Interpretability of Deep Neural Networks for Pre-Symptomatic Pathogen Exposure Detection

by

Maceo Richards

A Major Qualifying Project

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science

in Data Science

by

May 2023

APPROVED:

Liu, Xiaozhong

Abstract

The work described in this proposal outlines a process for exploring interpretability of deep neural network architectures designed for a specific domain application within time series classification. Work presented here is the continuation of an endeavor to explore pre-symptomatic pathogen exposure detection given multimodal time series physiology data. Previously, no studies have been conducted assessing interpretability of the algorithms being developed for this purpose. There is a necessity for this work to be done, as exploring interpretability is ultimately the driving force of trustworthiness between end users and the artificial intelligence platforms they will operate. The topic of pre-symptomatic pathogen exposure detection is within the broader domain of time series classification, and challenges within this domain are outlined. Various contending methods for addressing these challenges are also discussed. This report primarily details an adaptation of LIME (Local Interpretable Model-agnostic Explanations) to time series classification called LIMESegment and its implementation in the task of pre-symptomatic pathogen exposure detection.

Contents

List of Figures

Chapter 1

Introduction

1.1 Wearable Devices

The miniaturization of electronic components has initiated a wave of innovation in the field of wearable sensing, specifically for health informatics. In addition to this, more processing power than ever has been available to consumers so they may gain useful inferencing from collected data. This increasing access to wearables technology was another driver of the motivational backing for the commencement of this pilot study. While current practices for pathogen exposure detection often rely on the direct observation of symptoms, a system designed for pathogen exposure early warning would allow for public health measures reducing transmission to be implemented at a quicker rate.

1.2 Initial Work

In 2017, the work of [MDP+17] was a pilot study conducted in which a model was able to detect asymptomatic states while still in the incubation period given several modalities of physiological data coming from other non-human primate studies. These were studies of exposure to viruses (Nipah, Lassa, Ebola, Marburg) and bacteria (Y. pestis) in which data was recorded over a period of time to measure physiological response. After some processing, this data was passed to a random

forest binary classification algorithm for supervised learning. Here, the two classes being predicted were pre-exposure and post-exposure. The logic for this presented in the work of [MDP+17] follows that the act of "infection" is not a discrete event but rather a time-dependent process, whereas pre-exposure and post-exposure are more easily defined. Because every subject was exposed to some viral or bacterial agent, this analysis had equal balancing between the two prediction classes. Post-exposure was able to be detected well before the onset of fever in most subjects. In [MDP+17], researchers identified that features derived from blood pressure, temperature, and electrocardiography (ECG) were the most indicative of exposure. Researchers in [MDP+17] limited features to only those that may be recorded from wearable devices and observed comparable model performance. This ties in to the initial motivation for the pilot study, the intention of which was to be proof of concept that pre-symptomatic agent exposure may be predicted from non-invasive physiological data.

1.3 PRESAGED

Now, many years later, the PRESAGED initiative has continued this exploration of pre-symptomatic exposure detection. Numerous studies provide multimodal time series data from a variety of wearable devices. The set of devices providing wearable data include the Oura ring, Garmin watch, and Fitbit watch. While all provide data pertaining to heart rate and sleep quality, the Oura ring exclusively also provides data regarding heart rate variability, skin temperature, MET (metabolic equivalent, often representative of heightened activity levels), and respiratory rate. Similarly, while the Garmin watch and Fitbit watch both track step count, Garmin additionally provides their own proprietary features they call "stress" and "body battery". While there is overlap in the type of information coming from these devices, there

are also a number of discrepancies. Along with the wearables data, these studies provide corresponding survey data containing information about the study participants. The survey data is comprised of numerous questions regarding different aspects of the participant, whether that be their demography, medical history, medication they take, symptoms they experience, or diagnostic results. Examples of questions include those about race or ethnicity, serious previous illness or underlying conditions, and symptom detection among others. This study data must all be harmonized before it is inserted into a centralized database as there is little to no communication between the organizations conducting the studies. This harmonization process entails the transforming of data from the original format to one that is compatible with a centralized database and standardized across all studies. For an example from the wearables data, the Oura ring samples heart rate at 5-minute intervals while the Garmin watch samples heart rate at 15-second intervals and the Fitbit watch samples heart rate at 1-minute intervals. This issue persists across all data modalities from the various wearable devices, and methods such as decimation or interpolation must be used to reduce or increase the data sampling rate, respectively. This wearables data is used to train random forest and deep neural network (DNN) models that attempt to classify pathogen exposure or non-exposure in study participants.

1.4 Black Box Problem

The majority of machine learning algorithms in use today are DNN models and have driven significant advancements in fields such as computer vision and natural language processing. DNN models similarly have the potential to capture information within time series data, although current implementations lack stability and

hence have not been widely accepted for practical use [LZ21]. The acceptance of machine learning algorithms depends on their trustworthiness [Ign20]. Trustworthiness is an inherent problem in deep learning due to the fact that DNN models classify as black box algorithms, meaning how they operate is not directly human understandable. This drives the need for development of interpretability methods designed to assess exactly how these complex learning systems generate predictions. Explainable artificial intelligence (XAI), a relatively new field within machine learning research, has gained significance within recent years in response to the increasing necessity of trustable explanations for deep neural networks and their observed behavior. As XAI is still a new field, there is no strict definition for interpretability. While some studies in XAI have analyzed how well a model is able to determine cause and effect relationships, others have endeavored to explore both what model internal mechanisms represent and the overall importance of these mechanisms in prediction performance. In this work, model interpretability is assessed by identifying salient components of the input data that contribute most during inferencing.

Chapter 2

Background

2.1 Saliency Methods

Many solutions for evaluating saliency of input data have been explored, and a number of these broadly classify as gradient-based methods. While many variations of gradient-based methods exist for differing applications, they typically backpropagate relevance scores from the prediction output through the internal mechanisms of the model until the input is reached. These techniques allow saliency maps to be generated for an input given both the model and the corresponding prediction output. A method proposed by [SGK17] known as DeepLIFT first establishes a reference activation for each neuron in the DNN model. Once an input is passed to the black box model, a comparison is drawn between the activation of all neurons and their respective reference activation. Contribution scores are assigned based on the difference between each activation and its reference activation, and these scores are then backpropagated across all neurons in the DNN model to each input feature. Another method known as integrated gradients proposed in the work of [STY17] begins with the assumption that any DNN model may be represented by a function F. This method then allows any input to be represented by x and also establishes a baseline input, denoted by x'. Gradients along all points on the straight-line path from the baseline x' to the input x are computed, and then the path integral of these gradients along this straight-line path define the integrated gradients for the input

x. This may be more easily understood as an averaging over gradients of the input space while a specific input to be explained changes, or moves away, from a baseline that has previously been established as non-informative. Perturbation methods are another class of interpretability frameworks and involve removing specific sections of data or entire features from the input. In their evaluation, prediction accuracy on perturbed inputs is compared to that of the original instances. The logic behind these methods is that more salient sections of data or features, when perturbed, will result in a greater accuracy loss than sections of data or features that carry less meaningful information.

