Visual Explanations and Perturbation-Based Fidelity Metrics for Feature-Based Models

Maciej Mozolewski[1] [0000−0003−4227−3894],
Szymon Bobek[2] [0000−0002−6350−8405], and Grzegorz J. Nalepa[3] [0000−0002−8182−4225]

[1] Jagiellonian Human-Centered AI Lab, Mark Kac Center for Complex Systems Research, Institute of Applied Computer Science, Faculty of Physics, Astronomy and Applied Computer Science, Jagiellonian University, prof. Stanisława Łojasiewicza 11 St., 30-348 Krakow, Poland
m.mozolewski@doctoral.uj.edu.pl
https://github.com/mozo64
[2] szymon.bobek@uj.edu.pl
[3] grzegorz.j.nalepa@uj.edu.pl

Abstract. This work introduces an enhanced methodology in the domain of *eXplainable Artificial Intelligence* (*XAI*) for visualizing local explanations of black-box, feature-based models, such as LIME and SHAP, enabling both domain experts and non-specialists to identify the segments of *Time Series* (*TS*) data that are significant for machine learning model interpretations across classes. By applying this methodology to electrocardiogram (*ECG*) data for anomaly detection, distinguishing between healthy and abnormal segments, we demonstrate its applicability not only in healthcare diagnostics but also in predictive maintenance scenarios. Central to our contribution is the development of the *AUC Perturbational Accuracy Loss metric* (*AUC-PALM*), which facilitates the comparison of explainer fidelity across different models. We advance the field by evaluating various perturbation methods, demonstrating that perturbations centered on time series prototypes and those proportional to feature importance outperform others by offering a more distinct comparison of explainer fidelity with the underlying black-box model. This work lays the groundwork for broader application and understanding of *XAI* in critical decision-making processes.

Keywords: XAI · Visualizations · Anomaly Detection · Time Series · AUC-PALM · ECG · Dynamic Time Warping Barycenter Averaging · Time Series classification · Deep Learning · RNN-autoencoder · reconstruction loss · SHAP · LIME · Healthcare Analytics · Feature Importance · Model Interpretability

1 Introduction

In many applications, domain experts possess expectations regarding the features that should generally influence the classification of cases into specific classes. Our focus is on time series data, particularly with two classes: normal and anomalous. Medical data, such as ECG, which constitutes time series data, serves as an example. However, it is crucial to note that medical data is not limited to time series but also includes images, textual, and tabular data. Analyzing these data types and explaining models working on them is often more straightforward due to their inherent characteristics, which sometimes make visualization and interpretation more accessible. This is true particularly in medical imaging, where visualization and interpretation

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63772-8_27

are more straightforward due to the data's visual nature [11,14]. This ease of analysis contrasts with the complexities of time series data, where explaining model decisions poses significant challenges.

Interpretability in feature-based black-box explainers, such as Local Interpretable *Model-agnostic Explanations* (*LIME*) and *SHapley Additive exPlanations* (*SHAP*), presents a significant challenge, especially when aiming to discern the average influence of features on predictions for a given class. The study by [15] underscores the applications of XAI in healthcare, highlighting the critical need for transparency, fairness, and accuracy in AI-driven decision-making processes.

Visualizing explanations for time series data is particularly challenging due to the complexity of capturing and representing temporal relationships and dynamics effectively. The study by [21] developed a variety of metrics, such as fidelity, monotonicity, stability, and interpretability, to validate and evaluate the effectiveness of explanation techniques. Our visualizations enable experts to ascertain which segments of the series, in this case, ECG, the model focuses on concerning the normal and abnormal classes. This insight is crucial as it allows for a deeper understanding of the model's decision-making process, potentially leading to improved diagnostic and predictive outcomes.

However, visualization alone, i.e., understanding the model mediated by explainers, is insufficient. It's imperative to ensure that our explainers faithfully represent the black-box model they aim to elucidate. We employ a perturbation method, widely recognized in literature, to achieve this. Yet, there are numerous proposals on how to calculate the *AUC Perturbational Accuracy Loss metric* (*AUC-PALM*), especially for time series data where introducing perturbations thoughtfully is paramount. Our method introduces perturbations around the prototype of a given class and examines whether perturbations should be proportional or inversely proportional to feature importance.

In conclusion, our work contributes to the field by providing a methodology that not only enhances the interpretability of black-box models through visual explanations but also ensures the fidelity of these explanations through a novel application of the *AUC-PALM* metric. By focusing on the specific challenges presented by time series data, we offer insights that are broadly applicable, underscoring the importance of continuous research and diverse applications in XAI, as indicated by comprehensive reviews [10,20].

Our main contributions are:

- We highlight the importance of visualizing *Time Series* (*TS*) data to improve understanding with local explainers such as SHAP and LIME, making model interpretations clearer.
- Utilizing a metric described in literature [3], the *AUC Perturbational Accuracy Loss metric* (*AUC-PALM*), we adapted it for time series analysis, allowing for a finer distinction in model fidelity, crucial for better model evaluation. Our developed perturbation methods improve the *AUC-PALM* measure for any XAI algorithm that attributes importance to features. These methods are fitted for the analysis of *TS* data.
- Our method is universally applicable to *Time Series* data, with a special benefit for healthcare due to its extensive use in this field. This is showcased through its application to *ECG* data, providing precise and understandable model explanations.

The paper is structured as follows: Section 2 reviews current TS classifiers research. Section 3 describes our method and a use-case study with the ECG dataset and Deep Incep-

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63772-8_27

tionTime Model. Section 4 presents visualizations and results using *AUC-PALM*. Section 5, and Section 6 concludes the work and future development perspectives.

2 Related works

This section focuses on *XAI evaluation metrics* and *Time Series* (TS) classification. Authors [21], presents a comprehensive overview of evaluation metrics for ML explanations. Metrics for model-based and example-based explanations primarily evaluate interpretability and simplicity, while attribution-based explanations primarily evaluate fidelity and soundness. In [8], the authors discuss methods for evaluating various aspects of explainable AI, such as user satisfaction, trust, reliance, curiosity, and system performance. In [16] authors proposes a suite of multifaceted metrics to objectively compare different explainers based on correctness, consistency, and confidence. The paper shows that the proposed metrics are computationally inexpensive and can be used across different data modalities. Work [1] introduces the concept of robustness for interpretability, arguing that it is a crucial feature. The authors show that current methods lack robustness and propose methods to enforce robustness in existing interpretability approaches. Finally, the aim of the paper [6] is to help researchers to map existing tools and apply evaluation metrics when developing an XAI system.f The work summarizes the state-of-the-art review in XAI evaluation metrics and highlights challenges and future developments.

