Explaining Deep Classification of Time-Series Data with Learned Prototypes

Alan H. Gee\textsuperscript{\ast,1,2}, Diego Garcia-Olano\textsuperscript{\ast,1}, Joydeep Ghosh\textsuperscript{1}, and David Paydarfar\textsuperscript{2}

1 Electrical and Computer Engineering, The University of Texas at Austin, Austin TX 78712, USA
\{alangee, diegoolano\}@utexas.edu; ghosh@ece.utexas.edu
2 Dell Medical School, The University of Texas at Austin, Austin TX 78712, USA
david.paydarfar@austin.utexas.edu

The emergence of deep learning networks raises a need for algorithms to explain their decisions so that users and domain experts can be confident using algorithmic recommendations for high-risk decisions. In this paper we leverage the information-rich latent space induced by such models to learn data representations or prototypes within such networks to elucidate their internal decision-making process. We introduce a novel application of case-based reasoning using prototypes to understand the decisions leading to the classification of time-series data, specifically investigating electrocardiogram (ECG) waveforms for classification of bradycardia, a slowing of heart rate, in infants. We improve upon existing models by explicitly optimizing for increased prototype diversity which in turn improves model accuracy by learning regions of the latent space that highlight features for distinguishing classes. We evaluate the hyperparameter space of our model to show robustness in diversity prototype generation and additionally, explore the resultant latent space of a deep classification network on ECG waveforms via an interactive tool to visualize the learned prototypical waveforms therein. We show that the prototypes are capable of learning real-world features - in our case-study ECG morphology related to bradycardia - as well as features within sub-classes. Our novel work leverages learned prototypical framework on two dimensional time-series data to produce explainable insights during classification tasks.

\textbf{Keywords:} Explainable Machine Learning; Deep Learning; Prototypes; Time Series; ECG; NICU; Healthcare

1 Introduction

Explaining predictions made by deep learning models is necessary for applications in decision-critical domains, like medicine. To safeguard against risk towards humans, regulations, and now legislation, on autonomous, decision-making algorithms demand the incorporation of explainability in machine learning models. Deep neural networks are especially difficult to interpret and explain because\textsuperscript{\ast} Equal contribution
the outcome of such networks are obtained from complex features derived from multiple layers of nonlinear transformations; a process that makes it difficult for humans to interpret model behavior. Because of the recent surge of deep learning models for use in tackling modern problems [6], there is a pressing need for deep learning models to have mechanisms to describe their prediction-generating process. These mechanisms allow for interpretable and faithful explanations of deep learning results so that high-risk domains can trust and leverage their power.

Case-based reasoning offers insight into the decisions algorithms make by providing previous examples or cases as reasons for its decision making. This method is ideal for problems that are difficult to describe but have historical examples. Prototypes are one example of case-based reasoning. Prototypes are learned representative cases that describe influential regions of data and provide insight into global dynamics; the features of those regions are utilized for classification. Previous work [11] has used prototypes as an in-process training tool to explain what the algorithm learns from the latent space as the algorithm evolves. An important distinction between in-process and post-hoc (or black-box explainable) methods are that the latter do not have access to the model generating predictions. Instead, post-hoc explainability attempts to infer the decision-making process of the black-box model by solely using the inputs and predictions of the original black-box model via a secondary model in order to extract explanations. Training a second model based on a prior model’s outputs increases complexity, and more importantly, does not provide transparency into the original model’s decision making. Thus, explanations can dramatically changed based on the robustness of the post-hoc explainable model chosen [16]. In-process methods, on the other hand, are actually learned representations from the model during training, and thus are truly indicative of how the deep model is making decisions.

Much of the explainability methods in machine learning [15] [1] have largely focused on well-labeled image datasets, like MNIST or CIFAR-10, and well-structured tabular data, like UCI’s census income data set where classes are clearly separable. On data with less clear class separation such methods can misbehave. For example, Li et al. [11] introduce a novel method of learning prototypes for classification of training images in-process with an autoencoder. When this model is applied to the MNIST dataset, the prototypes easily separate in the latent space because the latent data representation is separable and well-structured (Fig. 1). However, when class boundaries and features do not form distinguishable clusters, learned prototypes become archetypes (extreme corner cases) and exist near the convex hull of the latent space (Fig. 9). This phenomenon yields prototypes that are representative of the extreme class types, but may not describe the data composition well in the overlapping class regions.

