InteDisUX: Intepretation-Guided Discriminative User-Centric Explanation for Time Series

Viet-Hung Tran[1*], Zichi Zhang[1*], Tuan Dung Pham[1*], Ngoc Phu Doan[1*], Anh-Tuan Hoang[1], Peixin Li[1], Hans Vandierendonck[1], Ira Assent[2], Son T. Mai[1]

[1]Queen's University Belfast, UK
[2]Aarhus University, Denmark
{h.tran, h.vandierendonck, thaison.mai}@qub.ac.uk, ira@cs.au.dk

Abstract

Explanation for deep learning models on time series classification (TSC) tasks is an important and challenging problem. Most existing approaches use attribution maps to explain outcomes. However, they have limitations in generating explanations that are well-aligned with humans's perceptions. Recently, LIME-based approaches provide a more meaningful explanation via segmenting the data. However, these approaches are still suffering from shortcomings in the processes of segment generation and evaluation. In this paper, we propose a novel time series explanation approach called InteDisUX to overcome these problems. Our technique utilizes the segment-level integrated gradient (SIG) for calculating importance scores for an initial set of small and equal segments before iteratively merge two consecutive ones to create better explanations under a *unique* greedy strategy guided by two new proposed metrics including discrimination and faithfulness gains. By this way, our method does not depend on predefined segments like others while being robusts to instability, poor local fidelity and data imbalance like LIME-based methods. Furthermore, *InteDisUX is the first work to use the model's information to improve the set of segments* for time series explanation. Extensive experiments show that our method outperforms LIME-based ones in 12 datasets in terms of faithfulness and 8/12 datasets in terms of robustness.

1 Introduction

Time series classification (TSC) plays a vital role in many practical domains, such as finance (Liang, Wang, and Wu 2023), healthcare (Lin et al. 2019), and supply chain management (Keller, Thiesse, and Fleisch 2014). However, it remains one of the most challenging problems despite many research efforts (Mohammadi Foumani et al. 2024). Recently, Deep Learning (DL) has been emerged as an effective approach for TSC compared to conventional statistical methods due to its ability to learn relevant latent feature representations from long-term data. Unfortunately, DL models are deemed to be black boxes as their decision-making mechanisms are difficult to understand. This consequently hinders trustworthiness, makes it challenging to identify and correct errors, and limits the ability to extract meaningful insights from the model outputs. To overcome this non-transparent issue in TSC, Explainable AI (XAI) has emerged as an effective yet promising realm (Rojat et al. 2021).

Most existing methods in XAI aim to explain a time series model using attribute maps, which contain a score to represent the importance of each time point, e.g., (Liu et al. 2024a,b; Enguehard 2023; Queen et al. 2024; Doddaiah et al. 2022). However, these attribute map explanations are not similar to people's perceptions in terms of psychology and philosophy since humans have a preference for explanations that are both simple and informative (Singh, Murdoch, and Yu 2018; Read and Marcus-Newhall 1993). Hence, a good explanation should be able to effectively capture the relationships between features while still being simple enough for users to search manually. Nevertheless, attribution maps for a time series are usually packed with huge amounts of time points, making them impractical for users to use. Furthermore, temporal correlations among features, which are a key difference between time series and tabular data, are not paid enough attention. Different to the attribute maps, LIME-based methods suggest the use of time series segmentation to divide time series into segments before employing different LIME variants to assign an importance score to each segment, e.g., (Neves et al. 2021; Guillemé et al. 2019; Sivill and Flach 2022). This segmentation scheme helps to reduce the amount of information provided to users while retaining temporal information inside segments. However, these approaches requires a predetermined number of segments to divide all time series into pieces before assessing their importance scores. This limits their ability to capture an appropriate number of segments and their locations (due to fixed precut segments/pieces) for each instance, hence reducing the effectiveness of the explanation. Additionally, current LIME approaches are also suffering from instability and poor local fidelity as pointed out in (Tan, Tian, and Li 2024). Moreover, their sampling sensitivity when dealing with imbalanced data is another limitation which we will further discuss in Section 2.4 of this paper.

Contributions. In this paper, we propose a novel time series explanation method, called Intepretation-Guided Discriminative User-Centric Explanation for Time Series (Inte-

*These authors contributed equally.