In [19], the authors present the first extensive literature review on XAI for TS classification. They propose a taxonomy for explanation methods and highlight open research directions. In "A Model-Agnostic Approach to Quantifying the Informativeness of Explanation Methods for Time Series Classification" [13], the authors propose a model-agnostic approach for quantifying and comparing different saliency-based explanations for TS classification. The authors list a number of explainers for classifiers based on deep neural networks. They also use perturbations as an evaluation measure. With perturbation-based analysis they show that the discrimination of TS parts plays a critical role in classification accuracy. They distinguish 2 approaches to perturbation: applied only to discriminative region (*Type 1*) and applied only to non-discriminative region (*Type 2*).

2.1 Evaluation metrics for eXplainable AI - Challenges and Prospects

There exists a multitude of techniques in the field of *XAI* designed to interpret and understand the decision-making processes of AI models. Some prominent *XAI* techniques include *Local Interpretable Model-agnostic Explanations* (*LIME*), *SHapley Additive exPlanations* (*SHAP*), and *Counterfactual Explanations* [4]. These approaches aim to provide human-understandable explanations to clarify AI system behavior.

Evaluation metrics in *XAI* are categorized into subjective, based on human preferences, and objective, achieved through formal definitions [6], which is depicted in Figure 1. XAI methods come with metrics specific to the method or focused on behavior of model being explained and its other aspects like cognitive complexity or computational cost affecting algorithm execution for large datasets. For evaluating explanation effectiveness in task-specific methods, metrics like non-representativeness or diversity are critical [18]. Counterfactual explanations demand metrics for diversity of changes and feasibility [12]. Recommendation systems require tailored, task-specific metrics [17], while model-related metrics usually focus on feature importance

DOI: 10.1007/978-3-031-63772-8_27

e.g., sensitivity, monotony [18]. Some methods use oversimplified *post-models*, evaluated based on size, complexity, or accuracy [5]. Data modality (e.g. images, text, tabular data), type and structure of explanations influence selection of metrics [5,13,16]. The evaluation of explanations consistency and stability is vital for model trustworthiness [3], with particular challenges and requirements highlighted in space exploration contexts [2]. Comprehensive reviews indicate a need for continuous research and diverse applications in XAI [10,20].

The *Intelligible eXplainable AI* (InXAI) framework provides tools for computing metrics such as *Perturbational Accuracy Loss*, *Stability*, and *Consistency* for explanation models like SHAP and LIME [4]. This framework is significant as the code within the InXAI package is actively developed with the goal of merging it into the master branch.

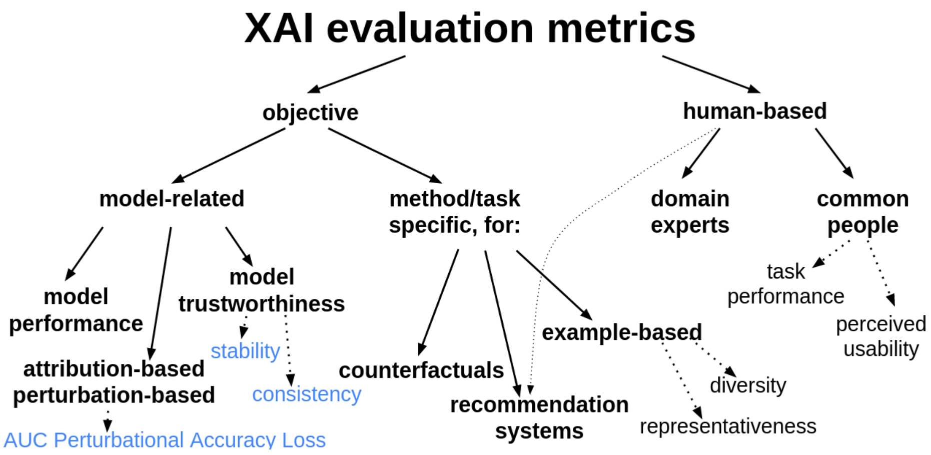


Fig. 1: Explanatory evaluation metrics are a relatively young field. There are 2 groups of explanatory evaluations - those based on objective metrics, calculated numerically, and human-centric, i.e. reflecting the final utility for the user. Sometimes the division between these 2 groups is blurred, such as in the case of XAI evaluation metrics for recommendation algorithms. The metrics highlighted in blue are part of the INXAI package.

2.2 InceptionTime Deep Network for Time Series classification

InceptionTime is a highly successive model [9] for classification of TS. It is an ensemble of deep Convolutional Neural Network (CNN) models inspired by the Inception-v4 architecture. InceptionTime shows high accuracy and scalability in TS classification tasks. The network has its implementation in *PyTorch* [5].

Considering the gaps identified in the state-of-the-art analysis of explanations for *Time Series* data, we propose a methodology with two key aspects: visualization and the adaptation

[4] Github: https://github.com/sbobek/inxai

[5] InceptionTime (in Pytorch) - GitHub: https://github.com/TheMrGhostman/InceptionTime-Pytorch

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63772-8_27

of the *AUC Perturbational Accuracy Loss metric*, aiming to address these shortcomings effectively.

3 Methodology

We present proposal of a methodology for evaluating eXplainable Artificial Intelligence (XAI) algorithms for anomaly detection machine-learning models in Time Series (TS). Our focus is on local explanations in the form of feature importance, as they are the most general, popular, and versatile approaches in the domain of XAI. Visuals include among others *AUC-PALM*. This metric assesses fidelity of the explainer (its consistency with the explained model) via perturbing TS data proportionally or inversely to the importance of the given explainer. Formula for perturbing TS takes into account both the feature importances at the observation level for the given the explainer and the class prototype. Class prototypes were obtained with Dynamic Time Warping (DTW) Barycenter Averaging. We demonstrate that, in the case of TS, it matters whether the damages are consistent with a given TS class or the opposite, which offers another indication of fidelity.