2.2 LIME

One of the most popular techniques in use today for exploring interpretability of deep neural networks is a perturbation technique known as Local Interpretable Model-agnostic Explanations (LIME) proposed in the work of [RSG16]. This method attempts to approximate the prediction behavior of a black box model when given an individual test case with an interpretable model. This interpretable model is a linear model where each predictor corresponds to an interpretable representation within the data and the weight for each predictor corresponds to the importance of its respective presence or absence. Therefore, this technique allows for interpretable representations that are important in model prediction for a local test case to be identified. In the implementation of this method, a test case is first split into interpretable representations. Samples are then generated around this test case where for each sample interpretable representations are randomly perturbed from the data. These generated samples are then given to the black box model to obtain their labels. The generated samples and their labels comprise the

dataset on which the linear model is trained, with each sample weighted by a distance metric that determines its relative locality to the original test case. The work of [RSG16] utilizes the coefficients from the linear model as the explanations for the original test case.

2.3 Time Series Interpretability

Interpretability methods that generate explanations in the form of feature importance scores often fail to consider the temporal relationships within time series data. In 2020, a method proposed by [TJC+20] known as FIT determines the relative importance of observations by comparing their contributions to the distributional shift of a black box model. FIT is then adapted in the work of [RSL+21] to produce WinIT, a method that analyzes how groups of observations effect distributional shift of a black box model. An alternative method CEM, devised in 2020 by [LZC20], assigns feature attribution scores by identifying the minimal perturbation necessary to for a black box model to change the classification of a time series. Other frameworks closest to the one proposed here are LEFTIST, suggested by [GMRT19], and one proposed by [NFS+21]. These are considered closer in nature of their implementation due to the reasoning that they both attempt a direct adaptation of LIME to time series classification.

2.4 Challenges Adapting LIME

While LIME has been well explored in image classification, its adaptation to time series classification gives rise to several challenges. The first of these challenges is that no one method exists for meaningfully segmenting time series data into interpretable representations, and the various methods that previously existed have their

individual drawbacks. These interpretable representations should be identifiable homogeneous super segments, meaning they correspond to distinguishable temporal patterns within the data. In any image, these interpretable representations are recognizable clusters of pixels dubbed "super pixels". While super pixels may easily be visually identified in images, there is no visual homolog to this within time series data. This lack of visual interpretation is what drives the necessity for meaningful segmentation techniques. Because pathogenic exposure is assumed to be a process that occurs on the order of days as opposed to hours in previous work for this study [MDP+17], an ideal segmentation method will be sensitive to latent changes within the time series data. In 2017, the work of [GDY+17] devises a method for segmentation which assumes that homogeneous segments are comprised of many short sub-sequences that all share latent characteristics and vary in similarity. The work of [ZINK18] alternatively provides a method that searches for "time series chains". These may be thought of as a consecutive series of short sub-sequences that are comparatively higher in similarity than other sub-sequences in the time series.

Another challenge is that no standard methodology for generating realistic background content of a time series exists. Ideally, any background content generated to replace perturbed sections of data would be comprised of latent waveform characteristics of the original time series. In image classification, perturbed super pixels historically were replaced with constant values, injected noise, or blurring. It was not until recent years that methods, known as inpainters, for producing realistic background content for an image became popular. The work of [AN20] suggests an inpainter for replacing perturbed interpretable representations during LIME implementation and demonstrates that this far out performs methods that do not generate realistic background content. In the case of physiological data, this background content would be any segment of data that corresponds to healthy, restful states. It

is assumed that these states, being minimally influenced by outside stimuli, best represent the true health state of an individual.

Lastly, it has remained uncertain how distance should be measured between two time series such that their true locality to each other is accurately reflected in a time series LIME adaptation. This metric should indicate that two time series are closer together the more identical they are to each other in the specific context of the application being studied. Through previous experimentation with LIME in image classification, it has been determined that the best distance metric for this interpretability method is dependent on the number of inactivated super pixels in a given image that are represented in a reference image [GM21]. This would not hold true applied to time series classification as the binary presence or absence of a super segment within a time series does not encompass any information pertaining to its global location within the data. A method for computing distance between two time series known as Dynamic Time Warping (DTW) was proposed by [BK59] in 1959 and has proven to be a staple for comparing time series. This method is able to compare two time series even if they are not temporally aligned. Following certain guidelines, each index from one sequence is mapped to an index in the other sequence and vice versa. An optimal match between indices is defined as one that both satisfies all criteria and minimizes a cost function, typically the sum of absolute differences.

2.5 Project Goals

The aim of this work is to address these challenges within the specific application of pre-symptomatic pathogen exposure detection. A first goal is to identify meaningful super segments of physiology data that correspond to distinct temporal

patterns. This will enable end users to analyze these super segments and establish their distinguishing characteristics. Eventually, this information could be used to search for common or similar super segments across multiple test cases. If a super segment is common across multiple test cases and is also determined to be uninformative or misleading during prediction, testing whether its removal from the data improves model performance is justified. Another goal of this work is to determine latent characteristics of physiology data for a given individual across a variety of multimodal inputs. Akin to reasons for addressing the previous challenge, common or similar characteristics of background content across test cases for all data channels may be identified and experimentally perturbed. The knowledge gained from such analyses could provide the logical reasoning for future data cleaning and preprocessing strategies. A final goal is to identify specific super segments of physiology data that are suggestive of either pathogen exposure or non-exposure. This will allow end users to examine positive test cases and know exactly which components of the data were most important in model prediction. Pathogen exposure detection models ideally would predict positive with minimal time having passed since viral contraction. This necessitates faster recognition of super segments that are indicative of pathogen exposure. The need for analyzing super segments that correlate to pathogen non-exposure is driven by a desire for increased confidence that the black box model is not generating false negative predictions.

Chapter 3

Methods

3.1 LIMESegment and NNSegment

An implementation from the work of [SF22] called LIMESegment is currently the best performing adaptation of LIME to time series classification. The segmentation framework proposed in their report is called Nearest Neighbors Segment (NNSegment) and addresses the aforementioned challenge of meaningful segmentation. This framework disputes the assumption that super segments are comprised only of similar sub-sequences and instead poses that a time series is better characterized as a mixture of both super segments similar in shape and anomalous super segments. This segmentation methodology is designed to classify a larger variety of time series segments and has shown itself in the LIMESegment paper [SF22] to be more versatile than other methods proposed in literature.