In this work, we aim to remedy the formation of archetypes by creating diverse prototypes that focus on areas of the latent space where class separation is most difficult and least defined. We propose a new prototype model that explicitly accounts for prototype interaction and encourages the model to learn more diverse prototypes to help better explain classification results. Our model uses
the autoencoder framework of Li et al., and introduces a prototype diversity penalty to prevent prototypes from clustering together and learning the same latent representations. We show the utility of this approach on the task of classification of time-series data, specifically classification of normal and abnormal electrocardiogram (ECG) waveforms of patients in the intensive care unit (ICU).

We explore the ECG waveforms in the same manner a clinician would evaluate a signal: evaluating the morphology of ECG signals with visual cues. We present snippets of raw ECG waveforms from preterm infants, as 2-D images, to a deep learning model for classification of bradycardia, a slowing of heart rate. We demonstrate that our model can find a diverse set of meaningful prototypes that help explain what features the algorithm believes are useful for the classification task. Through the prototype clustering, we can analyze potential physiologic dependencies (i.e. clustering of similar severity) related to the time series data of a preterm infant to gain clinically relevant insights based on physiological phenotypes rather than conventional practices of threshold-crossing alarms [3]. To the best of our knowledge this is the first application of prototypes and latent space analysis for health time-series data to map waveform phenotypes.

2 Background

Time-series classification on 1-D data with deep neural networks is a rapidly growing field, with almost 9,000 deep learning models largely focused on recurrent neural networks (RNNs) and convolutional neural networks (CNNs) [6]. However, the number of available healthcare datasets, specifically ECG waveforms, is limited [6]. Within this context, time-series classification on ECG wave-
forms has been done on a small scale, typically with single beat or short-duration arrhythmia classification [5][10][19].

There has been significant effort to use large neural networks to operate on time series represented as images, specifically spectrograms for classification [14], or on raw time-series signals with dilated CNNs [4][9][18]. One such example leverages global average pooling to produce class attention maps (CAMs) to provide explainability for a deep CNN to classify atrial fibrillation in ECG data [9]. CAMs provide probability maps to highlight areas of images that lead to a certain prediction [20], but do not give examples of prototypical examples of the data or explanations of how the training data relates to the end result. In fact, most deep time-series methods are missing in-process interpretability that explain exactly what the model believes is important.

Recent work has focused on using prototypes to provide in-process explainability of classification models, either by learning meaningful pixels in the entire image [11] or by applying attention through the use of sub-regions or patches over an image [2]. We focus on the former work [11] for example-based explainability where the generation of prototypes are intended to look like global representations of the training data. We extend prototypes to consider real-world time-series data and to promote diversity in representation of data where the class boundaries highly overlap.

![Prototype Architecture from Li et al.](image)

**Fig. 2.** Prototype Architecture from Li et al. [11]

## 3 Methods

### 3.1 Time-Series Classification and Explanation via Prototypes

We adopt the autoencoder-prototype architecture from [11]. Let $\mathcal{X} = (x_i, y_i)_{i=1}^n$ be the training set with $x_i \in \mathbb{R}^p$ and class labels $y_i \in \{1, ..., K\}$ for each training point $i \in \{1, ..., n\}$. The front-end autoencoder network learns a lower-dimension
latent representation of the data with an encoder network, \( f : \mathbb{R}^p \rightarrow \mathbb{R}^q \). The latent space is then projected back to the original dimension using a decoder function, \( g : \mathbb{R}^q \rightarrow \mathbb{R}^p \). The latent representation, \( f(x) \), can be passed to a feed-forward prototype network, \( h : \mathbb{R}^q \rightarrow \mathbb{R}^K \), for classification. The prototype network learns \( m \) prototype vectors, \( p_1, p_2, \ldots, p_m \in \mathbb{R}^q \) using a simple three-layer fully-connected network over the latent space. These prototypes are then used to help learn a probability distribution over the class labels \( y_i \) (Fig 2). This in-process prototype classification network in the autoencoder creates a tool that can be used for classification with explainability.