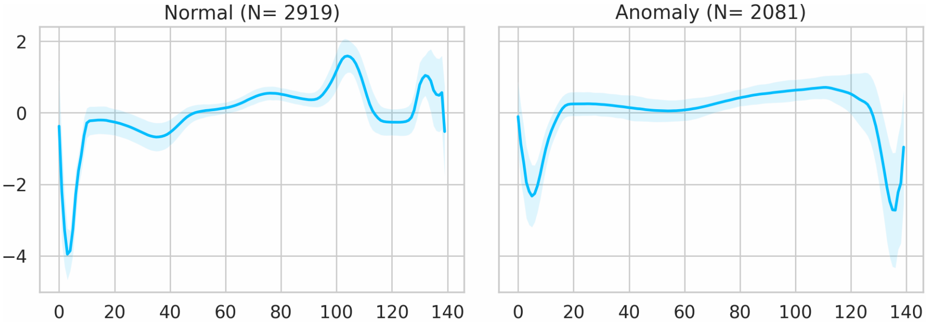


Fig. 2: The plot presents mean for each class with standard deviation plotted around it. The classes associated with pathological patterns (*R-on-T PVC*, *PVC*, *SP or EB* and *UB*) were recoded as *Anomaly* class.

3.1 Dataset

The dataset used in the experiments is derived from [7]. It was prepared by Y. Chen, E. Keogh [6]. The preparation process included two steps: extraction of each heartbeat, and making each heartbeat equal length using interpolation. The *"ECG5000"* dataset consists of *5,000* univariate TS, divided into *4,500* for training and *500* for testing. These TS, acquired through ECG, contain *140* timesteps each. The dataset comprises *5* different classes (with *2* major ones), each TS has *1* dimension ECG.

Each sequence corresponds to a single heartbeat from a single patient with congestive heart failure. The dataset includes five classes of heartbeats: Normal (*N*), R-on-T Premature

[6] See: http://timeseriesclassification.com/description.php?Dataset=ECG5000

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63772-8_27

Ventricular Contraction (*R-on-T PVC*), Premature Ventricular Contraction (*PVC*), Supraventricular Premature or Ectopic Beat (*SP or EB*), and Unclassified Beat (*UB*). Since we utilize a binary classifier in the article, we applied the re-encoding of all ECG classes associated with pathological patterns into a single *Anomaly* class. This is illustrated in 2.

Table 1 Confusion Matrix for the classifier on the test set across all classes.

Predicted	**True class**	**#**
Normal	Normal	306
Anomaly	R on T	164
	SP	14
	PVC	8
Normal	SP	4
	PVC	3
	UB	2
Anomaly	UB	1

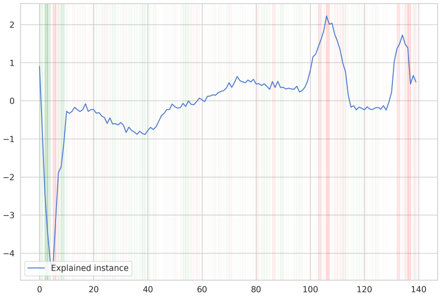


Fig. 3: Example LIME importances for single observation as plotted by *LimeTimeSeriesExplainer package*. One can observe usage of number of features equal to the length of the Time Series (140).

3.2 RNN-autoencoder extension

We developed a deep learning model using the Transfer Learning technique. We utilized a pre-trained RNN-autoencoder model implemented in *PyTorch*. The model was available online and trained on the data discussed in Section 3.1, specifically on the *Normal* class from the test subset. This allowed the author of the original code, Valkov V. [7], in *Collab notebook* [8] to determine a cut-off threshold for the cumulative reconstruction loss, beyond which there was a high probability of belonging to the *Anomaly* class.

In contrast, we extended the autoencoder with a classification head, consisting of Linear, Dropout, and Sigmoid layers. The input for the classification head was the difference between the input and the reconstruction, i.e., the reconstruction loss vector with the length of the TS (140 samples in each series). Incorporating the reconstruction loss vector allowed us to later use this vector as one of the explainers and compare it with SHAP and LIME. The model was fine-tuned with frozen autoencoder weights, on the test subset, this time for both classes: *Normal* and *Anomaly*. We achieved an accuracy of 0.98 on the test set. One can observe with *Confusion Matrix* in Table 1 that the classifier mistakes for 3 classes recoded to *Anomaly* (SP, PVC, UB) for just 9 observations, classifying them as *Normal*.

[7] See Github: https://github.com/curiousily/Getting-Things-Done-with-Pytorch

[8] Time Series Anomaly Detection using LSTM Autoencoders with PyTorch in Python: https://curiously.com/posts/time-series-anomaly-detection-using-lstm-autoencoder-with-pytorch-in-python

3.3 Visualisation of importances

SHAP was calculated using the DeepExplainer package [9]. For LIME, we employed the LimeTimeSeriesExplainer package [10], with the number of features (*num_features*) set equal to the length of the series, allowing us to obtain an explainer comparable to SHAP, which is depicted on Figure 3. We used the output from the RNN-autoencoder for obtaining reconstruction loss (referred to as *"LOSSES"*). We developed visualizations to display feature importances for each explainer, categorized by class. These visualizations show both the average TS and its standard deviation. A novel aspect of our method is an area around the average TS, indicating the importance of each TS segment. This area's upper and lower outlines represent average values of positive and negative importances, respectively.

$$\begin{aligned} \Phi^e_{\text{down}} &= \overline{\min(\Phi^e,0)} \\ \Phi^e_{\text{up}} &= \overline{\max(\Phi^e,0)} \\ \Phi^e_{\text{under-line}} &= \bar{TS} + \Phi^e_{\text{down}} \cdot const \\ \Phi^e_{\text{over-line}} &= \bar{TS} + \Phi^e_{\text{up}} \cdot const \end{aligned} \tag{1}$$

$\bar{TS}$ denotes the mean value of TS, while $\Phi^e_{\text{under-line}}$ and $\Phi^e_{\text{over-line}}$ are outlines of importances. The local explainer, Φ^e, matches the TS length. Φ^e_{down} and Φ^e_{up} represent the averages of negative and positive explainer values, respectively. We draw a range around $\bar{TS}$ on the plot, with lower and upper edges corresponding to Φ^e_{down} and Φ^e_{up}, respectively. A constant multiplier enhances visibility. Additionally, we calculated class prototypes using *tslearn*'s *barycenters.dtw_barycenter_averaging* function. These prototypes were visualized and used in our TS permutation method.

