets of our work. Then, we present and analyze the experimental results. We first train the black-box classifiers on the time-series datasets, achieving exceptionally high accuracy, precision, recall, and F1 scores for most models and datasets<sup>3</sup>. Then, we use baseline perturbation methods and FIA in the time-domain to collect the top-1 important time-segments. These segments are collected to compare with important time-frequency features, which are determined with the SpectralX framework. This comparison experiment is conducted to determine the optimal domain for time-series explanation with average Faithfulness, which measures the average decrease in class probability of the black-box classifier when the most important

<sup>3</sup>We intentionally use time-series datasets that allow near-perfect classification, so we can safely focus on model explanation without having to worry about model training.

**Table 4: Faithfulness@k comparison of various methods in the Time-Frequency Domain for nine UCR datasets. Mean and standard deviation are shown across three classifiers. The boldface highlights the best performance in each @k.**

	Faithfulness@k				
	@1	@2	@4	@6	@8
<i>Baselines</i>					
LIME	0.146 $\pm$ 0.06	0.196 $\pm$ 0.08	0.245 $\pm$ 0.06	0.285 $\pm$ 0.09	0.312 $\pm$ 0.07
KernelSHAP	0.128 $\pm$ 0.06	0.147 $\pm$ 0.09	0.204 $\pm$ 0.08	0.212 $\pm$ 0.10	0.278 $\pm$ 0.08
RISE	0.128 $\pm$ 0.06	<b>0.202 <math>\pm</math> 0.08</b>	0.258 $\pm$ 0.09	0.287 $\pm$ 0.09	0.332 $\pm$ 0.07
<i>FIA (Ours)</i>					
Insertion	0.115 $\pm$ 0.08	0.154 $\pm$ 0.09	0.199 $\pm$ 0.09	0.229 $\pm$ 0.07	0.273 $\pm$ 0.04
Deletion	0.149 $\pm$ 0.08	0.189 $\pm$ 0.05	0.238 $\pm$ 0.06	0.292 $\pm$ 0.05	0.333 $\pm$ 0.08
Combined	<b>0.154 <math>\pm</math> 0.08</b>	0.192 $\pm$ 0.06	<b>0.263 <math>\pm</math> 0.06</b>	<b>0.295 <math>\pm</math> 0.08</b>	<b>0.343 <math>\pm</math> 0.06</b>

feature is removed for each class, and average Robustness, which is the difference in explanation before and after perturbation with randomly generated noise.

Afterwards, given the domain that demonstrated superior performance, we conduct extensive experiments in order to further compare the explanation performance of the baseline perturbation methods and FIA. In the domain of superior performance, we first generate a synthetic dataset to compare each method’s predicted explanations to the ground-truth explanations. Then, for real-world time-series datasets, the average Faithfulness@k, which measures the average Faithfulness value for  $k$  different top-ranked features, is measured. This shows the consistency of explanations for the top- $k$  importance rankings from each method.

Finally, we conduct human evaluation with twenty graduate students studying machine learning to rank the explanation quality between FIA and other perturbation methods for each class of seven different time-series datasets. We instruct the users to rank the explanation that captures the essence of each target class.

## 5.1 Experimental Settings

We used three different black-box classifiers: 2-layer bi-directional LSTM, 1D ResNet-34, and 2-layer Transformer model. For all methods we used batch size of 64 and learning rate of  $2e-4$ . The Transformer model contained Transformer-encoder layers with model dimension of 64, feedforward dimension 256, and 8 self-attention heads. The dimension of model was also 64 for LSTM. For the configuration of baseline and FIA,  $P$ , which is the number of perturbations, is set to 2000. For RISE and FIA,  $R$ , which is the number of masked regions, is set to 10. Finally, for the combined method in FIA, the insertion weight was set to 0.2 ( $\alpha$ ) and deletion weight was set to 0.8 ( $1 - \alpha$ ). For STFT and ISTFT we used window size of 16, hop size of 8, and the Hann window. All experiments were conducted with NVIDIA GeForce RTX 3090 GPUs.