3.2 RBP

To address the need for realistic background content generation, the work of [SF22] also proposes a method Realistic Background Perturbations (RBP) as part of the LIMESegment algorithm. This method first transforms any time series to be explained into the frequency domain and identifies the frequency bin with highest representation and lowest variance. It is assumed in the work of [SF22] that this

frequency bin most closely approximates realistic background content when converted back to time domain. Other methods such as noise injection and adding constant values have been used in the past to replace perturbed segments, however the LIMESegment paper [SF22] demonstrates that RBP yields improved results over these methods.

3.3 DTW

The work of [SF22] finally proposes the use of Dynamic Time Warping (DTW) as the distance metric for establishing locality to the original instance when generating LIMESegment explanations. The LIMESegment paper [SF22] makes use of a computationally optimized DTW implementation known as FastDTW. While the original DTW implementation has O(n2) complexity, the FastDTW implementation reduces the complexity to O(n) [SC07]. The LIMESegment paper [SF22] also demonstrates that this method produces more stable explanations than other time series distance metrics such as Euclidean distance.

3.4 Code Adaptations

Fortunately for this work, the authors of the LIMESegment paper [SF22] made their code implementation available on GitHub. While this provided a strong baseline to start from, many changes were required to adapt their implementation to this specific application. For example, the provided implementation was built almost entirely using a python library known as NumPy. This library is popular for its ability to balance efficient computation with ease of use. PyTorch, the library predominantly used to build the PRESAGED models, is also popular but for ease of machine learning development. As a result, data conversions between the two

libraries were required. Another reason for change was that while the PRESAGED data is multi-channel, the provided LIMESegment implementation only supports single-channel time series data. Because of this, the original code was modified to support multi-channel inputs. This entailed first the identification of segments and super segments for each data channel in a time series instance and then the comparative weighting of all super segments across data channels while generating explanations.

3.5 Model Frameworks

A prime goal of the PRESAGED program is to develop models that leverage performance with interpretability. For this reason, the multivariate LIMESegment adaptation was used to evaluate two model frameworks in addition to the original best performer before this analysis took place. These two model frameworks were devised specifically to potentially yield a gain in interpretability over the original model. The original model framework (CNN-LSTM) is comprised of a convolutional (CNN) layer series that then passes feature representations of the data to a series of long short-term memory (LSTM) layers. The reasoning behind this framework is that the CNN layers will reduce dimensionality along the time axis and allow the LSTM layers to learn patterns within this learned data representation. The first of the two new model frameworks to be assessed (LSTM-CNN) switches the order of the CNN and LSTM layers. Support in literature exists for application of this DNN model framework to time series classification. In the work of [KMD19], ablation tests were performed to evaluate the LSTM-CNN framework and its sub-components. These ablation tests demonstrated that the LSTM and CNN sub-components yield higher model performance when adjoined in this respective manner. The second of

the two new model frameworks to be evaluated for interpretability (CNN-TREN) is akin to the original except that it replaces the LSTM layers with transformer encoder layers. In 2017, the work of [VSP+17] disrupted machine learning research with the proposition of the transformer model architecture and demonstrated how its application across numerous domains yielded improved results over previous best performers. Transformers make use of the attention mechanism, which is their proposed technique for identifying important contextual components of an input. In the original transformer model architecture, transformer encoder layers reduce the input to a compact feature representation. Transformer decoder layers then process this information to generate output predictions. Today, many state-of-the-art models working with sequence data utilize transformer encoder layers. The attention mechanism has also been used to generate attention maps highlighting regions of the input space that the black box model identified as important. While attention maps are not generated in this work, the logic behind incorporating transformer encoder layers into the model framework is that the attention mechanism may help the black box model accurately identify important segments of data thus aiding in interpretability.

3.6 Model Hyperparameter Search

A limited hyperparameter search optimizing validation prediction performance was conducted for each of the three model frameworks to be evaluated. These hyperparameters define aspects of the neural network configuration, for example the number of layers and neurons in each layer. Constraints were placed on the specific configuration of each model framework. While performance is held in high regard, the specific aim of this work is to address interpretability. It is assumed

here that the three model frameworks presented are of equal performance quality, as further tuning would likely result in slightly higher test scores.

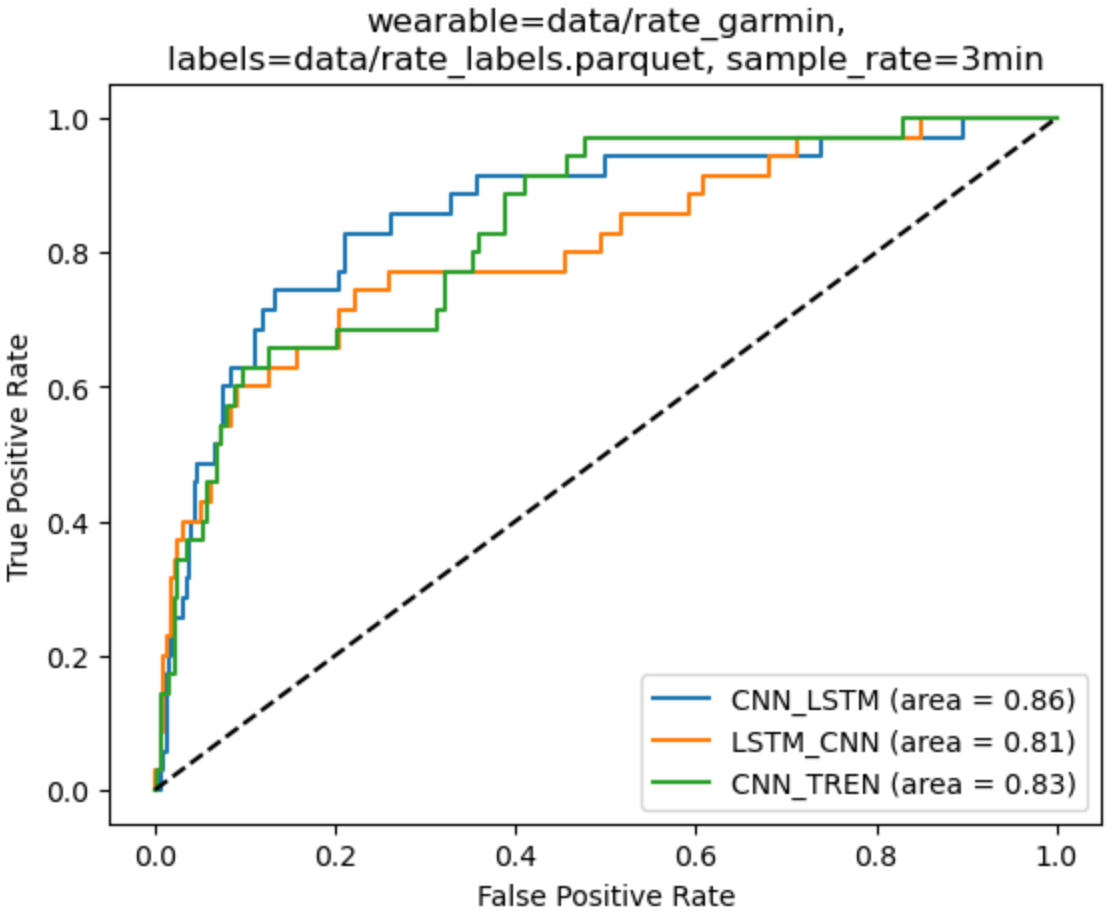

Figure 3.1: AUCROC plots for each of the three model frameworks being evaluated.