The resulting plot offers easy interpretation, particularly in binary classification. For example, with SHAP, the final part of the plot distinctly contributes to different classes. SHAP identifies cohesive areas, emphasizing their centers more than edges, while LIME finds almost the entire series area crucial. Comparing with reconstruction loss, both SHAP and LIME highlight series parts with significant reconstruction loss. These observations are shown in Figure 4.

3.4 AUC Perturbational Accuracy Loss Metric

The next step involved comparing explainers using the *Area Under Curve Perturbational Accuracy Loss Metric* (*AUC-PALM*). This metric serves as a direct measure of the decline in model performance, providing an intuitive and straightforward method to assess the impact of perturbations on the explanatory power of the models.

To address visualization, we developed plots where the noisy *AUC-PALM* is smoothed exponentially, still ensuring accurate calculation of the metric, as shown in Figure 5. The graphical representation of *AUC-PALM* showcases the accuracy loss on the Y-axis against the degree of perturbation on the X-axis. From this graph, the area under the curve is calculated, representing the *AUC-PALM*. This area is a crucial metric for evaluating the robustness of the explanation models in relation to the changes introduced in the input data. The calculation of *AUC-PALM* involves normalizing both the X and Y axes to 100%, making the maximum possible area under the curve equal to 1.0. Consequently, the area under the curve essentially represents the average accuracy loss across the perturbed instances.

[9] Documentation: https://shap-lrjball.readthedocs.io/en/latest/generated/shap.DeepExplainer.html

[10] GitHub: https://github.com/emanuel-metzenthin/Lime-For-Time

DOI: 10.1007/978-3-031-63772-8_27

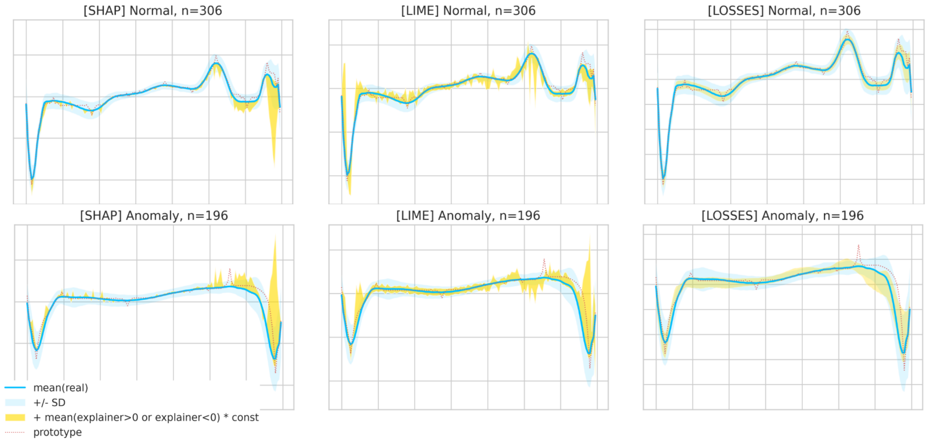


Fig. 4: Comparison of SHAP, LIME and reconstruction loss (denoted "LOSSSES") as explainers. One can observe, that they differ considerably. In the case of SHAP, the final part of the graph negatively contributes to the *Normal* class, and positively to the *Anomaly* class. The same is true for LIME, but in this case almost the entire area of the series is important for explanation. Both SHAP and LIME tend to indicate those areas of the series for which reconstruction loss is greatest. The red line furthermore indicates the prototype for the class. Examples are for test set.

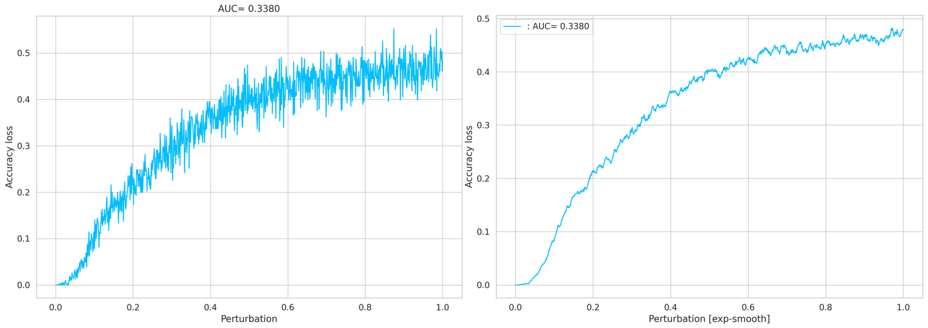


Fig. 5: Exponential smoothing of AUC Perturbational Accuracy Loss curves improves clarity, however does not change the metric value.

We faced challenges in perturbing Time Series (TS) and visualizing results. For perturbing TS, we considered perturbation around the original TS, around the prototype of the same class, and around the prototype of the opposite class. The perturbation level should increase smoothly to transition accuracy from unperturbed to random classification. It should also reflect the explainer's feature-importance for a given observation.

For the type of perturbation, we identified three approaches: inverse to feature-importance (*"inverted"*), consistent with feature-importance (*"straight"*), and consistent but not proportional beyond a certain threshold (*"zoned"*).

For TS perturbations, we tested a number of formulas. We will discuss *"inverted"* and *"zoned"* perturbations in detail, as those are the most complex cases. The process has 2 stages: 1. calculation of the perturbation vector, 2. application of the perturbation to the observation.