We revise the loss function presented in [11] by adding a penalty for learned prototypes that are too close to one another:

\[
PDL(p_1, \ldots, p_m) = \frac{1}{\log\left(\frac{1}{m} \sum_{j=1}^{m} \min_{k>j} \|p_j - p_k\|_2^2\right)}
\]

We calculate the average minimum squared \( l_2 \) norm between any two prototypes, \( p_i, p_j \). By applying the inverse log to the prototype distances, we penalize prototypes that are close in distance while making sure the minimum distance between prototypes does not get too large. This prototype diversity loss (PDL) promotes prototype diversity and coverage over the latent space. We update the loss function of [11] to:

\[
\mathcal{L}((f, g, h), X) = E(h \circ f, X) + \lambda_R R(g \circ f, X) + \lambda_1 R_1(p_1, \ldots, p_m, X) + \lambda_2 R_2(p_1, \ldots, p_m, X) + \lambda_{pd} PDL(p_1, \ldots, p_m)
\]

where \( E \) is the classification (cross entropy) loss, \( R \) is the reconstruction loss of the autoencoder, and \( R_1 \) and \( R_2 \) are the loss terms that relate the distances of the feature vectors to the prototype vectors in latent space [11]:

\[
R_1(p_1, \ldots, p_m, X) = \frac{1}{m} \sum_{j=1}^{m} \min_{i \in [1, n]} \|p_j - f(x_i)\|_2^2,
\]

\[
R_2(p_1, \ldots, p_m, X) = \frac{1}{n} \sum_{i=1}^{n} \min_{j \in [1, m]} \|f(x_i) - p_j\|_2^2.
\]

The minimization of the \( R_1 \) loss term promotes each prototype vector to learn one of the encoded training examples, while the minimization of \( R_2 \) loss promotes encoded training examples to be close to one of the prototypes. This balance gives meaningful pixel-to-pixel representations in the prototypes.

We train our models for 500 epochs with a batch size of 100. The data are randomly shuffled during each epoch to ensure robust results. We parameterize the number of prototypes and the regularization term \( \lambda_{pd} \) for the classification task of bradycardia severity while keeping the other hyperparameters as in [11].
3.2 Visualization of Latent Space

We take the latent space vectors produced by our encoder and use PCA to reduce the vectors down to a dimension of 500, which retains 98% of the variability of the original latent space vectors. We then calculate the cosine similarity between these 500 dimensional vectors to produce a similarity matrix and use t-distributed stochastic neighbor embedding (t-SNE) from [17] to reduce the 500 x 500 similarity matrix down to three dimensions for visualization purposes. This technique calculates the KL-Divergence between the higher-order dimensional latent space and the lower dimensional space used to represent the former visually. This approach is non-deterministic so the global position in the lower space is uninformative and instead proximity to neighbors is the key insight to gain. Additionally while the first two dimensions of the projection show the general spread of information, the second and third dimensions maybe useful for visualizing within group information.

3.3 Datasets

The neonatal intensive care unit (NICU) dataset used here is composed of two sources: (1) ECG waveforms from the PICS database on PhysioNet [7][8]; and (2) ECG waveforms (500 Hz, Intellivue MP450) from the entire stay of a preterm infant at Seton Medical Center Austin. The study protocol was approved by The University of Texas at Austin Institutional Review Board for human subjects.

**Fig. 3.** Example extraction of bradycardia using Morlet Wavelet transformation (right). The severe bradycardia (55 bpm) starts at the green marker (left). The R peaks were extracted by evaluating the highest scale-average power in the frequency of QRS complex fluctuation. The heart rate (bpm) over time is displayed (left).

The R-R intervals for the ECG of the NICU dataset were extracted using a peak extraction method on a Morlet wavelet transformation of the ECG signal. A peak open-source peak finder was applied to the wavelet scale range (0.01 to .04 scales) related to QRS complex formation in the spectrogram (Fig. 3). The ECG waveforms were clipped at 15 seconds with the event in the middle. All segments were band-passed filtered from 3 to 45 GHz, scaled to zero-mean,
unit-variance, and scaled to the median QRS complex amplitude. Images were then captured to mimic what a clinician would see upon investigation of an ECG signal. Waveforms with no visibly distinguishable QRS complexes were thrown out, since these waveforms would be too obscure for even a clinician to evaluate.