Figure 3.1 illustrates the respective AUCROC scores for each of the three model frameworks being evaluated. Ideally, a model is able to correctly classify all positive cases while minimally producing false positive predictions. AUCROC is a metric for evaluating the performance of a classification model in this regard. While performance is held in high standards, the specific aim of this work is to address interpretability. While Figure 3.1 demonstrates that the original CNN-LSTM framework is the best performer, it is assumed here that the three model frameworks presented

are of equal performance quality. Due to the limited nature of the hyperparameter search, it is assumed that further tuning would likely result in slightly higher test scores for the two newly proposed model frameworks.

Chapter 4

Results

4.1 Issue with Feature Attribution Techniques

Before proceeding with the results of this work, it is important to address common problems in literature with feature attribution techniques. The work of [ZBRS22] discusses these problems, first highlighting that no consensus exists throughout literature for a strict definition of what "attribution" means. The result of this is the constant production of new feature attribution techniques, most of which have yet to be systematically evaluated across disciplines. [ZBRS22] then draws attention to another complication, that being the nonexistence of labels for ground truth feature attributions. While LIME is among both the most explored and most successful interpretability frameworks, no ground truth examples of interpretable representations from the data exist for this specific application of pre-symptomatic pathogen exposure detection. Therefore, nothing may be held in comparison with the results of this work to establish its quality. A person may analyze an input and observe higher than normal heart rate and a decreased step count in the days leading up to a positive COVID test, but there is no way to single out either of these and conclude that one was truly more a biproduct of virus contraction than the other. This brings to light other key inquiries such as the true length of the most influential super segment and whether all components of the super segment are of equal importance.

4.2 Example Explanations

Figure 4.1 illustrates the explanations generated for one example instance of a positive test case by multivariate LIMESegment applied to each of the three model frameworks. Red segments denote those regions where LIMESegment believes the model associated high importance for positive prediction and blue indicates where LIMESegment claims the black box model interprets the data to yield a negative prediction. Darker hues of red and blue are indicative of stronger importance for positive and negative classification, respectively. While the multivariate LIMESeg-

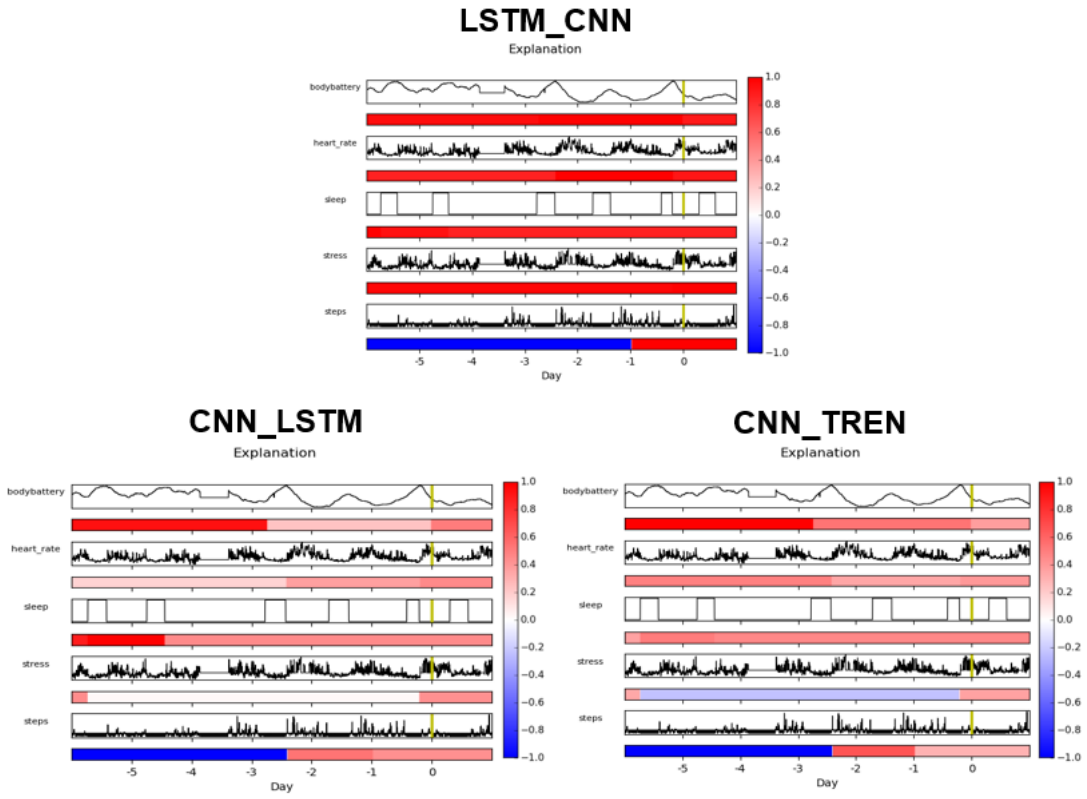

Figure 4.1: Example explanations generated by the multivariate LIMESegment adaptation across all model frameworks for one specific test case. The colorbar to the right of each figure denotes saliency.

ment implementation is able to generate explanations, it is currently unclear how the highly salient super segments are indicative of positive COVID contraction.

4.3 LIMESegment Weaknesses

While [SF22] has shown LIMESegment to be more successful than other interpretability frameworks proposed in literature, they also highlight its key weaknesses. A first weakness is dependency of the performance of LIMESegment on the proper fitting of parameters passed to the framework. The parameters of utmost importance are the window size and the number of change points, in respective order. The window size parameter controls the length of segments to be identified within the time series, and fitting this parameter to a specific dataset is a common problem in time series data mining. The parameter representing number of change points is one less than the total number of super segments identified in each time series channel, and specifying more or less change points controls the granularity with which explanations are generated. Another weakness of the interpretability framework is that NNSegment does not consider the frequency coherence assumption. This is the assumption that neighboring frequency bands similarly influence black box model behavior. While this is suggestive that better results could be obtained by segmenting time series data in the frequency domain, LIMESegment [SF22] presents the argument that time domain segmentation is the most human-interpretable. It is for this reason in light of interpretability that they choose to neglect the frequency coherence assumption in NNSegment. A final drawback of LIMESegment is that it is still an incomplete framework. In addition to RBP being rather unexplored, the original LIMESegment implementation was only tested on single-channel time series data having relatively short length and no missing values. The [SF22] paper

specifically states that LIMESegment adaptations designed to handle larger length, multi-channel time series with missing data are left to future work. This directly describes the work presented in this report as all data in usage comes from wearable devices providing long-sequence-length, multimodal inputs where missing is a common issue.