Stage 1 For feature importance visualization in TS, we normalize the absolute value of the explainer's output to create a unit length vector, $\boldsymbol{F}_{norm}$. In the *"inverted"* case, perturbations involve subtracting this vector from the unit vector (*"[V1]"*) or dividing the unit vector by $\boldsymbol{F}_{norm}$, with a small constant to avoid division by zero (*"[V3]"*). The *"zoned"* method zeros values below the *80th* percentile and replaces higher values with a constant, while the *"straight"*r method does not invert feature importances. All methods lead to a normalized perturbation vector, $\boldsymbol{P}_{norm}$. This is represented by the Equation 2.

$$\begin{gathered}\boldsymbol{F}_{norm} = \frac{\boldsymbol{F}}{\|\boldsymbol{F}\|} \\[1em] [V1]: \boldsymbol{P} = 1 - |\boldsymbol{F}_{norm}| \\ [V3]: \boldsymbol{P} = \frac{1}{|\boldsymbol{F}_{norm} + \boldsymbol{c}|} \\[1em] \boldsymbol{P}_{norm} = \frac{\boldsymbol{P}}{\|\boldsymbol{P}\|}\end{gathered} \tag{2}$$

$$\begin{gathered}[around\ prototype]: \boldsymbol{R} = \boldsymbol{Pr} \\ [around\ observation]: \boldsymbol{R} = \boldsymbol{O} \\[1em] \boldsymbol{P_{final}} = \boldsymbol{R} \circ \boldsymbol{P_{norm}} \\[1em] \boldsymbol{O_P} = \boldsymbol{O} + [i \cdot \text{rand}(\{1, -1\}) \text{ for } i \text{ in } \boldsymbol{P_{final}}] \cdot \alpha\end{gathered} \tag{3}$$

Stage 2 Perturbation application varies based on whether it's around a class prototype or relative to the observation itself. We define the *"reference"* as this chosen TS. $\boldsymbol{P_{final}}$ - the perturbation vector - is obtained by elementwise multiplication of the *importance vector* and *reference vector*. The final perturbed TS, $\boldsymbol{O_P}$, is the sum of the input observation and perturbation vector, each component multiplied by a random sign and a scalar for perturbation strength. See the Formula 3.

As being stated, the reference level can be either the prototype of the class to which the TS instance belongs: *[around prototype]*, or the observation itself: *[around observation]*. The prototype can be either from the same class or from the opposite class. The prototype is calculated using the Dynamic Time Warping (DTW) Barycenter Averaging method, although this is not shown in the formula. The final perturbation $\boldsymbol{P_{final}}$ is calculated as an element-wise product of this reference $\boldsymbol{R}$ and the $\boldsymbol{P_{norm}}$ from the previous formula. The perturbation is applied to the $\boldsymbol{O}$ to obtain the perturbed observation $\boldsymbol{O_P}$ in such a way that each component of the observation has added perturbation components with a random sign. Furthermore, it is multiplied by the α, which determines the strength of the perturbation and is changed from a small value to a large one as the *AUC-PALM* is drawn (calculated).

4 Results

4.1 Joint evaluation of *Normal* and *Anomaly* classes

We analyzed the *AUC-PALM* graphs and values for the test subset, including both *Normal* and *Anomaly* classes, to understand the impact of perturbation strategies on explainer performance. Figures 6 and 7 show that both *"inverted"* and *"straight"* perturbation methods affect SHAP, LIME, and reconstruction losses differently. The *"straight"* method results in larger *AUC-PALM* differences and quicker convergence to *0.5*, indicating a random classifier. The *"zoned"* case, performing similarly to *"straight"*, is not shown. SHAP outperformed in *AUC-PALM* results, followed by LIME, with reconstruction losses being slightly less effective but still a viable explainer.

4.2 Insights from individual class analysis

Since SHAP proved to be a more accurate explainer according to the *AUC-PALM* on the combined *Normal* and *Anomaly* classes, providing better feature importances, we investigated whether such a relationship also occurs when analyzing the *Normal* and *Anomaly* classes separately. The Table 2 provides a detailed overview of the results from various experiments performed on two classes of data: *Anomaly* and *Normal*. A range of tests were conducted for each data class, taking into account different types of perturbations. The outcomes are represented in relation to three distinct methods: LIME, reconstruction losses ("REC. LOSSES"), and SHAP.

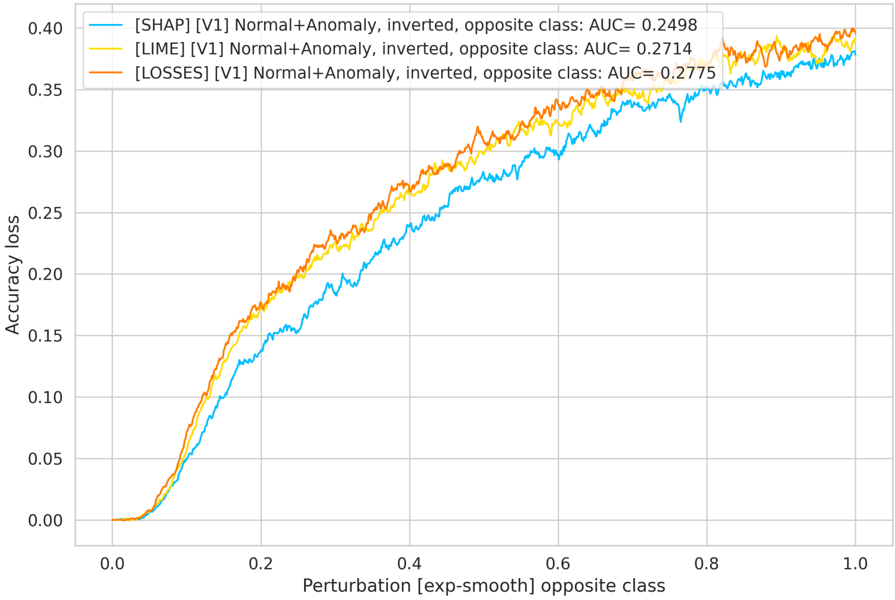


Fig. 6: An "inverted" method of perturbation for LIME, SHAP, and reconstruction losses-based explainer. The lower the area AUC, the better the explainer. SHAP performs the best. Results obtained on test set for Normal and Anomaly classes altogether. The line is smoothed exponentially.