## 5.2 Datasets

We generate a synthetic dataset in which the discriminative time-frequency features are known and considered as ground-truth features. As shown in Figure 3, the synthetic dataset contains three classes, with each class representing a different frequency range in each time position. Specifically, class 1 samples contain low frequency with less than two oscillations in the first 16 timesteps, and

high frequency in the last 16 timesteps with more than 10 oscillations. Class 2 samples have high frequency in the first 16 timesteps, and low frequency in the last 16 timesteps. Class 3 samples contain mid frequency in the mid 16 timesteps with more than four and less than 10 oscillations. Important regions in the time-frequency domain are all different, but the important regions in the time domain overlap between classes 1 and 2. For samples in each class, random Gaussian noise were added to base sinusoidal signal for variability across different samples while maintaining frequency. 1000 samples were generated for each class to create total of 3000 samples.

We also conduct experiments on nine different datasets of UCR repository [3]. The datasets consist of mixture of multi-class and binary-class datasets, which are: CincECGTorso, Twopatterns, Mixed-shapes, Arrowhead, Strawberry, Yoga, Ford A, Ford B, and Gunpoint-MaleFemale. We combine the train and test splits from the synthetic dataset and UCR repository datasets and randomly split the datasets into train (80%), validation (10%), and test (10%) sets.

## 5.3 Baselines

We consider the following four baseline perturbation methods for integration into SpectralX and comparison to FIA:

First, LIME [26], whose flexibility and wide applicability makes it a standard for assessing the local fidelity of explanations. Second, KernelSHAP, a specific implementation of SHAP that uses a kernel-weighted local regression to estimate Shapley values, which employs efficient optimization and approximation of SHAP values. Third, RISE [24], whose simplicity and effectiveness in identifying important features in the image-domain, serves as a pertinent baseline to highlight critical features in the time-frequency domain, which contains many similarities to the image-domain. Finally, we compare LIMESegment [31] in the time-domain, which is the current state-of-the-art method for model-agnostic time-series local explanations.

## 5.4 Time and Time-Frequency Domain

In Table 1, we present the average Faithfulness results with standard deviations in both time and time-frequency domains for baseline perturbation methods and FIA. When determining the most important explanations, we consider a group of timesteps as a single feature, also known as super-segments. we fix the number of timesteps



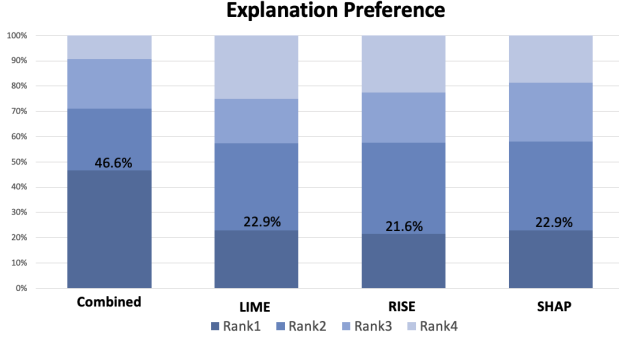


Figure 4: Human evaluation results

to 16 for each super-segment in the time-domain, which is the same as the window size used during STFT for time-frequency domain conversion.

Note that perturbation in the time-domain has an inherent advantage over perturbation in the time-frequency domain in influencing the model output (*i.e.* class probability). This is because during perturbation, the entire super-segment is masked to zero in the former, but only a specific frequency range is masked in the latter. Therefore, the former always makes a larger change in the original signal than the latter. Despite this disadvantage, all methods show competitive performance in the time-frequency domain compared to the time domain, meaning that we were able to obtain only the essence of each class by minimally manipulating the original signal. We hypothesize that this is due to important features from the time-frequency domain capturing the class-specific essential regions better than the important features from the time domain. With the exception of KernelSHAP, the average Faithfulness in time-frequency domain performs better than time-domain for all perturbation methods.