4.4 Quantitative Metrics for Interpretability

There are a multitude of factors potentially inhibiting performance of interpretability frameworks across domains, and many of them lack a quantitative metric for meaningfully evaluating the performance of the framework. As a result, many interpretability techniques use saliency maps to visually evaluate their explanations. This is not optimal for applications within time series classification as the data if often not visually interpretable, thus driving a need for quantitative metrics to assess interpretability frameworks. This then begs the question of what constitutes better performance of an interpretability framework. Ideally, small anomalous observations within the data should not have influence on the explanations being generated by the interpretability framework. The LIMESegment paper [SF22] first proposes a metric Robustness, which is defined as the robustness of the interpretability framework to small added noise. For each instance to be explained, a noisy instance is generated by adding a small amount of Gaussian noise to the raw signal. LIMESegment explanations are generated for both, and the Robustness score increases if the explanation produced for the noisy instance is the same as that for the original instance. What the Robustness score measures is the ratio of original instances to be explained in which the original explanation is equivalent to the noisy explanation. Another ideal property of an interpretability framework is that it is able to identify highly salient

super segments within the data. The LIMESegment paper [SF22] proposes a metric Faithfulness which addresses this concern. Faithfulness is calculated by first identifying the most salient super segment in each instance to explain and then generating a perturbed instance in which the most salient super segment has been perturbed from the data. For each instance to explain, the difference in prediction confidence for the black box model is calculated between the original instance and the copy instance with the identified most salient super segment perturbed. These differences are then averaged across all instances to explain. Therefore, the Faithfulness score measures the mean drop in prediction confidence across all instances to be explained when their identified most salient super segment has been perturbed.

4.5 LIMESegment Evaluation

Many different combinations of parameters were tried in the application of LIMESegment to all model frameworks. Large scale window sizes of 6-hour, 12-hour, and 24-hour periods were tested in conjunction with 1, 2, and 3 change points specified per data channel. With all input features z-score normalized (remove mean and divide by standard deviation) before being passed to a black box model for inferencing, Gaussian noise of mean 0 and standard deviation 0.000001 was added to each instance in the generation of noisy instances. 1,000 local samples were generated for each instance to be explained to guarantee explanation stability.

	CNN_LSTM		LSTM_CNN		CNN_TREN	
Number of change points	1	2	1	2	1	2
Robustness	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Faithfulness	3.6%	-0.8%	-6.2%	-1.0%	2.6%	5.5%

Figure 4.2: Tabular display for initial results of the multivariate LIMESegment adaptation.

Table 4.2 displays the resulting Robustness and Faithfulness scores for the multivariate LIMESegment adaptation applied to each of the three model frameworks, specifying 1 and 2 change points with a 6-hour window size. It may first be noted that Robustness scores for all trials were 0.0%, informing that the current multivariate LIMESegment adaptation with the specified parameters is not at all robust to small observational anomalies within the data. It may also be noted that while the Faithfulness scores are non-zero, removal of the most salient super segment identified by the multivariate LIMESegment adaptation either minimally decreases the prediction confidence of the black box model or in some cases even boosts the prediction confidence. These results indicate that the multivariate LIMESegment adaptation is currently untrustworthy and will require larger efforts to obtain meaningful results.

Chapter 5

Discussion

5.1 RBP Exploration

While this method has been shown to produce better realistic background content than other methods in [SF22], it is still in need of refinement. Although RBP does show that it is capable of producing near-realistic background content for some time series in this work, it is not a versatile method to produce realistic background content for all time series. This requires a series of methodologies that handle realistic background content generation for each data channel. Each of these methods should be independently evaluated for quality with a slightly modified version of the technique originally used to evaluate RBP in [SF22]. For all input data, split each input by data channel and create groupings where each group is comprised of all input sequences for only one data channel. Then, for each grouping of data channels, utilize the corresponding background content generation method to produce perturbed instances of every input sequence in the grouping. The number of segments to replace with realistic background content and the length of each segment are left to the developer. Once this has been done, create a dataset for each channel containing all the original input sequences, all the perturbed input sequences, as well as corresponding binary labels indicating whether each sequence is "original" or "perturbed". After this, train an individual DNN classifier on each dataset to predict whether each sequence is an original input or a perturbed input.

Higher quality methods for generating realistic background content will result in poorer prediction performance of these DNN classifiers. This evaluation technique follows the logic that it will be harder for a DNN classifier to differentiate between "original" and "perturbed" instances if the background content generation method is more realistic.

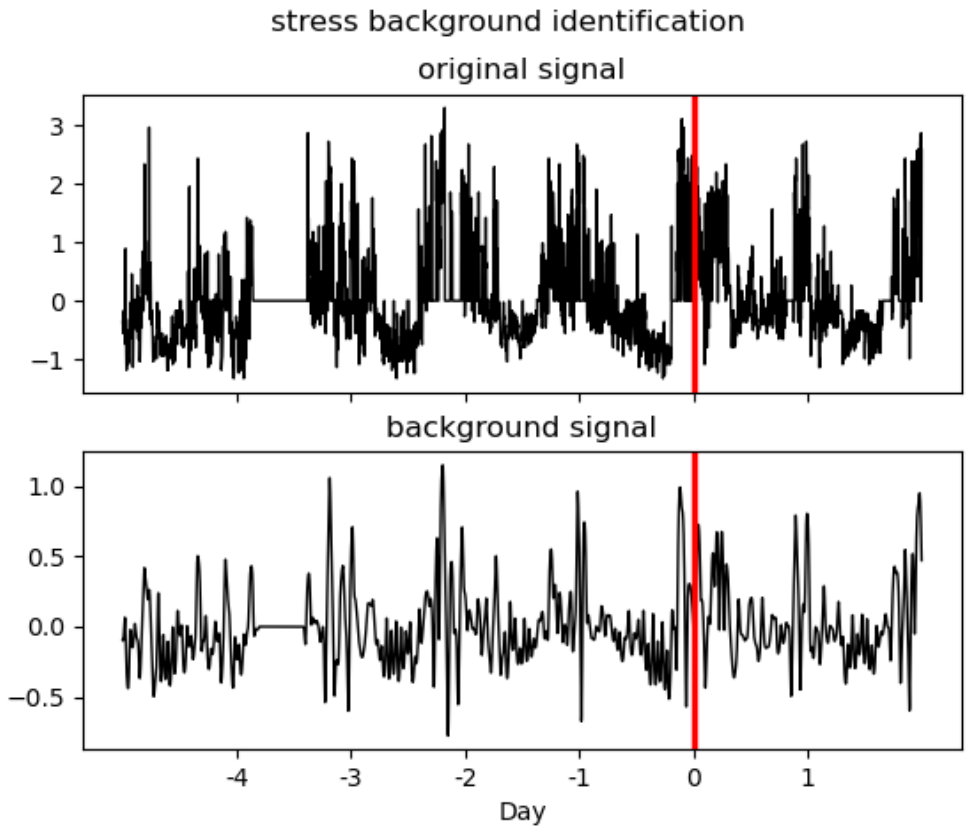

Figure 5.1: Example stress channel with corresponding background content generated by original RBP implementation. X-axis units are days since positive COVID test result. Y-axis is unitless, input data was z-score normalized before passed to black box classifier.