We adopt the ratio of variance to mean as a measure of the differentiation strength between different explainers. Notably, the highest ratio of variance to mean is observed for *"straight"* condition, both for *Anomaly* and *Normal* class. For *Anomaly*, perturbation around

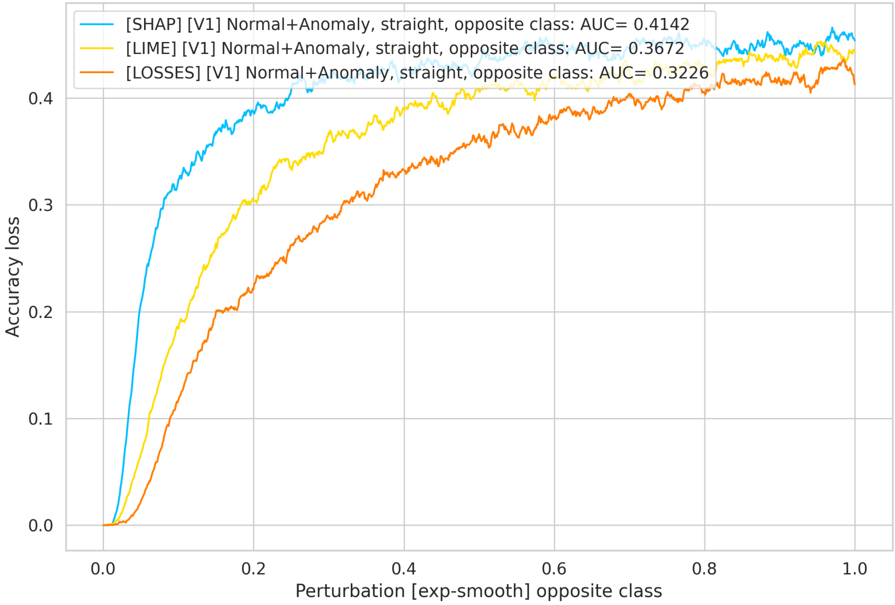


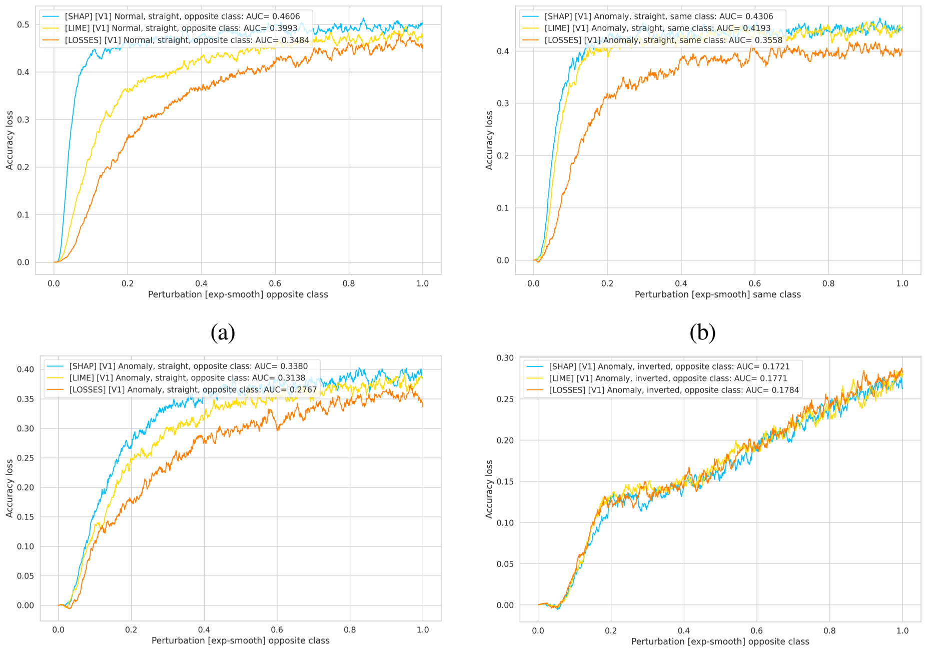
Fig. 7: A "straight" method of perturbation for LIME, SHAP, and reconstruction losses-based explainer. The higher the area AUC, the better the explainer. SHAP is the best once again. Results obtained on test set for Normal and Anomaly classes altogether. The line is smoothed exponentially.

Table 2 Results of experiments with various types of perturbations

Class	**Proportionality**	**Prototype**	***LIME***	***REC. LOSSES***	***SHAP***	**mean**	**var**	**var/mean**
Anomaly	**inverted**	**around obs.**	0.1853	0.2045	0.1697	0.1865	3.032×10^{-4}	1.626×10^{-3}
		opposite class	0.1771	0.1784	0.1721	0.1759	1.103×10^{-5}	6.273×10^{-5}
		same class	0.1769	0.1988	0.1584	0.1780	4.090×10^{-4}	2.298×10^{-3}
	straight	**around obs.**	0.4014	0.3745	0.4093	0.3950	3.336×10^{-4}	8.445×10^{-4}
		opposite class	0.3138	0.2767	0.3380	0.3095	9.525×10^{-4}	3.077×10^{-3}
		same class	0.4193	0.3558	0.4306	0.4019	1.626×10^{-3}	4.045×10^{-3}
	zoned	**around obs.**	0.3481	0.3563	0.3515	0.3520	1.717×10^{-5}	4.877×10^{-5}
		opposite class	0.2680	0.2691	0.2934	0.2768	2.068×10^{-4}	7.469×10^{-4}
		same class	0.3444	0.3537	0.3480	0.3487	2.201×10^{-5}	6.313×10^{-5}
Normal	**inverted**	**around obs.**	0.3269	0.3378	0.3332	0.3326	3.020×10^{-5}	9.081×10^{-5}
		opposite class	0.3294	0.3376	0.2957	0.3209	4.944×10^{-4}	1.541×10^{-3}
		same class	0.3354	0.3403	0.3388	0.3382	6.347×10^{-6}	1.877×10^{-5}
	straight	**around obs.**	0.4134	0.3566	0.3975	0.3892	8.5962×10^{-4}	2.209×10^{-3}
		opposite class	0.3993	0.3484	0.4606	0.4028	3.161×10^{-3}	7.847×10^{-3}
		same class	0.4027	0.3555	0.4029	0.3870	7.461×10^{-4}	1.928×10^{-3}
	zoned	**around obs.**	0.3944	0.3592	0.3733	0.3756	3.127×10^{-4}	8.326×10^{-4}
		opposite class	0.4054	0.3568	0.3956	0.3860	6.613×10^{-4}	1.713×10^{-3}
		same class	0.3903	0.3564	0.3755	0.3741	2.888×10^{-4}	7.720×10^{-4}

same class gave the highest result, while for *Normal* around opposite class. Nevertheless, perturbing *Anomaly* class around the *"opposite class"* is also a viable and almost equally potent alternative, only slightly lagging behind the leading solutions. It is apparent from the results that perturbation centered around the anomalous class yields the most significant effect. These findings underscore the utility of tailored perturbation strategies in maximizing the differentiation between various explainers. This indicates that applying perturbations in this particular way provides the most valuable insights into performance of explainers. Hence, it can be inferred which approach towards perturbations is the most effective.