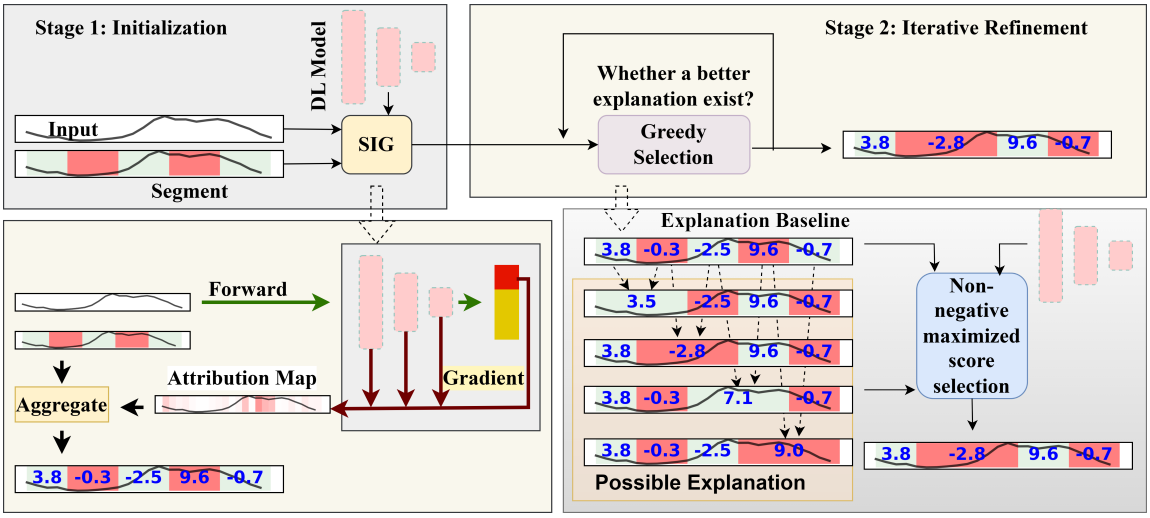

Figure 1: The overall process of InteDisUX. It has two different stages including initialization and iterative refinement. Details of two stages are described in the method Section (InteDisUX Method).

DisUX). InteDisUX also follows the segment-based explanation scheme due to the above mentioned benefits compared to the attribute map approach. However, our method can produce more faithful and robust segment-level explanations for time series models compared to other existing works, e.g. (Neves et al. 2021; Guillemé et al. 2019; Sivill and Flach 2022) due to some *unique* approaches described briefly in the following.

First, motivated by limitations of LIME, we propose a Segment-level Integrated Gradient (SIG), which is built upon the concept of Integrated Gradient (IG) (Sundararajan, Taly, and Yan 2017) for segments in time series. Using SIG method helps to avoid limitations that exist in LIME and to improve time series explanations significantly. We also provide theoretical analysis to show that SIG satisfies some important axioms, which help our algorithm avoid instability and reduce complexity (c.f. Section 2.6).

Second, we propose a novel bottom-up segmentation strategy for generating segments for the purpose of explanation. Our method fundamentally differs from previous researches in the way that it does not rely on precut segments (c.f. Section 2.3 for more detailed analysis). Instead, it captures the number of segments during a cycle using a greedy selection strategy and segment merging process. By this way, it can capture most important segments at more appropriate locations and lengths than other existing works as we shall demonstrated in Section 3.1.

Third, we introduce two new gain metrics for *iteratively* assessing suitable segments for better explanation during the greedy selection scheme mentioned above. The first metric is the *discrimination gain*, which focuses on the increment of distinctness between consecutive segments' patterns. This helps algorithm split segments such that time steps inside a segment have same behaviour and have different behaviour with surrounding segments. The second proposed metric is the *faithfulness gain*, which evaluates the increment of faithfulness if we merge nearly segments for an explanation.

Summarization. *To the best of our knowledge, our work is the first to use the learning model's information to improve the set of segments for time series explanation.* We conduct *extensive experiments* on 12 real-world datasets and 5 different deep learning architectures (including CNN, Transformer, FC, Dense-FCN, and LSTM-FCN) to evaluate the performance of our approach compare to the three state-of-the-art (SOTA) LIME-based methods (including (Neves et al. 2021; Guillemé et al. 2019; Sivill and Flach 2022)). The results show that our approach significantly outperforms existing work in terms of explanation ability.

2 Our Proposed Method InteDisUX

2.1 Preliminary

Problem formulation. We consider a TSC setting where the model F takes an input time series data point x from a space $X \subset \mathbb{R}^{d_X \times T}$, and a corresponding ground truth output y from a space $Y \subset \mathbb{R}^{d_Y}$. Every instance x contain d_X time series x_i, and each x_i is a sequence of data points $(x_{i,t})_{i\in[1:d_X],t\in[1:T]}$ indexed by dimension i, and time step t, where T is the total step. The dimensionality of each data point in the sequence is denoted by d_X. This allows for handling both univariate (single data stream, $d_X = 1$) and multivariate (multiple data streams, $d_X > 1$) time series data. For classification tasks, it is a one-hot vector $(y_i)_{i\in[1:d_Y]}$. The model F is trained by minimizing a loss function $L(F(x), y)$.

We define explanation for a time series input x as S and I, where $S = \{s_k | 1 \leq k \leq n\}$ is a set of n non-overlapped segment in which $\cup s_k = x$, and $I = \{i_k | 1 \leq k \leq n\}$ in which i_k is the correspondent important score of s_k. The higher value of $|i_k|$ means higher importance of s_k for the time series model. Our task is to determine S and I to maximize the explanability of the prediction outcome of $F(x)$.

Integrated Gradients (IG). IG is a gradient-based explanation method that assigns importance scores to each fea-

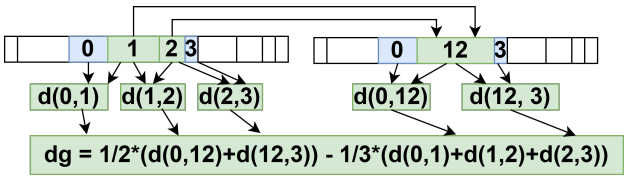

Figure 2: Discrimination gain calculation process. The discrimination gain (dg) is calculated by estimating the increase from average distance between consecutive segments. The distance function is d.

ture in an input data point. IG is proposed as an attribution method that aims to satisfy two fundamental axioms: Sensitivity and Implementation Invariance (Sundararajan, Taly, and Yan 2017). The following equation calculates the IG score for the j^{th} dimension of the input z:

$$IG_j(z) = (z_j - z'_j) \int_{\alpha=0}^{1} \frac{\partial F(z' + \alpha \times (z - z'))}{\partial z_j} d\alpha \quad (1)$$

where F is our model, z' is the baseline sample; z_j, z'_j is j^{th} dimension of z and z', respectively.