![Fig. 4. Examples of ECG segments in the 3-class classification task of bradycardia.](image)

The following clinical thresholds were used: normal heart rates as >100 beats per minute (bpm), and clinical bradycardias as mild (100-80 bpm), moderate (80-60 bpm), and severe (<60 bpm) (Fig 4). The data has the following class breakdown: \( X_{NICU} = \{ \text{normal} : 1039, \text{mild} : 634, \text{moderate} : 306, \text{severe} : 132 \} \), \( X_{\text{patient1}} = \{ \text{normal} : 713, \text{mild} : 490, \text{moderate} : 172, \text{severe} : 22 \} \). We additionally consider the case of combining moderate and severe classes into one class, changing the original 4 class formulation into a 3 class task, since these two classes both reduce cerebral blood velocity by at least 40%.

### 3.4 Prototype Diversity Score

We adopt a version of the group fairness metric presented in [12]. We refer to this metric as the prototype diversity score, \( \Psi_D \), which is defined as:

\[
\Psi_D = \frac{1}{t} \sum_{i=1}^{t} \sqrt{|\phi_i|} \tag{5}
\]

where \( \phi_i \) is the set of nearest neighbors for prototype \( i \in \{1, ..., t\} \). High scores will occur when prototypes have more unique nearest neighbors. For example, consider \( t = 3 \) prototypes. If all the prototypes have unique nearest neighbors, we get \( \Psi_{D1} = \frac{1}{3}(\sqrt{1} + \sqrt{1} + \sqrt{1}) = 1 \). If two of the prototypes share the same nearest neighbor, we get \( \Psi_{D2} = \frac{1}{3}(\sqrt{2} + \sqrt{1}) = 0.80 \). Thus, \( \Psi_{D1} = 1 > 0.80 = \Psi_{D2} \), and the set with the highest diversity score is the set where the prototypes have the most unique nearest neighbors. Note, \( \max(\Psi_D) = 1 \).

### 4 Results

We alter the loss function described by Li et al. [11] by penalizing learned prototypes that are close in \( l_2 \)-norm distance in the latent space. The new term,
\( \lambda_{pd} \) \( PDL(p_1, ..., p_m) \), in the loss function promotes prototype diversity while improving classification accuracy. We present improved results for learning prototypes across the latent space, then visualize the latent representation of learned prototypes with their nearest-training neighbors, and finally provide insight into the physiology of our time-series data through a case study of the prototypes and latent space representation of the data.

**Fig. 5.** Accuracy results for 3-class NICU bradycardia classification with 10 prototypes: (A) model from Li. et al. [11], and (B) our implementation with added prototype distance loss of \( \lambda_{pd} = 10^3 \). The maximum accuracies are outlined in the legend.

### 4.1 Classification of Waveforms with 2-D Prototypes

We test our prototype implementation with ECG waveforms related to bradycardia using the NICU data for a 3-class and 4-class classification task. We treat the input waveforms as 2-D images and use a four-layer autoencoder to learn complex representations over the data. We find comparable or better test accuracy with our prototype model. With hyperparameter \( \lambda_{pd} = 10^3 \), in the 3-class classification problem, we report a maximum test accuracy of 93.2\%, while the baseline model from [11] achieved a maximum test accuracy of 91.5 \%. For the \( \lambda_{pd} = 10^3 \), 4-class problem, we report a maximum test accuracy of 87.25\%, while the model from [11] achieved a maximum test accuracy of 87.5 \%. Evaluating the normalized confusion matrices for the classification problems (Fig 6), we find that both models perform well on the classification of the normal class, as expected since normal waveforms have near-constant phase. Both models suffer from the difficult task of separating the waveforms in the mild and moderate/severe classes, often confusing the classification between the two classes (Fig 6). This is expected since data near these two class boundaries are difficult to discern, even for domain experts, due to events existing in both classes with possible subtle gap increases. Nonetheless, we find that the addition of a prototype diversity loss performs at least, if not better, than the baseline model.
We note that the 4-class problem is a more difficult problem due to class imbalance resulting from data availability for the 'severe' class. For the analysis going forward, we focus on the 3-class problem where moderate and severe bradycardia are combined into one class.