Figure 5.1 shows the realistic background content generated by RBP for the stress data channel, which is appears to be near-realistic. RBP clearly demonstrates its ability to identify latent trends within, and characterize the shape of, time series

of this type. As there are no smooth trends within this data, time series of this type could be classified as "stochastic" by nature. This drives the necessity of a "stochastic" method to produce realistic background content for time series data of this type. While the original RBP implementation may provide this, it must be evaluated against other methods to determine its quality.

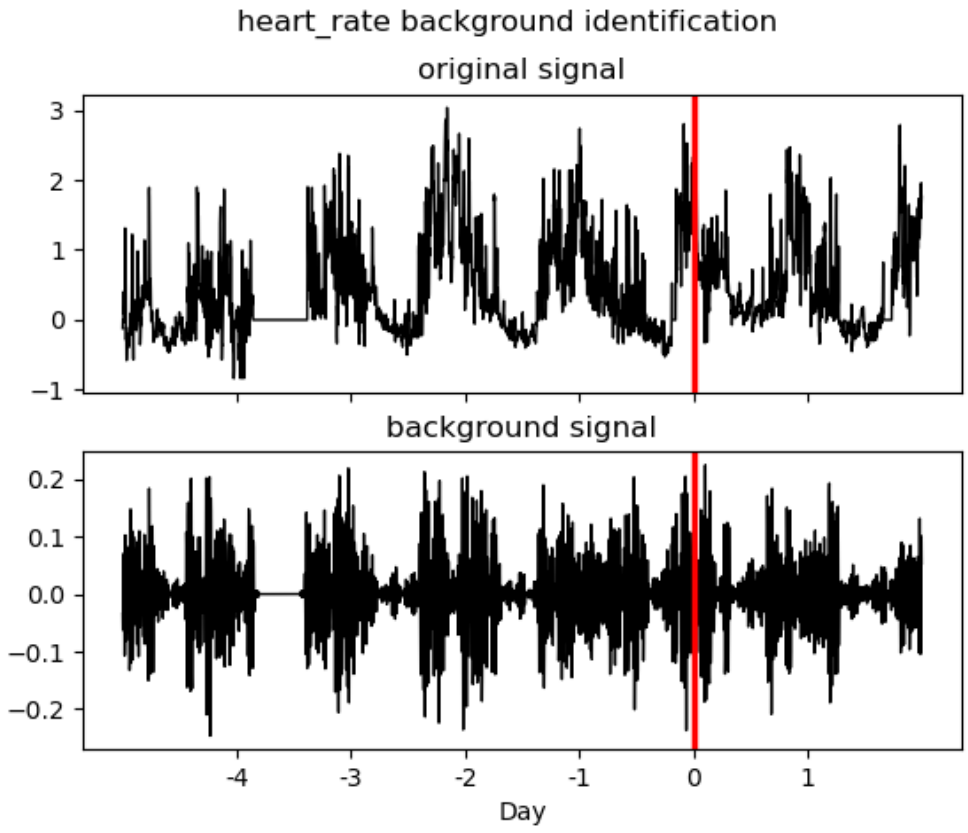

Figure 5.2: Example heart rate channel with corresponding background content generated by original RBP implementation. X-axis units are days since positive COVID test result. Y-axis is unitless, input data was z-score normalized before passed to black box classifier.

Figure 5.2 illustrates how RBP defines realistic background content for the heart rate channel. It appears to be characterized by rather low heart rates with increased variance during periods, which is indicative of restful states and therefore not un-

reasonable. Similar to the stress data channel, the heart rate data channel also requires a "stochastic" method to produce realistic background content. This raises the question of whether the same approach may be used to produce realistic background content for both the stress and heart rate data channels. Whether RBP provides an optimal solution remains to be determined through evaluation.

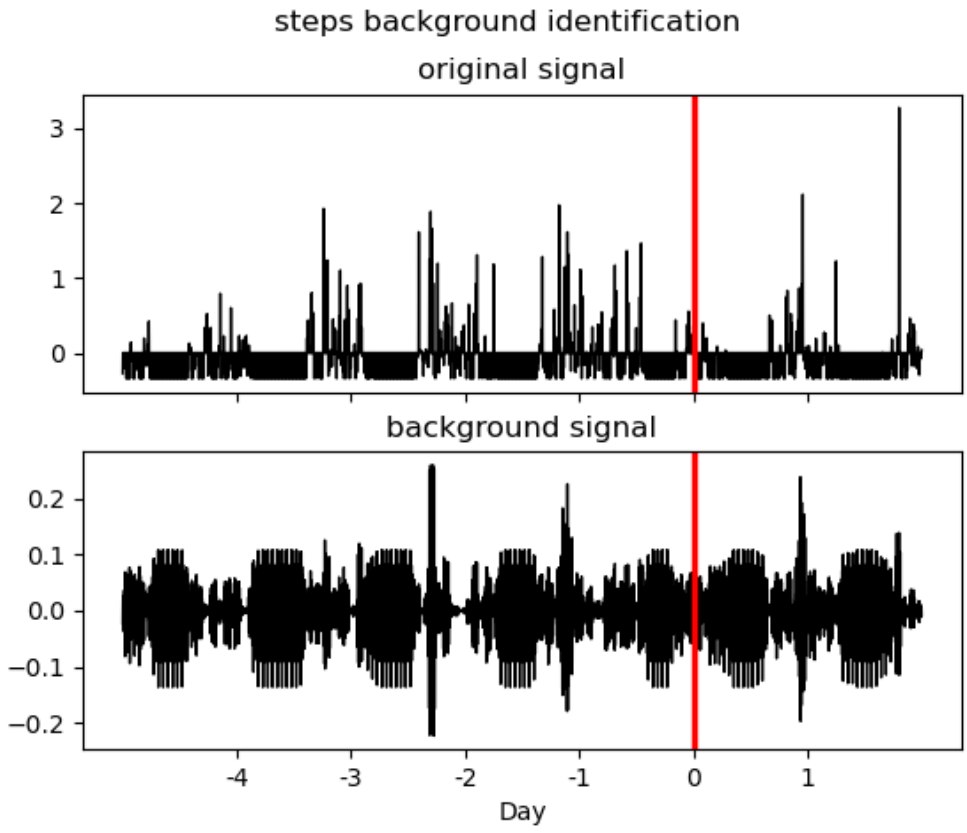

Figure 5.3: Example steps channel with corresponding background content generated by original RBP implementation. X-axis units are days since positive COVID test result. Y-axis is unitless, input data was z-score normalized before passed to black box classifier.