2.2 The Proposed InteDisUX Method

Figure 1 illustrates a overview our proposed method. It takes a time series x, an initial arbitrary number of segments N_s, and a deep learning model F as inputs to produce the explanation sets S and I. The algorithm consists of two primary stages: *initialization* stages and *iterative refinement* stages. In the first stage, the initial set of equal segments is evaluated by assigning a score to each segment using the SIG technique described in Section 2.5. The second stage involves an iterative refinement process (i.e. a circular process) in which InteDisUX iteratively computes two gain scores including discrimination gain and faithfulness gain (described in Section 1 and below) and chooses a more optimal set of segments for explanation until no better candidate found.

Stage 1: Initialization. The time series x is divided into a set S of N_s segments of equal length ($|x|/N_s$). For each segment in S, the algorithm calculates its segment-level Integrated Gradient (SIG) score. These SIG scores represent the influence of each segment on the model's prediction. Essentially, they act as an *importance score* for the current segmentation. By this way, we create initial units that are both small enough for analysis and contain meaningful data. These segments will be further refined in subsequent stages.

Stage 2: Iterative Refinement. To choose a better set of segments, we propose a *unique* greedy strategy to merge nearby segments using two metrics.

The first metric is *Discrimination Gain*, which refers to the increase in the average distance between consecutive segments surrounding the merged segments as shown in Figure 2 in which Hausdorf distance (Huttenlocher, Klanderman, and Rucklidge 1993) is employed to calculate distance between two segments. The other metric is *Faithfulness Gain*, which is the improvement of the faithfulness score

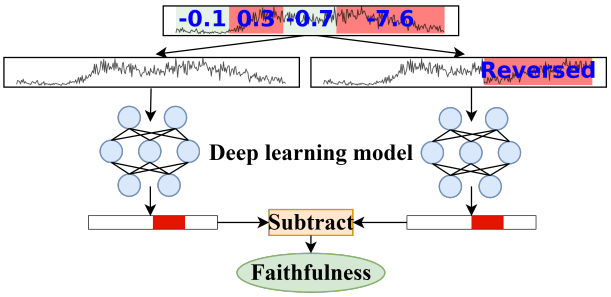

Figure 3: Faithfulness score calculation process. To estimate the faithfulness score, we put input and a partial reversed version into the model to obtain outputs and compute the difference between outputs corresponding to the true label.

from the current explanation set S. We obtain the *Faithfulness* score in two steps. In the first step, we put the time series x and its reversed instance into the model F to get outputs. We get partial reversed instance by reversing the most important segment defined by our explanation in time series data point. After that, the *Faithfulness* score is obtained by computing the reduction of the ground truth label's confidence score from the original time series data point to the partial reversed version. In addition, this metric is also one of the metric to evaluate the time series explanation in Section 3 (c.f. Figure 3 and supplementary for details).

The greedy strategy first iteratively generates $|S| - 1$ segment sets by merging two consecutive segments i and $i + 1$ in the explanation baseline, while the others stay unchanged. Then, we compute the discrimination gain and faithfulness gain. In the selection step, we select the segment set with the highest discrimination gain from the segment sets generated before. However, we ensure that the discrimination gain and faithfulness gain are both non-negative.

What make InteDisUX special? It generates an interpretable representation using a bottom-up approach to segment time series data without predetermined segments like others. The motivation arises from the need for a varying number of segments in time series samples as it is fully data dependent and is hard to set. We study the number of segments to support this observation in Section 2.3.

Secondly, our algorithm differs from LIMESegment (Sivill and Flach 2022) and other LIME-based methods in the way that it utilizes a discrimination score to determine the distinctness between consecutive segments rather than focusing on producing homogeneous regions while making decisions about time series segmentation. Moreover, our method is the first to using the model's information through the faithfulness score to make a decision about interpretable representation. In Section 3.3, we show that the two proposed scores contribute significantly to the performance of our method.

Instability in LIME arises when the sample weights are significantly reduced for samples that are far from the sample we are trying to explain. This results in an imbalanced distribution of information for the surrogate model to learn from. As a result, importance scores often approach zero,

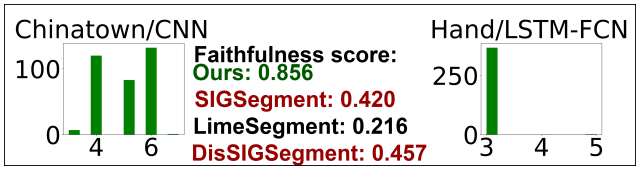

Figure 4: Distribution of the number of segments in final sets generated by our method.

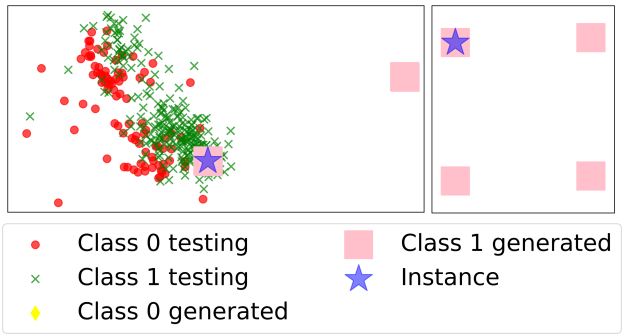

Figure 5: (Left) Visualization of the Chinatown testing dataset, an instance, and its samples in LIME around the instance. (Right) Visualization of an instance and its samples.

and the algorithm converges slowly. In addition, LIME also suffers from poor local fidelity. This means that the samples generated for LIME are far from the instance, rendering them non-local. Lastly, the imbalanced data sampling issue is that all samples generated around an instance fall into a single class, which leads to importance scores becoming zero for all segments when we conduct ridge regression. Our analysis of LIMESegment and other LIME-based methods in Section 2.4 shows that they suffer from these three issues, except that LIMESegment can avoid instability with a new weighting approach. By following different approach, InteDisUX can avoid all these problems.