4.2 Analysis of 2-D Waveform Prototypes

We compare the learned time-series prototypes, one from each class, to their nearest-neighbor training point in the latent space from the last epoch of training (Fig 7). The learned prototypes share similar waveform morphology and cardiac spiking when compared to their nearest neighbors, regardless of class type. Because the prototypes are generated during training, we can see the maturation of these prototypes in (Fig 8) and infer features that the algorithm utilized to classify waveforms at different points during training. At epoch 0, the prototypes are initialized as random noise. At epoch 100 and with high test accuracy, we see that some of the prototypes exhibit global morphological features of the normal waveform class (Prototype 7, Fig 8). As training progresses, we observe other complex phenotypes emerging: prototype 5 learns that the large gaps in cardiac firings are important for identifying the severe cases and prototype 8 learns the consistent pattern of spikes for the mild case. The other prototypes associated with the mild class learn different features of the waveform in different time
Fig. 7. Three of the ten learned prototypes across severity for our updated loss function model. We observe that the learned prototypes have similar waveform morphology as their nearest neighbors (each from different classes). Model details: NICU data, 3-class, 10-prototype, $\lambda_{pd} = 10^3$.

Fig. 8. Prototype evolution with in-process explainability over training time. High level features are easily learned in early epochs of training (Prototype 5 and 7), while more complex features are developed over time. The final nearest neighbors and their corresponding class are depicted on the right. The prototype numbers correspond to the $\lambda_{pd} = 10^3$ latent space cloud in Fig 9. Model details: NICU data, 3-class, 10-prototype.

windows, i.e. prototype 4 learns features in the latter half of the signal while prototype 9 and 10 learn early signal features. Since the mild class shares mixed features of both normal and positive events, it is not surprising that more prototypes are needed in this class to learn subtleties of the class features. Thus, these prototypes highlight important waveform structures that the algorithm deemed
as important when trying to learn the classification of bradycardia. This finding aligns with the idea of clinicians using features present in a bradycardia (i.e. the increasing distance between QRS complexes) to decide whether or not a bradycardia exists in an image.

We compare the latent space of Li et al. [11] to the latent space of our model with prototype diversity loss. Both spaces are projected to two dimensions using T-SNE. The proximity of data points in the 2-D projection suggests that the points are “close” in distance in the original high-dimensional latent space. We represent the learned prototypes by mapping each prototype to its nearest neighbor in the space which itself is an encoded image from the training data (Fig 9). We find that by increasing the scaling of our loss term, PDL, we can increase the local coverage of the prototypes. However, if we regularize our loss term too much (i.e. \( \lambda_{pd} > 10^4 \)), we begin to introduce clustering of prototypes (Table 1). Without the loss term (i.e. \( \lambda_{pd} = 0 \)), we obtain a diversity score of 0.87 and 8 unique nearest neighbors to the prototypes. With the additional prototype distance penalty, we achieve higher diversity scores and classification accuracies for various hyperparameters (Table 1). This suggests that the nearest neighbors to the prototypes are more unique and offer more features that help improve performance in the classification task.

![Fig. 9. Effect of loss regularization on the latent space representation of 3-class, 10-prototype classification results. We compare [11] \( \lambda_{pd} = 0 \) using the their default hyperparameters for the loss function to various regularization penalties \( \lambda_{pd} \) on the prototype distance loss. We depict the second and third dimensions after t-SNE transformation. We note that \( \lambda_{pd} = 10^4 \) slightly suffers from over-regularization; a phenomena which is more evident for higher order \( \lambda_{pd} \) (data not shown).](image-url)
Table 1. Diversity score $\Psi_D$ for various hyperparameters. We report $\Psi_D$ related to the epoch with the highest test accuracy. Our model, $\lambda_{pd} > 0$, returns a better diversity score than the original model, which is $\lambda_{pd} = 0$. Note that there is a set of $\lambda_{pd}$ and prototypes, PT, that maximizes diversity and accuracy.

| $\lambda_{pd}$ | $\Psi_D$ | Distinct NN | Accuracy  |
|----------------|----------|-------------|-----------|
| 0              | 0.87     | 8           | 92.00%    |
| 1              | 0.74     | 6           | 93.75%    |
| $10^2$         | 1.00     | 10          | 94.25%    |
| 500            | 0.94     | 9           | 93.50%    |
| $10^4$         | 0.94     | 9           | 92.25%    |
| $10^4$         | 0.88     | 8           | 93.50%    |
| $10^5$         | 0.88     | 8           | 93.00%    |

| PT  | $\Psi_D$ | Distinct NN | Accuracy  |
|-----|----------|-------------|-----------|
| 6   | 1.00     | 6           | 91.25 %   |
| 10  | 0.94     | 9           | 93.5 %    |
| 15  | 0.88     | 12          | 93.5 %    |
| 25  | 0.65     | 12          | 93.5 %    |
| 50  | 0.49     | 14          | 91.5 %    |