Table 2 shows the average Robustness values with standard deviations in both time and time-frequency domain for all methods. Gaussian noise is added to raw time-series data in the time domain and to the magnitude of STFT output in the time-frequency domain. The Robustness scores represent the proportion of unchanged super-segment explanations for the time-domain, and proportion of unchanged time-frequency explanations for the time-frequency domain. To calculate the Robustness scores, we used the top-8 features from each method for a clearer comparison of performance across the time and time-frequency domains. Considering fewer than 8 features made it challenging to determine the superior domain. For all methods, average Robustness outperforms in time-frequency domain compared to the time-domain. From the average Faithfulness and Robustness results, we can conclude that the time-frequency domain is the better performing domain for time-series explanations.

## 5.5 Time-Frequency Evaluations

As a more direct and objective measure of the accuracy of the predicted explanations, we use the generated synthetic dataset to compare each method’s predicted explanations to the ground-truth explanations. The ground-truth explanations are synthesized by

**Table 5: Average Faithfulness comparison for different  $\alpha$  values in Time and Time-Frequency Domain for all nine UCR datasets. Mean and standard deviation are shown across three classifiers. The boldface highlights the best performance in each domain.**

Average Faithfulness		
$\alpha$	Time	Time-Frequency
0.8	0.115 $\pm$ 0.06	0.122 $\pm$ 0.06
0.5	0.131 $\pm$ 0.06	0.133 $\pm$ 0.05
0.2	<b>0.147 <math>\pm</math> 0.04</b>	<b>0.154 <math>\pm</math> 0.08</b>

the descending magnitude order, from greatest to smallest above zero magnitude. Figure 3 shows a sample from each class of the synthetic dataset, where the prominent features in both time and time-frequency domains are displayed. The rankings between the ground-truth and predicted explanations are compared with Rank-Biased Overlap (RBO) [37], which takes into account the rank positions in final score calculations. The formula of RBO is as follows:

$$RBO(L_1, L_2, d) = (1 - \lambda) \sum_{k=1}^d \lambda^{k-1} \frac{|L_{1:k} \cap L_{2:k}|}{k} \quad (12)$$

where  $L_1$  and  $L_2$  are the two ranked feature lists,  $d$  represents the depth of the lists to compare,  $\lambda$  is the parameter to control the emphasis on the top of the lists, which we set to 0.9. Although there are many popular ranking methods such as Kendall Tau [16] and Spearman’s rank correlation coefficient [39], they do not inherently assign higher weights to top ranks in their calculations. It is crucial in our ground-truth and prediction comparison to take into account the order of the top explanation features, and give more weight to the agreement between ground-truth and predicted at the top of the list, which the RBO is able to do. The RBO calculates a score between 0 and 1, where 0 indicates no overlap and 1 indicates identical rankings. The calculation is based on the cumulative overlap up to each depth  $d$  of the explanations, weighted by a factor that decreases the importance of items further down the list. Additionally, we measure the quality of explanations by measuring the proportion of predicted features that are indeed the ground-truth with area under the precision curve (AUP), and measure the portion of ground-truth features identified with area under the recall curve (AUR). For all metrics, higher is better. The results are shown in Table 3, where combined method in FIA scores above other methods in most cases.

We then measure the Faithfulness@ $k$  in the nine datasets of the UCR repository. Table 4 presents the average Faithfulness@ $k$  that shows the average Faithfulness values for  $k$  regions. This metric shows the consistency of explanations not only in the top-1, but also up to the top-8 explanations. The combined method in FIA shows best performance in most  $k$  values except for when  $k$  is 2. This indicates its superiority in various explanation granularity.