2.3 Analysis on Number of Segments

To justify the hypothesis that the optimal number of segments is different between samples, Figure 4 shows the histogram of the final number of segments in InteDisUX, which indicates that the number of segments varies across instances. Please note that in LIMESegment, the number of segments is always 3, which is equivalent to the 2 change points in the LIMESegment paper (Sivill and Flach 2022)). Our final explanation has a significantly higher faithfulness score than LIMESegment, which further supports our hypothesis. Furthermore, we also conduct a comparison between SIGSEGment and DisSiGSegment. Both of them utilize the same scoring method, known as our SIG, but employ distinct segmentation approaches. The two approaches mentioned are specified in an ablation study. The results in Figure 4 show that DisSiGSegment is more faithful than SIGSegment, which uses LIMESegment's segmentation strategy because it uses different numbers of segments for each instance. Moreover, the histogram of the Hand dataset, which is in the figure on the right in Figure 4 shows that all instances still have the same number of segments. Please note that the Chinatown dataset is obtained from an automated pedestrian counting system in different time of the day. The number of segments for this dataset depends on various events in various days. However, the hand dataset only contains hand outlines, which are unaffected by various events. It leads to the same number of segments for almost all instances in the dataset.

2.4 Limitations of LIME in Time Series

We analyze the limitation of LIME in time series to justify why we propose SIG instead of using LIME. Concerning instability, the LIME-based explanations in (Neves et al. 2021; Guillemé et al. 2019) use the same weighting mechanism as conventional LIME and still face the instability indicated in (Tan, Tian, and Li 2024). If LIMESegment uses a different weighting mechanism, which uses dynamic time warping (Senin 2008) distance, then it does not suffer from instability. However, LIMESegment still has poor local fidelity as found in (Tan, Tian, and Li 2024).

Figure 5 (left) visualizes the Chinatown testing data, an instance, and its samples in LIME around the instance using PCA (Abdi and Williams 2010) for visualization. It shows that the instance's samples may be significantly distant from it, making them non-local. In addition, it also illustrates the imbalanced data sampling problem in LIME. This is evident in the figure where there are no yellow marks, which correspond to samples in class 0. In this situation, it indicates that all the local samples generated for the LIME mechanism belong to class 1, which is the class of the instance. The cause is that the dataset consists of only two classes, making it likely to have samples mostly belonging to the majority class. Imbalanced data sampling leads to the issue that all segments have a zero-importance score. The visualization for that situation in the Chinatown dataset is in the supplementary material. In addition, the right figure shows that there are many duplicate samples in the total 100 samples generated in LIMESegment for the Chinatown dataset. This is due to the fact that if LIMESegment only has 2 change points or 3 segments, then the number of binary representations, which is b^d with d being the number of segments, is 8. In (Neves et al. 2021; Guillemé et al. 2019; Sivill and Flach 2022), pertubations are used to replace segments with 0 in the binary representation. In (Neves et al. 2021), they use the mean valued segments, while the work in (Guillemé et al. 2019) uses randomly selected authentic segments from the original dataset, so the diversity of the instance's samples is still good. However, there are only 8 distinct time series samples because the perturbations in LIMESegment are instance-based and the same for all the isntance's samples. This leads to a lack of diversity in the set of samples, making the ridge regression of the LIME mechanism of LIMESegment learn with limited data.

2.5 Segment-level Integrated Gradient

Since segment of time series can be defined independently for each dimension, we define time series x_i as x for simplicity, then x is a sequence of $(x_t)_{t\in[1:T]}$ indexed by time

step t. We define a segment of a time series x as $s = (x_t)_{t\in[t_{low}:t_{high}]}$, where t_{low} and t_{high} are time steps within the sequence $[1:T]$, and $t_{low} \leq t_{high}$. To compute an importance score for a segment, we define segment-level integrated gradients (SIG) based on Integrated gradients (IG) (Sundararajan, Taly, and Yan 2017). IG values calculated for each data point x_t within the segment s are summed up to the the importance score for segment s. The following equation defines the calculation of SIG for segment s:

$$SIG(s) = \sum_{t\in[t_{low}:t_{high}]} IG_t(x) \tag{2}$$

For theoretically analysis, we generalize SIG Group Integrated Gradient (GIG). Given a data point with N features, denoted as $(x_j)_{j\in[1:N]}$, a group of features, named g, is simply a subset of these features, written as $g = \{x_k | k \in [1:N]\}$. The GIG score for this group, g, is defined as the sum of the individual IG scores for each feature within the group:

$$GIG(g) = \sum_{x_j\in g} IG_j(x), \tag{3}$$

In the special case, which is the empty group, we define:

$$GIG(\emptyset) = 0 \tag{4}$$

2.6 Theoretical Analysis for GIG

Based on the principles of cooperative game theory, we establish the following properties:

Axiom: Implementation Invariance. Two neural networks, f and f', are functionally equivalent if $f(x) = f'(x)$ for all inputs x. Assuming GIG_f is GIG for model f, GIG satisfies *implementation invariance* if $GIG_f(x) = GIG_{f'}(x)$ for all inputs x. Since GIG comes from IG and simply aggregates values from IG, it follows that if IG satisfies implementation invariance, then GIG also satisfies this axiom. In (Sundararajan, Taly, and Yan 2017) and Equation 1, it is proven that IG satisfies *implementation invariance*.