4.3 Prototype Case Study: Exploring ECG Morphology and Bradycardia Classification.

We investigate local clusters of data and the learned prototypes generated by our model as related to physiology. Figure 10 shows a local neighborhood of the 3-class, 10-prototype model with a regularization on the prototype diversity loss of $\lambda_{pd} = 10^4$. With the bottom 3 images, we observe that ECG events in a local neighborhood share similar QRS complex morphology, despite having different class labels and cardiac firing periods. This result suggests that there are physiological dependencies (i.e. clustering based on cardiac morphology and function) that can be learned using our framework to investigate other physiological phenomena, like cardiac ischemia, atrial fibrillation, or even apnea of prematurity in respiration - all of which exhibit visible, abnormal waveform characteristics. To the best of our knowledge this is the first application of prototypes and latent space analysis for health time-series data that could help reveal clinically relevant phenotypes.

We can also examine how the algorithm parses waveforms within a particular class. Even though we did not impose a class constraint, we observe that the algorithm found two separate features within the moderate/severe class that were important in the classification task (i.e. prototypes 2 and 10 shown at the top of the (Fig 10). These two prototypes explore two different cardiac timings as prototype 2 exhibits a progressive delay in cardiac firing, while prototype 10 exhibits a large spontaneous delay. The incorporation of the prototype diversity loss encouraged this exploration of the latent space and prevented prototypes from learning the same feature on the same training point.

4.4 Patient-specific Modeling

We calculate and map the latent space of ECG waveforms which pertain only to one patient’s data, which represents bradycardia events over an entire stay
Deep Prototypical Time-Series Detection

Fig. 10. Learned prototypes showcase the diversity of features that are important for understanding ECG morphology while classifying bradycardia events. Model details: NICU data, 3-class, 10-prototype, $\lambda_{pd} = 10^4$.

Table 2. Varying number of learned prototypes for 3-class setting with fixed $\lambda_{pd}$

| PT | $\Psi_D$ | Distinct NN | Accuracy |
|----|----------|-------------|----------|
| 6  | 1.00     | 6           | 93.0 %   |
| 10 | 0.88     | 8           | 96.0 %   |
| 15 | 0.63     | 7           | 98.5 %   |
| 25 | 0.67     | 12          | 94.5 %   |
| 50 | 0.41     | 10          | 94.5 %   |

Furthermore, we can analytically investigate class pairings by evaluating the weights from the last layer prior to softmax conversion (Fig 12). A smaller weight (underscored in red) indicates a larger contribution from a particular class to a prototype. We observe, like in the full NICU dataset, that the mild examples were important in helping to discern the different classes. This analysis is made
Fig. 11. 3-class, 10-prototype latent space representation of ECG segments with new loss function with $\lambda_{pd} = 500$. This representation is for all bradycardia events of one patient over their entire maturation in the NICU. Of the 10 prototypes, 7 are associated with mild cases, 2 with moderate to severe, and 1 with normal.

possible due to the in-process learning of prototypes, an advantage that post-hoc black-box explainability models lack.

5 Discussion and Future Work

We present a new autoencoder-prototype model that promotes diversity in learned prototypes by penalizing prototypes that are too close in $l_2$-norm distance in the latent space. The new term, $\lambda_{pd} PDL(p_1, ..., p_m)$, in the loss function promotes prototype diversity while improving classification accuracy and prototype coverage of data represented in the latent space. These prototypes help explain which global features of the training data are being used for deep time-series classification. This in-process generation of prototypes offers explainable insights into an algorithm’s metric for classification.

Interestingly, we observe that one prototype is needed to learn and accurately classify regular waveforms - we get at least 98% classification accuracy for normal waveforms (Fig 6). We find that our model allocates more prototypes to learn the intricacies of the more indistinguishable classes (i.e. mild and moderate/severe), which are hard for even a human to discern, especially for the mild cases since they are a mixture of the two other classes. Modifying the complexity of the network to generate more expressive prototypes to better learn the subtle features associated with the positive event classes could improve results.