## 5.6 User Study

We conducted a user study to assess how well FIA combined method captures class-wise explanations for UCR time-series datasets [3]. Each question presented a selection of options featuring plots of class average values of each dataset, and the top-1 important feature

**Table 6: Average Faithfulness@k values for different methods and window/hop sizes in the CinCECGTorso dataset. Mean and standard deviation are across three classifiers. Boldface highlights the best performance for each @k and window/hop size.**

		Faithfulness@k				
Win./Hop		@1	@2	@4	@6	@8
<i>Baselines</i>						
LIME	64/32	<b>0.064 ± 0.04</b>	0.098 ± 0.08	0.291 ± 0.09	0.577 ± 0.03	0.688 ± 0.05
	128/64	0.155 ± 0.02	<b>0.340 ± 0.06</b>	0.508 ± 0.03	0.681 ± 0.02	0.810 ± 0.07
KernelSHAP	64/32	0.059 ± 0.06	0.104 ± 0.07	0.295 ± 0.08	0.541 ± 0.06	0.653 ± 0.07
	128/64	0.150 ± 0.01	0.332 ± 0.04	0.502 ± 0.04	0.695 ± 0.04	0.834 ± 0.05
RISE	64/32	0.062 ± 0.08	0.092 ± 0.08	0.274 ± 0.09	0.562 ± 0.09	0.671 ± 0.07
	128/64	0.151 ± 0.02	0.328 ± 0.06	0.496 ± 0.02	0.674 ± 0.04	0.818 ± 0.06
<i>FIA (Ours)</i>						
Insertion	64/32	0.055 ± 0.09	0.101 ± 0.06	0.291 ± 0.05	0.560 ± 0.06	0.647 ± 0.04
	128/64	0.148 ± 0.01	0.325 ± 0.05	0.485 ± 0.03	0.659 ± 0.05	0.802 ± 0.04
Deletion	64/32	0.062 ± 0.06	0.105 ± 0.07	<b>0.313 ± 0.06</b>	0.575 ± 0.05	0.683 ± 0.08
	128/64	0.153 ± 0.02	0.336 ± 0.06	0.514 ± 0.05	0.703 ± 0.07	0.842 ± 0.03
Combined	64/32	0.060 ± 0.05	<b>0.108 ± 0.05</b>	0.308 ± 0.08	<b>0.583 ± 0.04</b>	<b>0.691 ± 0.06</b>
	128/64	<b>0.159 ± 0.02</b>	0.332 ± 0.03	<b>0.524 ± 0.04</b>	<b>0.725 ± 0.04</b>	<b>0.855 ± 0.04</b>

produced by each method in the time-frequency domain masked and reverted to the time-domain. Explanation samples are shown at this [repository](#). Figure 4 presents the ranking results of the explanation preference for each method across a total of seven datasets and 19 classes. We excluded the Mixedshapes and Ford B datasets from the user study because the important features for each class in these datasets were very difficult to distinguish, and including them would have introduced significant noise into the user study. Users ranked the explanation preference of each region highlighted by the four methods from first to fourth place. Ties were allowed for multiple choices. The combined method achieved the highest rank 1 proportion of 46.6%, followed by LIME and SHAP which were tied at 22.9% and then RISE with rank 1 ratio of 21.6%.

## 5.7 Ablation Study

We explore the  $\alpha$  values for FIA combined method, and various window and hop sizes for STFT and ISTFT. In Table 5, lower  $\alpha$  values for combined method indicate less weight towards the insertion method, and more weight towards the deletion method according to Equation (11). The results show that decrease in the insertion weight and increase in the deletion weight increases the average faithfulness values for both time and time-frequency domains. However,  $\alpha$  value of 0.2 outperforms  $\alpha$  value of 0 (deletion method) as shown in Table 4. We hypothesize a slight bias towards the deletion method without completely losing the complementary insights from the insertion method produces the optimal synergy that can leverage the strengths of both deletion and insertion methods.