Axiom: Completeness. GIG satisfies the *completeness* axiom if all attribution scores of separate groups that cover input add up to the difference between the output of F at x and baseline input x'. The baseline input x' is an uninformative input, considered baseline in the IG method, as indicated in the paper (Sundararajan, Taly, and Yan 2017) and Equation 1. It can be formulated below.

Proposition 1. *Let an input x, the baseline x', $G = \{g_k|$ for each $(k_1, k_2), g_{k_1} \cap g_{k_2} = \emptyset; \bigcup_{g\in G} = x\}$ a set of separate groups that cover the whole input. If $F : \mathbb{R}^n \to [0.1]$ is differentiable almost everywhere, then $\sum_{g\in G} GIG(g) = F(x) - F(x')$.*

Axiom: Normality. It holds per definition in Equation 4.

Axiom: Superadditivity. GIG satisfies *superadditivity* if the GIG of the union of two disjoint sets of features is higher or equal to the sum of individual sets. In fact, if $G_1 \cap G_2 = \emptyset$, then $GIG(G_1 \cup G2) = GIG(G_1) + GIG(G_2)$.

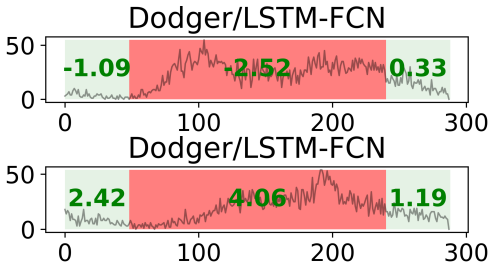

Figure 6: Visualisation of an example that explains the time series data of the Dodger dataset: (top) the normal day and (bottom) the game day.

Axiom: Sensitivity (a). It implies that a group that has a non-zero difference in outputs of F at x and x' should have a non-zero importance score. This holds for GIG:

Proposition 2. *Let a feature group g such that $F(x) \neq F(x')$ for every input vector x and baseline vector x' that x and x' can only differ in feature x_i if $x_i \in g$. Then, $GIG(g) \neq 0$.*

Axiom: Sensitivity (b). It implies that a group that has a zero difference in outputs of F at x and x' should have a zero importance score. This holds for GIG:

Proposition 3. *Let a feature group g, such that $F(x) = F(x')$ for every input vector x and baseline vector x' that x and x' can only differ in feature x_i if $x_i \in g$. Then, $GIG(g) = 0$.*

Axiom: Symmetry Preservation. It implies that if two features hold equivalent roles for a model to make a decision, then their impact in a group g should be equal. This holds true for GIG:

Proposition 4. *Two features x_i and x_j are functionally equivalent if $F(x) = F(z)$ for every input pair x and z such that $x_i = z_j$, $x_j = z_i$, and $x_k = z_k$ for every $k \notin \{i, j\}$. If x_i and x_j are functionally equivalent, for every g which contains both x_i and x_j or does not contain any of x_i or x_j, then $GIG(g \setminus \{x_i\}) \cup \{x_i\}) = GIG(g \setminus \{x_j\}) \cup \{x_j\})$.*

Axiom: Linearity. *linearity* implies that if we combine two models into a new model with a linear function, then the importance score in the new model will be a linear combination of the importance scores in the two first models. This holds true for GIG:

Proposition 5. *Let $F(x) = a \times F_1(x) + b \times F_2(x)$ for every x, GIG_f is GIG for model f, then, $GIG_F(g) = a \times GIG_{F_1}(g) + b \times GIG_{F_2}(g)$.*

In conclusion, our technique, GIG and SIG, which is a specific instance of GIG, satisfies the axioms for model explanation at the group level and the segment level. Proofs for all propositions are in supplementary material. The axioms also support the stability of our proposed method. Firstly, *Implementation invariance* property of GIG, as well as SIG, ensures that our system assigns the same importance scores set to the set of segments when using the same deep

learning model. Superadditivity property with strict equality condition *if* $G_1 \cap G_2 = \emptyset$, *then* $GIG(G_1 \cup G2) = GIG(G_1) + GIG(G_2)$ helps us design the algorithm to add up two segments' scores when merging without computing SIG again. *sensitivity (a)* and *sensitivity (b)* help our method avoid giving zero scores for useful segments and giving non-zero scores for redundant segments.

3 Experiments

3.1 Experimental Setup

Dataset. We utilize 12 time series datasets from the UCR archive for classification tasks incl. Coffee, Strawberry, GunPointOldVersusYoung (abbreviated as GPAge), HandOutlines (abbreviated as Hand), Yoga, ECG200 (abbreviated as ECG), GunPointMaleVersusFemale (abbreviated as GPGender), Dodger, Chinatown, FreezerSmallTrain (abbreviated as Freezer), HouseTwenty (abbreviated as House20), and WormsTwoClass (abbreviated as Worms). Details about the datasets are provided in the supplementary.