The Figure 8 presents the top three results from the Table 2 summarizing the experiments. As can be observed, under the *"straight"* condition and *"opposite class"* for the *Normal* class, all three explainers are well separated. In the case of *Anomaly*, the var/mean metric indicated that the best perturbation is around the same class, although the chart shows that the *"opposite class"* provides more information. For completeness, a chart for one of the worst var/mean values is also presented. In this case, the chart does not provide significant insight. At the same time, it is noticeable that explanations for the *Normal* class are better than those for the *Anomaly* class, reflecting the fact that the autoencoder was trained only on the *Normal* class.



(a)

(b)

(c)

(d)

Fig. 8: SHAP, LIME and reconstruction losses-based explainer compared with best perturbation method according to *var/mean* metric (charts *(a)* - *(c)*). The higher the area under curve, the better the explainer. For both *Normal* and *Anomaly* class, the best is SHAP. The chart *(d)* represents the worst perturbation method, for reference. The line is smoothed exponentially.

It is important to note that directly comparing the *AUC-PALM* metric values between classes carries some risk. The perturbation coefficient α is uncalibrated, so if the model's accuracy does not decline to the same level at the end of the plot, comparing the area under the curve will not be reliable. Moreover, the presented graph is derived from a different computation process than the one collected in the Table 2. It can be observed that the calculated *AUC-PALM* values exhibit high stability and are consistent up to the third decimal place for SHAP and reconstruction losses (here referred to as autoencoder losses), and up to the second decimal place for LIME. The notebook accompanying this paper can be found on our Github [11].

5 Discussion

Our study underscores the role of visualization in elucidating feature importances, as detailed in 3.3. These visualizations provide immediate insights into the segments of time series critical for distinguishing between normal and anomalous classes. Often, explanations overlook the fact that different segments of the series may contribute to anomalies, while others may indicate normalcy. In our visualizations, we have also considered the character of the data, creating visualizations that are significantly more useful for medical experts than the default ones provided by libraries such as *SHAP* or *LIME*. Moreover, these visualizations are compatible with other local, feature-based explanations. This innovation is adaptable to any black-box, feature-based explanation method, enhancing its utility across various applications.

We have presented an approach to the evaluation of explanation quality in feature-importance attribution algorithms, such as *SHAP* and *LIME*, with a special emphasis on per-class analysis in *Time Series Anomaly Detection*. Our work is important for Time Series analysis, and demonstrated through ECG data analysis described in 3.1, showcasing the applicability of our methods also in healthcare scenarios.

Central to our analysis is the exploration of various perturbation scenarios to understand their impact on model performance in *Time Series* (TS) data. By employing *Dynamic Time Warping* (DTW) *Barycenter Averaging* for prototyping, we navigate through perturbations around class prototypes and the observation itself, enhancing our understanding of feature-importance-based explanations (see 3.4). Our methodology introduces novel perturbation methods, such as *"straight"* (proportional to feature importance) and *"zoned"* (above selected threshold), alongside *"inverted"* (inversely proportional) approaches, to discern the efficacy of explainers, with the *"straight"* method proving superior in explainer differentiation. This approach not only facilitates rapid identification of class-dependent areas in binary classifiers but also extends to multi-class scenarios by treating them as a stack of binary classifiers, thereby underscoring its universality across different classifier architectures, including non-deep learning models (see 3.2). Furthermore, we explore the use of *reconstruction loss* from autoencoder-based models as a baseline for comparison with feature importance-based explainers.

6 Conclusion

This study has explored the important role of visualizations in conveying to domain experts which parts of a time series are globally significant for predicting a given class, whether nor-

[11] Jupyter .ipynb notebook for paper: https://github.com/mozo64/tsxai/blob/main/06_time_series_anomaly_detection_ecg_clear.ipynb

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63772-8_27

mal or anomalous. These visual aids provide a rapid insight into the model's decision-making process, based on local explanations for feature-based models in both machine learning (ML) and deep learning (DL) anomaly detection contexts.

While literature describes various methods for perturbation, our approach to calculating *AUC Perturbational Accuracy Loss metric* (*AUC-PALM*) allows for a more explicit differentiation between different explainers, facilitating informed decisions on which is more effective in specific scenarios.

We systematically investigated perturbations: around the prototype of the given class, the prototype of the opposite class, and directly around the observation itself (without a prototype), presenting formulas for calculating *AUC-PALM* in each case. In our experiments using SHAP, LIME, and reconstruction loss as a baseline, the most significant distinction was observed with perturbations around the prototype of the opposite class (or, for simplicity, anomalous class would suit for most cases), proportional to feature importance—a condition we termed *"straight."* This highlighted SHAP's superior performance and pointed out the reconstruction loss's limitations. Our findings underscore the necessity for nuanced and accessible explanatory tools in the field of anomaly detection. By providing clear visualizations and a robust metric for explainer evaluation, we aim to bridge the gap between complex data patterns and actionable insights, enabling more effective decision-making in critical applications.

Future directions include refining our approach by training class-specific autoencoders and conducting experimental studies with participants to evaluate the effectiveness of our techniques in real-world machine-learning problems, i.e. predictive maintenance scenarios, paving the way for more interpretable and reliable anomaly detection in *Time Series* data.