We observe average Faithfulness@k performance of various methods for different window and hop size values during STFT. We conduct experiments on the CinCECGTorso dataset, which contains the longest time length samples among the nine different datasets. The other datasets are too short to provide meaningful performance evaluations with gradual increase in window and hop size. This variation in window and hop sizes aims to demonstrate that the performance remains relatively consistent with the fixed window and hop size hyperparameters used in the previous experiments,

with the combined method showing optimal performance in most @k values. The results are presented in Table 6.

## 6 Discussion and Future Work

There are some limitations in our experiments. First, in SpectralX, we did not explore the optimal hyperparameters in-depth for the STFT window function (we used Hann window in all our experiments), which can possibly alter performance depending on the time-series dataset. Second, FIA provides class-specific explanations for the target class but lacks detailed explanations for individual samples (compared to local models such as LIME). We also acknowledge that FIA is a generic XAI method capable of being applied across a broader range of domains beyond TSC. This is a promising avenue for future research, therefore as future work we propose to extend FIA to other domains such as Natural Language Processing or Computer Vision to observe the efficacy in a broader context.

Work on it

## 7 Conclusion

In this work, we present a time-frequency based explanation framework, SpectralX, for time-series black-box model explanations. In addition, we propose FIA, which are insertion, deletion, and combined methods. We first compare the performance of various methods in the time and time-frequency domain to show the advantages of time-frequency based explanations, and then compared our proposed combined method in FIA with other baseline methods in the time-frequency domain. By conducting rigorous quantitative evaluations with diverse metrics and a user study to demonstrate the efficacy of our framework, our combined method in FIA showed superior performance compared to other baseline methods.

## 8 Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No.RS-2019-II190075, No.RS-2022-II220984), and National Research Foundation of Korea (NRF) grant (NRF-2020H1D3A2A03100945), funded by the Korea government (MSIT).