Similar to heart rate, it may be deduced by analyzing Figure 5.3 that RBP identifies realistic background content for the steps channel as low walking activity with increased variance in steps at certain periods. However, this characterization

of background content for the steps channel assumes that participants will exhibit walking behaviors while they sleep. Therefore, this is not realistic and other techniques to produce background content for steps should be developed. Since steps are considered count data, time series of this nature may require a "count" method to produce realistic background content. It remains to be determined whether a method more appropriately developed for count data produces better background content than the original RBP implementation.

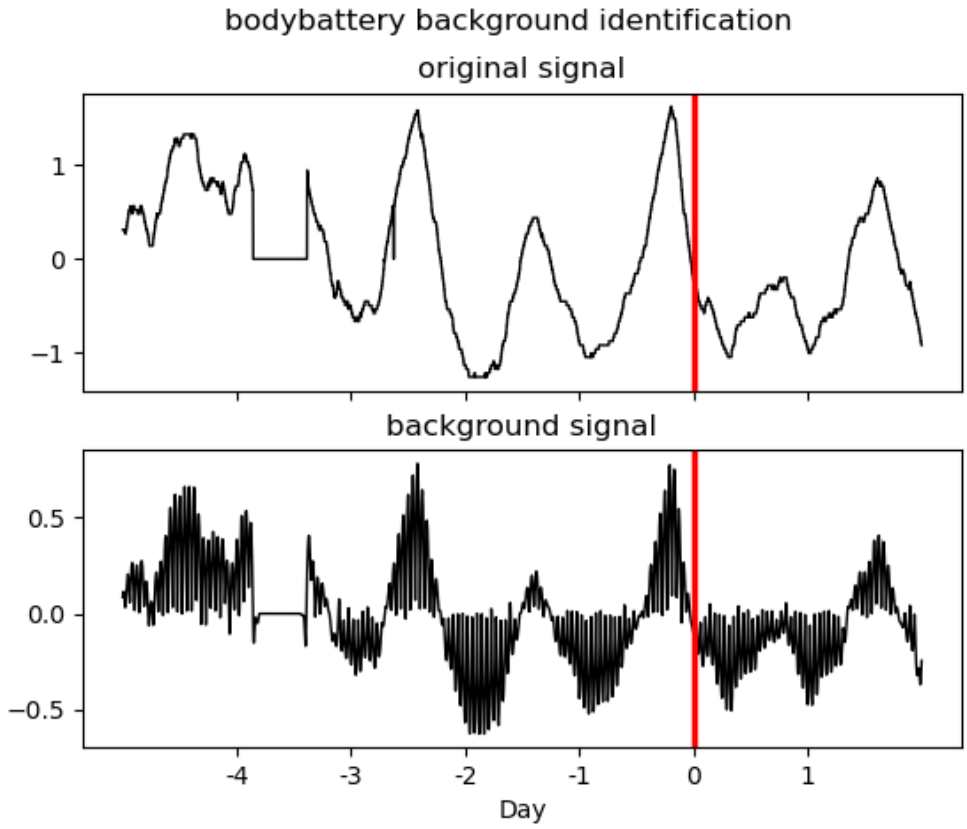

Figure 5.4: Example body battery channel with corresponding background content generated by original RBP implementation. X-axis units are days since positive COVID test result. Y-axis is unitless, input data was z-score normalized before passed to black box classifier.

Figure 5.4 demonstrates the first case where it is visually evident that RBP does

not produce realistic background content for all time series data. While RBP is able to characterize the shape of the original time series, the background content produced here is very stochastic whereas the original data appears to be rather smooth in nature. The body battery data channel is loosely interpreted as the energy level of a participant at any given time, and sensibly does not exhibit rapid spiking behavior. This makes the development of a "smooth" method for producing realistic background content necessary.

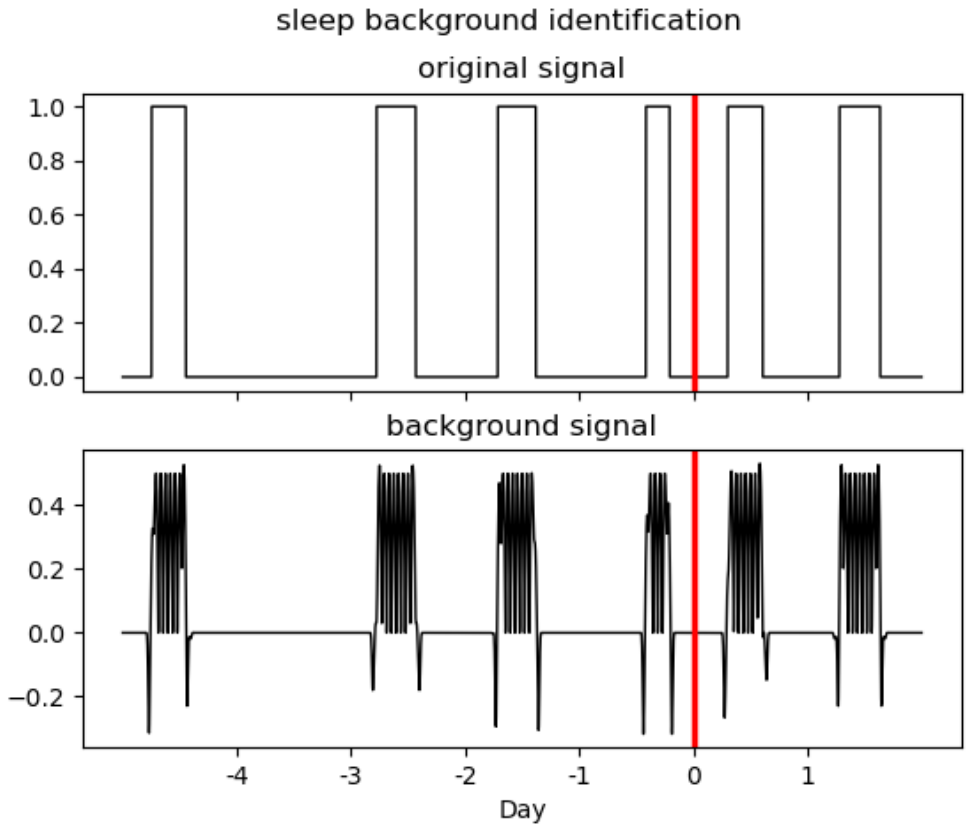

Figure 5.5: Example sleep channel with corresponding background content generated by original RBP implementation. X-axis units are days since positive COVID test result. Y-axis is a binary sequence indicating either sleep or awake states.