In preliminary analysis, we extend our model to examine ECG segments prior to a bradycardia event. To determine if there are unique information to learn via prototypes, we investigate 15 second segments prior to the occurrence of an event. These segments have no heart rate below clinical bradycardia levels, i.e. less than 100 bpm. We observe adequate classification accuracies; however, prototype generation was poor. Thus, future work is warranted to fully explore these results and innovate new ways to improve prototype reconstruction.
Despite high classification accuracies, we observe that autoencoder error was a major contributor to the overall loss. Additionally, the high number of loss terms creates a trade-off between prototype interpretability and model accuracy. For example, we observe that for a small number of prototypes, we achieve near-perfect prototype reconstruction but at the cost of classification accuracy. When the number of prototypes was large, we achieve higher accuracy but received noisy prototypes. We believe that a fair solution to this problem is to return the nearest training point to the prototype in the latent representation. In future implementations, we can replace the front-end autoencoder with a model that operates well on 1-D time series, like an RNN or another sequential model, to help balance accuracy and prototype interpretability. Additionally there has been work on computing prototypical patches over 2-D images to generate explainable sub-features [2]. Extending the idea of patches to 1-D time-series signals would allow for parsing the signal for sub-frequencies and features that could better explain how events are triggered. Nonetheless, the work present in this paper provides a more robust prototype model to help explain algorithmic behavior and decision-making in time-series classification tasks.
1. Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad.: Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In KDD. (2015)
2. Chen, C., Li, O., Barnett, A., Su, J., and Rudin, C.: This looks like that: deep learning for interpretable image recognition. arXiv preprint [arXiv:1806.10574] (2018)
3. Chambrin M-C.: Alarms in the intensive care unit: how can the number of false alarms be reduced? Critical Care. 5(184), (2001)
4. Cutler, A. and Breiman, L.: Archetypal analysis. Technometrics, 36(4) (1994)
5. Faust, O., Hagiwara, Y., Hong, T.J., Lih, O.S., Acharya, U.R.: Deep Learning for Healthcare Applications Based on Physiological Signals: a Review. Computer Methods and Programs in Biomedicine (2018)
6. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.A.: Deep learning for time series classification: a review, Data Mining & Knowledge Discovery, (2019)
7. Gee, A.H., Barbieri, R., Paydarfar, D., and Indic, P.: Predicting Bradycardia in Preterm Infants Using Point Process Analysis of Heart Rate. in IEEE Trans. on Biomed. Eng., 64(9)(2017) 2300-2308
8. Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.Ch., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C-K., and Stanley, H.E.: PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23), 215-220 (2000)
9. Goodfellow, S. D., A. Goodwin, R. Greer, P. C. Laussen, M. Mazwi, and D. Eyta.: Towards understanding ECG rhythm classification using convolutional neural networks and attention mappings. Machine Learning for Healthcare (2018)
10. Kiranyaz, S., Ince, T., Gabbouj, M.: Real-time patient-specific ECG classification by 1-d CNNs. IEEE Trans. Biomed. Eng., 63(3) (2016) 664-75.
11. Li, O., Liu, H., Chen, C., and Rudin, C.: Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions In AAAI (2018)
12. Mehrotra, R., McInerney, J., Bouchard, H., Lalmas, M., and Diaz, F.: Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In CIKM, (2018) 22432251.
13. Perlman, J.M. and Volpe, J. J.: Episodes of apnea and bradycardia in the preterm newborn: impact on cerebral circulation. Pediatrics. 76(3) (1985) 333-8
14. Pons, J., Nieto, O., Prockup, M., Schmidt, E. M., Ehmann, A. F., and Serra, X.: End-to-end learning for music audio tagging at scale. CoRR, (2017).
15. MT Ribeiro, S Singh, C Guestrin.: Why should i trust you?: Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD international (2016)
16. Rudin, C. Please stop explaining black box models for high stakes decisions. arXiv preprint arXiv:1811.10154 (2018).
17. van der Maaten, L., Hinton, G.: Visualizing Data using t-SNE. Journal of Machine Learning Research 9 (2008) 2579-2605.
18. Van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K.: Wavenet: A generative model for raw audio. CoRR, abs/1609.03499 (2016)
19. Yildirim, O., Plawiak, P., Tan, R.S., Acharya, U.R.: Arrhythmia detection using deep convolutional neural network with long duration ECG signals Comput. Biol. Med., 102 (2018) 411-420.
20. Zhou, B., Khorsla, A. L. A., Oliva, and A. Torralba.: Learning Deep Features for Discriminative Localization. In CVPR, (2016).