Deep learning architecture. We conduct experiments using 5 DL models including CNN, Transformer, FC, Dense-FCN, and LSTM-FCN (c.f. supplementary for details).

Evaluation metric. We use two evaluation metrics for our explanation of time series. The first metric is the *Faithfulness* score. This metric is described in Section 2.2. The second metric is *Robustness* described in (Sivill and Flach 2022). Both of their metrics are used in LIMESegment (Sivill and Flach 2022) in which we follow strictly their setups.

Baseline. We evaluate the proposed method by comparing it to three existing methods. The first two methods is based on LIME and are described in (Neves et al. 2021) and (Guillemé et al. 2019). And the other method is LIMESegment (Sivill and Flach 2022). These are cutting-edge methods for explaining time series models at the segment level. Please refer to supplementary for full experiments.

3.2 Results

The results are shown in Table 1. Our method surpasses the most state-of-the-art method LIMESegment, in terms of *Faithfulness* across 12 datasets up to a score of **0.487**. Additionally, InteDisUX has much better robustness than LIMESegment on 8 out of 12 datasets. The results consistently show that our method generates explanations that are more faithful and resilient to noise than those produced by current time series segment-level explanation methods.

Figure 6 illustrates the outcomes of our explanations for the dataset Dodger. These visualizations offer valuable insights into the algorithm's process of identifying significant segments for model predictions. Our technique selects the middle part of the Dodger dataset as the most important, distinguished by a high signal on game days and a low signal on regular days. The system provides a negative score to the segment that represents typical days, signifying that lower values aid in accurately forecasting days without games. In contrast, on game day, the middle segment earns a very positive score, indicating that this segment greatly contribute to the model's ability to accurately identify game days.

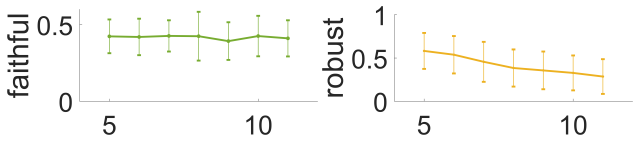

Figure 7: The impact of N_s (horizontal axis) on the model explanation for the yoga dataset.

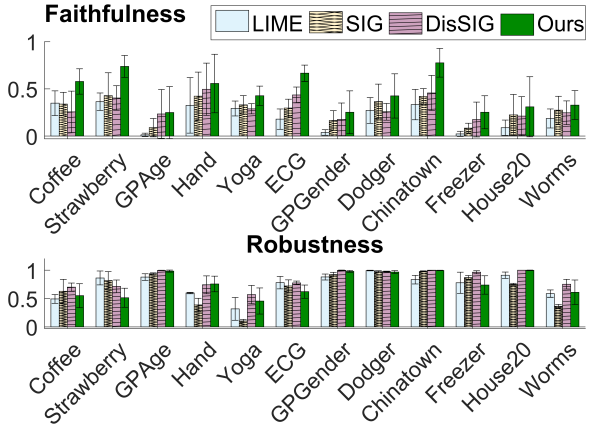

Figure 8: Study on SIG, discrimination and faithfulness gain (LIMESegment,SIGSegment, DisSIGSegment, Ours).

3.3 Ablation Studies

First, we examine the function of SIG. To assess the effectiveness of SIG, we conducted a comparison between SIG and LIME of LIMESegment. To compare this, we design a method called **SIGSegment**, which use the same segmentation in LIMESegment but using SIG instead of LIME. Moreover, our objective is to assess the benefit derived by discrimination gain, referred to as "Dis" in the InteDisUX in cycle in second stages. To do that, we design a basline algorithm called **DisSIGSegment**, which is similar to InteDisUX but do not use the *Faithfulness* score to choose optimal position to merge in second stage. Furthermore, we assess the influence of integrating the *Faithfulness* gain constraint, by comparing our approach with **DisSIGSegment**.

Study on SIG. Figure 8 presents the results of our study on SIG. We observe that SIGSegment achieves a significantly better *Faithfulness* score compared to the original LIMESegment method across 11/12 datasets. This indicates that, for the same set of segments, the SIG method assigns a more faithful score to each segment compared to LIME. However, the robustness score of SIGSegment is only better than LIMESegment in 5/12 dataset, indicating the contribution of InteDisUX also come from the set of segments.

Study on discrimination gain. In Figure 8, it is evident that DisSIGSegment is better than LIMESegment in 8/12 datasets in terms of *Faithfulness*, 8/12 datasets in terms of robustness. Compare to SIGSegment, DisSIGSegment is only better in 6/12 datasets in terms of *Faithfulness*. However, in terms of robustness, DisSIGSegment outperfrom SIGSegment in 11/12 dataset, which indicates that our