Acknowledgment

This paper is funded from the XPM (Explainable Predictive Maintenance) project funded by the National Science Center, Poland under CHIST-ERA programme Grant Agreement No. 857925 (NCN UMO-2020/02/Y/ST6/00070). The research has been supported by a grant from the Priority Research Area (DigiWorld) under the Strategic Programme Excellence Initiative at Jagiellonian University. We acknowledge the use of OpenAI's ChatGPT-4 for reviewing and improving the language and style of this manuscript.

References

1. Alvarez-Melis, D., Jaakkola, T.S.: On the robustness of interpretability methods (2018)
2. Barkouki, T., Deng, Z., Karasinski, J., Kong, Z., Robinson, S.: Xai design goals and evaluation metrics for space exploration: A survey of human spaceflight domain experts (01 2023). https://doi.org/10.2514/6.2023-1828
3. Bobek, S., Bałaga, P., Nalepa, G.J.: Towards model-agnostic ensemble explanations. In: Paszynski, M., Kranzlmüller, D., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M. (eds.) Computational Science – ICCS 2021. pp. 39–51. Springer International Publishing, Cham (2021)
4. Bobek, S., Mozolewski, M., Nalepa, G.: Explanation-driven model stacking. In: Paszyński, M., Kranzlmüller, D., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M.A. (eds.) Computational Science – ICCS 2021 : 21st International Conference, Krakow, Poland, June 16-18, 2021 : proceedings, part VI, pp. 361–371. Lecture Notes in Computer Science, ISSN 0302-9743, eISSN 1611-3349; 12747, Springer International Publishing, Cham (2021)

5. Bobek, S., Nalepa, G.J.: Local universal rule-based explanations (2023)
6. Coroamă, L., Groza, A.: Evaluation Metrics in Explainable Artificial Intelligence (XAI), pp. 401–413 (11 2022). https://doi.org/10.1007/978-3-031-20319-0_30
7. Goldberger, A.L., Amaral, L.A., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E.: PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation **101**(23), E215–20 (Jun 2000)
8. Hoffman, R.R., Mueller, S.T., Klein, G., Litman, J.: Metrics for explainable ai: Challenges and prospects (2019)
9. Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D., Weber, J., Webb, G., Idoumghar, L., Muller, P.A., Petitjean, F.: Inceptiontime: Finding alexnet for time series classification. Data Mining and Knowledge Discovery **34**, 1–27 (11 2020). https://doi.org/10.1007/s10618-020-00710-y
10. Kadir, M.A., et al.: Evaluation metrics for xai: A review, taxonomy, and practical applications. ResearchGate (2023), https://www.researchgate.net/publication/366917148_Evaluation_Metrics_for_XAI_A_Review_Taxonomy_and_Practical_Applications
11. Li, M., Jiang, Y., Zhang, Y., Zhu, H.: Medical image analysis using deep learning algorithms. Frontiers in Public Health **11**, 1273253 (2023). https://doi.org/10.3389/fpubh.2023.1273253, https://www.frontiersin.org/articles/10.3389/fpubh.2023.1273253/full
12. Nauta, M., Trienes, J., Pathak, S., Nguyen, E., Peters, M., Schmitt, Y., Schlötterer, J., van Keulen, M., Seifert, C.: From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI. ACM Computing Surveys **55**(13s), 1–42 (jul 2023). https://doi.org/10.1145/3583558, https://doi.org/10.1145%2F3583558
13. Nguyen, T.T., Le Nguyen, T., Ifrim, G.: A model-agnostic approach to quantifying the informativeness of explanation methods for time series classification. In: Lemaire, V., Malinowski, S., Bagnall, A., Guyet, T., Tavenard, R., Ifrim, G. (eds.) Advanced Analytics and Learning on Temporal Data. pp. 77–94. Springer International Publishing, Cham (2020)
14. Parmar, C., Barry, J.D., Hosny, A., Quackenbush, J., Aerts, H.J.: Data analysis strategies in medical imaging. Clinical Cancer Research **24**(15), 3492–3499 (07 2018). https://doi.org/10.1158/1078-0432.CCR-18-0385, https://doi.org/10.1158/1078-0432.CCR-18-0385
15. S Band, S., Yarahmadi, A., Hsu, C.C., Biyari, M., Sookhak, M., Ameri, R., Dehzangi, I., Chronopoulos, A.T., Liang, H.W.: Application of explainable artificial intelligence in medical health: A systematic review of interpretability methods. Informatics in Medicine Unlocked **40**, 101286 (2023). https://doi.org/https://doi.org/10.1016/j.imu.2023.101286, https://www.sciencedirect.com/science/article/pii/S2352914823001302
16. Santhanam, G.K., Alami-Idrissi, A., Mota, N., Schumann, A., Giurgiu, I.: On evaluating explainability algorithms (2020), https://openreview.net/forum?id=B1xBAA4FwH
17. Sisk, M., Majlis, M., Page, C., Yazdinejad, A.: Analyzing xai metrics: Summary of the literature review (10 2022). https://doi.org/10.36227/techrxiv.21262041
18. Sun, J., Shi, W., Giuste, F.O., Vaghani, Y.S., Tang, L., Wang, M.D.: Improving explainable ai with patch perturbation-based evaluation pipeline: a covid-19 x-ray image analysis case study. Scientific Reports **13**(1), 19488 (2023). https://doi.org/10.1038/s41598-023-46493-2, https://doi.org/10.1038/s41598-023-46493-2
19. Theissler, A., Spinnato, F., Schlegel, U., Guidotti, R.: Explainable ai for time series classification: A review, taxonomy and research directions. IEEE Access **10**, 100700–100724 (2022). https://doi.org/10.1109/ACCESS.2022.3207765
20. Vilone, G., Longo, L.: Notions of explainability and evaluation approaches for explainable artificial intelligence. Information Fusion **76**, 89–106 (2021). https://doi.org/https://doi.org/10.1016/j.inffus.2021.05.009, https://www.sciencedirect.com/science/article/pii/S1566253521001093
21. Zhou, J., Gandomi, A., Chen, F., Holzinger, A.: Evaluating the quality of machine learning explanations: A survey on methods and metrics. Electronics **10**, 593 (03 2021). https://doi.org/10.3390/electronics10050593