RBP exhibits the worst performance for this last data channel illustrated in Figure 5.5, that being the sleep category. This data channel is a binary sequence

indicating that a participant is in either a sleep state or an awake state. RBP assumes this channel contains signal data, and the resulting background content generated is far from realistic. A "binary" method for producing realistic background content is desired for time series of this type, and may be as simple as changing each data point within the perturbation region to the alternate state.

It is evident that the current RBP method does not produce high quality background content for all data channels. The development of a realistic background generation tool designed specifically for this case is required. Ideally, this tool would separately implement "stochastic", "count", "smooth", and "binary" methods for producing realistic background content. RBP is a starting point, and these other slightly modified techniques should be evaluated against it as a baseline.

5.2 LIMESegment Parameter Optimization

The case-specific sensitivity of LIMESegment performance on appropriate window size and change point parameters presents difficulty in fitting the interpretability framework to an application. This difficulty presents itself as the need for an extraordinarily exhaustive search over the space of possible parameters to optimize LIMESegment performance. Constraints placed on the window size parameter are that it is an integer value greater than zero and less than the length of the time series instance to be explained. With data sampled at three-minute intervals over a seven-day period, there are over three thousand possible values for the window size parameter. Through the exploration of larger window sizes corresponding to 6-hour, 12-hour, and 24-hour periods resulting in poor LIMESegment performance, it may be inferred that window sizes smaller than 6-hours are likely to be more optimal for this specific application. Once realistic background generation methods have

been developed, the ideal next step is to optimize the multivariate LIMESegment Robustness score by performing an exhaustive search over possible window sizes. During this initial search, the number of change points per channel should be fixed at a small number. The idea here is to provide a basic sanity check where it is established with certainty that the multivariate LIMESegment adaptation is consistently generating explanations that are robust to small thresholds of added noise. The number of change points per data channel may then be experimentally increased to optimize the Faithfulness metric. Once optimal Robustness and Faithfulness scores have been reached at this step, it may be explored whether having different window size and change point parameters for each data channel improves performance of the multivariate LIMESegment adaptation.

5.3 Other Saliency Methods

If these future works begin to no longer produce positive results, other saliency methods may provide better solutions. Arguably the first method to try is one known as Temporal Saliency Rescaling (TSR) from the work of [IGCBF20] in combination with gradient-based methods. Gradient-based methods work very well in computer vision, although their application to time series classification often fails to account for temporal dependencies in the data and therefore perform poorly. However, TSR is a method for rescaling time series data such that gradient-based methods are far more able to capture these temporal dependencies and generate quality saliency maps. With slight modifications made, the Robustness and Faithfulness calculations may be adapted for consistency of evaluation metrics across interpretability frameworks.

Bibliography

[AN20] Chirag Agarwal and Anh Nguyen. Explaining image classifiers by removing input features using generative models. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[BK59] Richard Bellman and Robert Kalaba. On adaptive control processes. *IRE Transactions on Automatic Control*, 4(2):1–9, 1959.

[GDY+17] Shaghayegh Gharghabi, Yifei Ding, Chin-Chia Michael Yeh, Kaveh Kamgar, Liudmila Ulanova, and Eamonn Keogh. Matrix profile viii: domain agnostic online semantic segmentation at superhuman performance levels. In *2017 IEEE international conference on data mining (ICDM)*, pages 117–126. IEEE, 2017.

[GM21] Damien Garreau and Dina Mardaoui. What does lime really see in images? In *International Conference on Machine Learning*, pages 3620–3629. PMLR, 2021.

[GMRT19] Maël Guillemé, Véronique Masson, Laurence Rozé, and Alexandre Termier. Agnostic local explanation for time series classification. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 432–439. IEEE, 2019.

[IGCBF20] Aya Abdelsalam Ismail, Mohamed Gunady, Hector Corrada Bravo, and Soheil Feizi. Benchmarking deep learning interpretability in time series predictions. *Advances in neural information processing systems*, 33:6441–6452, 2020.

[Ign20] Alexey Ignatiev. Towards trustable explainable ai. In *IJCAI*, pages 5154–5158, 2020.

[KMD19] Fazle Karim, Somshubra Majumdar, and Houshang Darabi. Insights into lstm fully convolutional networks for time series classification. *IEEE Access*, 7:67718–67725, 2019.

[LZ21] Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

[LZC20] Jokin Labaien, Ekhi Zugasti, and Xabier De Carlos. Contrastive explanations for a deep learning model on time-series data. In *International Conference on Big Data Analytics and Knowledge Discovery*, pages 235–244. Springer, 2020.

[MDP+17] Lauren Milechin, Shakti Davis, Tejash Patel, Mark Hernandez, Greg Ciccarelli, Steven Schwartz, Siddharth Samsi, Lisa Hensley, Arthur Goff, John Trefry, et al. Detecting pathogen exposure during the non-symptomatic incubation period using physiological data. *bioRxiv*, page 218818, 2017.

[NFS+21] Inês Neves, Duarte Folgado, Sara Santos, Marília Barandas, Andrea Campagner, Luca Ronzio, Federico Cabitza, and Hugo Gamboa. Interpretable heartbeat classification using local model-agnostic explanations on ecgs. *Computers in Biology and Medicine*, 133:104393, 2021.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceed-*

ings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pages 1135–1144, 2016.

[RSL+21] Clayton Rooke, Jonathan Smith, Kin Kwan Leung, Maksims Volkovs, and Saba Zuberi. Temporal dependencies in feature importance for time series predictions. *arXiv preprint arXiv:2107.14317*, 2021.

[SC07] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[SF22] Torty Sivill and Peter Flach. Limesegment: Meaningful, realistic time series explanations. In *International Conference on Artificial Intelligence and Statistics*, pages 3418–3433. PMLR, 2022.

[SGK17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017.

[STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[TJC+20] Sana Tonekaboni, Shalmali Joshi, Kieran Campbell, David K Duvenaud, and Anna Goldenberg. What went wrong and when? instance-wise feature importance for time-series black-box models. *Advances in Neural Information Processing Systems*, 33:799–809, 2020.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention

is all you need. *Advances in neural information processing systems*, 30, 2017.

[ZBRS22] Yilun Zhou, Serena Booth, Marco Tulio Ribeiro, and Julie Shah. Do feature attribution methods correctly attribute features? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9623–9633, 2022.

[ZINK18] Yan Zhu, Makoto Imamura, Daniel Nikovski, and Eamonn J Keogh. Time series chains: A novel tool for time series data mining. In *IJCAI*, pages 5414–5418, 2018.