Dataset	Method	Faithful ↑	Robust ↑
Coffee	N	0.179 ± 0.132	0.214 ± 0.298
	G	0.087 ± 0.034	0.143 ± 0.071
	LIMESegment	0.348 ± 0.131	0.493 ± 0.077
	Ours	$\mathbf{0.578 \pm 0.134}$	$\mathbf{0.550 \pm 0.211}$
Strawberry	N	0.255 ± 0.043	0.137 ± 0.127
	G	0.214 ± 0.022	0.371 ± 0.329
	LIMESegment	0.364 ± 0.093	$\mathbf{0.863 \pm 0.122}$
	Ours	$\mathbf{0.738 \pm 0.116}$	0.514 ± 0.168
GPAge	N	0.008 ± 0.013	0.431 ± 0.365
	G	0.003 ± 0.004	0.509 ± 0.222
	LIMESegment	0.013 ± 0.018	0.879 ± 0.058
	Ours	$\mathbf{0.249 \pm 0.275}$	$\mathbf{0.984 \pm 0.019}$
Hand	N	0.196 ± 0.188	0.183 ± 0.196
	G	0.186 ± 0.159	0.037 ± 0.066
	LIMESegment	0.325 ± 0.294	0.597 ± 0.130
	Ours	$\mathbf{0.557 \pm 0.310}$	$\mathbf{0.757 \pm 0.137}$
Yoga	N	0.187 ± 0.048	0.247 ± 0.376
	G	0.182 ± 0.055	0.340 ± 0.448
	LIMESegment	0.292 ± 0.079	0.317 ± 0.201
	Ours	$\mathbf{0.427 \pm 0.102}$	$\mathbf{0.456 \pm 0.231}$
ECG200	N	0.217 ± 0.146	0.292 ± 0.279
	G	0.137 ± 0.088	0.090 ± 0.050
	LIMESegment	0.179 ± 0.108	$\mathbf{0.780 \pm 0.110}$
	Ours	$\mathbf{0.666 \pm 0.086}$	0.622 ± 0.117
GPGender	N	0.007 ± 0.007	0.190 ± 0.241
	G	0.006 ± 0.007	0.178 ± 0.251
	LIMESegment	0.038 ± 0.029	0.880 ± 0.053
	Ours	$\mathbf{0.252 \pm 0.227}$	$\mathbf{0.979 \pm 0.016}$
Dodger	N	0.037 ± 0.013	0.300 ± 0.282
	G	0.048 ± 0.015	0.025 ± 0.022
	LIMESegment	0.271 ± 0.137	$\mathbf{0.994 \pm 0.006}$
	Ours	$\mathbf{0.425 \pm 0.234}$	0.967 ± 0.026
Chinatown	N	0.008 ± 0.007	0.433 ± 0.261
	G	0.006 ± 0.006	0.080 ± 0.013
	LIMESegment	0.333 ± 0.161	0.833 ± 0.077
	Ours	$\mathbf{0.776 \pm 0.152}$	$\mathbf{0.996 \pm 0.003}$
Freezer	N	0.049 ± 0.062	0.333 ± 0.424
	G	0.016 ± 0.014	0.174 ± 0.159
	LIMESegment	0.023 ± 0.027	$\mathbf{0.777 \pm 0.187}$
	Ours	$\mathbf{0.252 \pm 0.175}$	0.739 ± 0.163
House20	N	0.055 ± 0.049	0.366 ± 0.176
	G	0.056 ± 0.060	0.326 ± 0.194
	LIMESegment	0.088 ± 0.081	0.911 ± 0.056
	Ours	$\mathbf{0.310 \pm 0.319}$	$\mathbf{0.998 \pm 0.004}$
Worms	N	0.119 ± 0.035	0.164 ± 0.151
	G	0.090 ± 0.015	0.192 ± 0.309
	LIMESegment	0.186 ± 0.101	0.587 ± 0.065
	Ours	$\mathbf{0.328 \pm 0.153}$	$\mathbf{0.610 \pm 0.216}$

Table 1: We evaluate the efficacy of our explanation technique, InteDisUX (InteDis), by comparing it with three existing methods: LIMESegment, the LIME-based method proposed in (Neves et al. 2021) (N), and the way proposed in (Guillemé et al. 2019) (G). We assess the effectiveness of these techniques on all 12 datasets. The evaluation scores for *Faithfulness* and robustness (Robust) were calculated by taking the average of the scores obtained from five classifier architectures.

bottom-up appoach with discrimination gain help improve robustness of time series explanation.

Study on faithfulness gain. Figure 8 demonstrates the effectiveness of integration the faithfulness gain to better candidate selection in second stage . We observe that our method consistently achieves a higher faithfulness score compared to DisSIGSegment across all 12 datasets. This finding suggests that the faithfulness gain plays a crucial role in ensuring the explainer prioritizes segments that maintain a high degree of faithfulness of the explanation. Meanwhile, using *Faithfulness* gain does not adversely affect the segmentation method's robustness.

Study on the N_s in Initialization stage. We illustrate the variations in *Faithfulness* score and robustness score based on the value of N_s in Figure 7 with yoga dataset. Extensive study about that across 12 datasets is in suplemntary material. The results reveal that the *Faithfulness* score will reach its maximum value as we increase the value of N_s, however the robustness score usually decline as N_s increases.

4 Related Work

Explaining time series model is using segments as interpretable representation is a novel approach for time series classifier explanation. Two LIME-based approach (Neves et al. 2021; Guillemé et al. 2019; Sivill and Flach 2022) split time series to same size segments and genrate important scores for segment using LIME. Motivated from data point in a segment should be homogeneous, LIMESegment (Sivill and Flach 2022) design a novel approach using nearest neighbor segmentation to explain time series model. Among them, LIMESegment is our closest work.

In time series explanation, there is another perspective, which is using attribution map or saliency map to explain time seires. Saliency map is important scores set for every feature in time series input. These scores indicates the influency of the feature for model to make decision. These methods are developed and studied in (Crabbé and Van Der Schaar 2021; Leung et al. 2022; Doddaiah et al. 2022; Enguehard 2023; Queen et al. 2024; Liu et al. 2024a). To create saliency map in time series, Integrated gradient (IG) (Sundararajan, Taly, and Yan 2017) and LIME (Ribeiro, Singh, and Guestrin 2016) are also used as baselines.