## References

- [1] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE access* 6 (2018), 52138–52160.
- [2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS one* 10, 7 (2015), e0130140.
- [3] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. 2015. The UCR Time Series Classification Archive. [www.cs.ucr.edu/~eamonn/time\\_series\\_data/](http://www.cs.ucr.edu/~eamonn/time_series_data/).
- [4] Jonathan Crabbé and Mihaela Van Der Schaar. 2021. Explaining time series predictions with dynamic masks. In *International Conference on Machine Learning*. PMLR, 2166–2177.
- [5] Jamie Duell, Xiuyi Fan, Bruce Burnett, Gert Aarts, and Shang-Ming Zhou. 2021. A comparison of explanations given by explainable artificial intelligence methods on analysing electronic health records. In *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE, 1–4.
- [6] Ruth C Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*. 3429–3437.
- [7] Alex Gramegna and Paolo Giudici. 2021. SHAP and LIME: an evaluation of discriminative power in credit risk. *Frontiers in Artificial Intelligence* 4 (2021), 752558.
- [8] Maël Guillemé, Véronique Masson, Laurence Rozé, and Alexandre Termier. 2019. Agnostic Local Explanation for Time Series Classification. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. 432–439. <https://doi.org/10.1109/ICTAI.2019.00067>
- [9] Fredric J Harris. 1978. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proc. IEEE* 66, 1 (1978), 51–83.
- [10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [11] Tsung-Yu Hsieh, Suhang Wang, Yiwei Sun, and Vasant Honavar. 2021. Explainable multivariate time series classification: a deep neural network which learns to attend to important variables as well as time intervals. In *Proceedings of the 14th ACM international conference on web search and data mining*. 607–615.
- [12] Maksims Ivanovs, Roberts Kadikis, and Kaspars Ozols. 2021. Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters* 150 (2021), 228–234.
- [13] Anahid Jalali, Alexander Schindler, Bernhard Haslhofer, and Andreas Rauber. 2020. Machine learning interpretability techniques for outage prediction: a comparative study. In *PHM Society European Conference*, Vol. 5. 10–10.
- [14] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. 2017. LSTM fully convolutional networks for time series classification. *IEEE access* 6 (2017), 1662–1669.
- [15] Bob Kemp, Aeilko H Zwinderman, Bert Tuk, Hilbert AC Kamphuisen, and Josefien JL Obery. 2000. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG. *IEEE Transactions on Biomedical Engineering* 47, 9 (2000), 1185–1194.
- [16] Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika* 30, 1/2 (1938), 81–93.
- [17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.
- [18] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).
- [19] Georgios Makridis, Georgios Fatouros, Vasileios Koukos, Dimitrios Kotios, Dimosthenis Kyriazis, and John Soldatos. 2023. XAI for time-series classification leveraging image highlight methods. In *International Conference on Management of Digital*. Springer, 382–396.
- [20] Saumitra Mishra, Emmanouil Benetos, Bob LT Sturm, and Simon Dixon. 2020. Reliable local explanations for machine listening. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.
- [21] George Moody. 1983. A new method for detecting atrial fibrillation using RR intervals. *Proc. Comput. Cardiol.* 10 (1983), 227–230.
- [22] Ines Neves, Duarte Folgado, Sara Santos, Marília Barandas, Andrea Campagner, Luca Ronzio, Federico Cabitza, and Hugo Gamboa. 2021. Interpretable heartbeat classification using local model-agnostic explanations on ECGs. *Computers in Biology and Medicine* 133 (2021), 104393.
- [23] Thach Le Nguyen, Severin Gsponer, Iulia Ilie, and Georgiana Ifrim. 2018. Interpretable time series classification using all-subsequence learning and symbolic representations in time and frequency domains. *arXiv preprint arXiv:1808.04022* (2018).
- [24] Vitali Petsiuk, Abir Das, and Kate Saenko. 2018. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421* (2018).
- [25] Dominik Raab, Andreas Theissler, and Myra Spiliopoulou. 2023. XAI4EEG: spectral and spatio-temporal explanation of deep learning-based seizure detection in EEG time series. *Neural Computing and Applications* 35, 14 (2023), 10051–10068.
- [26] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “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*. 1135–1144.
- [27] Rohit Saluja, Avleen Malhi, Samanta Knapič, Kary Främling, and Cicek Cavdar. 2021. Towards a rigorous evaluation of explainability for multivariate time series. *arXiv preprint arXiv:2104.04075* (2021).
- [28] Udo Schlegel, Hiba Arnout, Mennatallah El-Assady, Daniela Oelke, and Daniel A Keim. 2019. Towards a rigorous evaluation of XAI methods on time series. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 4197–4201.
- [29] Udo Schlegel and Daniel A Keim. 2023. A Deep Dive into Perturbations as Evaluation Technique for Time Series XAI. In *World Conference on Explainable Artificial Intelligence*. Springer, 165–180.
- [30] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.
- [31] Torty Sivill and Peter Flach. 2022. Limesegment: Meaningful, realistic time series explanations. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3418–3433.
- [32] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*. PMLR, 3319–3328.
- [33] Andreas Theissler, Francesco Spinnato, Udo Schlegel, and Riccardo Guidotti. 2022. Explainable AI for time series classification: a review, taxonomy and research directions. *IEEE Access* (2022).
- [34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [35] Phongtharin Vinayavekhin, Subhagit Chaudhury, Asim Munawar, Don Joven Agravante, Giovanni De Magistris, Daiki Kimura, and Ryuki Tachibana. 2018. Focusing on what is relevant: Time-series learning and understanding using attention. In *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2624–2629.
- [36] Jingyuan Wang, Chen Yang, Xiaohan Jiang, and Junjie Wu. 2023. WHEN: A Wavelet-DTW Hybrid Attention Network for Heterogeneous Time Series Analysis. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2361–2373.
- [37] William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)* 28, 4 (2010), 1–38.
- [38] Zhou Wu and Ruiqi Jiang. 2023. Time-series benchmarks based on frequency features for fair comparative evaluation. *Neural Computing and Applications* (2023), 1–13.
- [39] Jerrold H Zar. 2005. Spearman rank correlation. *Encyclopedia of Biostatistics* 7 (2005).