The difference among these methods and our technique have been extensively discussed in Sections 1, 2 and 3.

5 Conclusion

This paper presents a new approach called InteDisUX for explaining time series data. Unlike the previous method, we employ a method to generate a set of segments by merging them based on discrimination and faithfulness gains. Our methods do not depend on a predetermined number of segments and can utilize model information to generate segments. In addition, our scoring technique SIG not only assists the explanation in avoiding the shortcomings of LIME but also enhances the stability of the refinement process. Empirical investigations demonstrate that our approach surpasses existing works in terms of effectiveness.

Acknowledgments

This research is part-funded by the European Union (Horizon Europe 2021-2027 Framework Programme Grant Agreement number 10107245. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. The European Union cannot be held responsible for them) and by the Engineering and Physical Sciences Research Council under grant number EP/X029174/1.

References

Abdi, H.; and Williams, L. J. 2010. Principal component analysis. *Wiley Interdiscip. Rev. Comput. Stat.*, 2(4): 433–459.

Crabbé, J.; and Van Der Schaar, M. 2021. Explaining time series predictions with dynamic masks. In *ICML*, 2166–2177. PMLR.

Doddaiah, R.; Parvatharaju, P.; Rundensteiner, E.; and Hartvigsen, T. 2022. Class-Specific Explainability for Deep Time Series Classifiers. In *ICDM*, 101–110. IEEE.

Enguehard, J. 2023. Learning perturbations to explain time series predictions. In *ICML*, 9329–9342. PMLR.

Guillemé, M.; Masson, V.; Rozé, L.; and Termier, A. 2019. Agnostic local explanation for time series classification. In *ICTAI*, 432–439. IEEE.

Huttenlocher, D.; Klanderman, G.; and Rucklidge, W. 1993. Comparing images using the Hausdorff distance. *IEEE Trans. Pattern Anal. Mach.*, 15(9): 850–863.

Keller, T.; Thiesse, F.; and Fleisch, E. 2014. Classification models for RFID-based real-time detection of process events in the supply chain: An empirical study. *TMIS*, 5(4): 1–30.

Leung, K. K.; Rooke, C.; Smith, J.; Zuberi, S.; and Volkovs, M. 2022. Temporal Dependencies in Feature Importance for Time Series Prediction. In *ICLR*.

Liang, M.; Wang, X.; and Wu, S. 2023. Improving stock trend prediction through financial time series classification and temporal correlation analysis based on aligning change point. *Soft Comput.*, 27(7): 3655–3672.

Lin, L.; Xu, B.; Wu, W.; Richardson, T. W.; and Bernal, E. A. 2019. Medical Time Series Classification with Hierarchical Attention-based Temporal Convolutional Networks: A Case Study of Myotonic Dystrophy Diagnosis. In *CVPR workshops*, 83–86.

Liu, Z.; Wang, T.; Shi, J.; Zheng, X.; Chen, Z.; Song, L.; Dong, W.; Obeysekera, J.; Shirani, F.; and Luo, D. 2024a. TimeX++: Learning Time-Series Explanations with Information Bottleneck. *arXiv preprint arXiv:2405.09308*.

Liu, Z.; Zhang, Y.; Wang, T.; Wang, Z.; Luo, D.; Du, M.; Wu, M.; Wang, Y.; Chen, C.; Fan, L.; et al. 2024b. Explaining Time Series via Contrastive and Locally Sparse Perturbations. *arXiv preprint arXiv:2401.08552*.

Mohammadi Foumani, N.; Miller, L.; Tan, C. W.; Webb, G. I.; Forestier, G.; and Salehi, M. 2024. Deep learning for time series classification and extrinsic regression: A current survey. *ACM Comput. Surv.*, 56(9): 1–45.

Neves, I.; Folgado, D.; Santos, S.; Barandas, M.; Campagner, A.; Ronzio, L.; Cabitza, F.; and Gamboa, H. 2021. Interpretable heartbeat classification using local model-agnostic explanations on ECGs. *Comput. Biol. Med.*, 133: 104393.

Queen, O.; Hartvigsen, T.; Koker, T.; He, H.; Tsiligkaridis, T.; and Zitnik, M. 2024. Encoding time-series explanations through self-supervised model behavior consistency. *NeurIPs*, 36.

Read, S. J.; and Marcus-Newhall, A. 1993. Explanatory coherence in social explanations: A parallel distributed processing account. *J. Pers. Soc. Psychol.*, 65(3): 429.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *KDD*, 1135–1144.

Rojat, T.; Puget, R.; Filliat, D.; Del Ser, J.; Gelin, R.; and Díaz-Rodríguez, N. 2021. Explainable artificial intelligence (xai) on timeseries data: A survey. *arXiv preprint arXiv:2104.00950*.

Senin, P. 2008. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, 855(1-23): 40.

Singh, C.; Murdoch, W. J.; and Yu, B. 2018. Hierarchical interpretations for neural network predictions. *arXiv preprint arXiv:1806.05337*.

Sivill, T.; and Flach, P. 2022. Limesegment: Meaningful, realistic time series explanations. In *AISTATS*, 3418–3433. PMLR.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *ICML*, 3319–3328. PMLR.

Tan, Z.; Tian, Y.; and Li, J. 2024. GLIME: general, stable and local LIME explanation. *NeurIPs*, 36.