CAN WE EXPLAIN IT: DETERMINING TIME SERIES DATA INTERPRETABILITY WITH CLASSIFICATION OF NEURAL DISCRETE REPRESENTATION

Hyunseung Chung, Sumin Jo, Edward Choi

Kim Jaechul Graduate School of AI, KAIST, Daejeon, Korea

ABSTRACT

Although there has been continuous growth of explainable AI (XAI) in the field of computer vision and natural language processing, interpretability in the field of time series is yet to gather much attention. However, the importance of interpretability is very high in time series data such as in the high-risk medical domain: Electrocardiogram-based heartbeat classification and arrhythmia detection. The lack of interpretability of state-of-the-art heart disease classification methods hamper the deployment of the models in real-world clinical settings. Moreover, existing perturbation-based and gradient-based XAI methods are feature importance based algorithms, which do not consider the difficulty of explanation depending on the dataset. In this paper, we propose a novel evaluation metric to assess the difficulty of XAI in time-series datasets using Area Under the Receiver Operating Characteristics (AUROC). This evaluation method observes the change in AUROC value depending on the number of important features used, while the unused features are masked. The important features are obtained from various state-of-the-art XAI methods such as Local Interpretable Model-agnostic Explanations (LIME), SHapley Additive exPlanations (SHAP), Permutation Feature Importance (PFI), and Integrated Gradients (IG). To best visualize the representative shape of each class, we first quantize the raw time series signals with Vector Quantized Variational-Autoencoder constructed in non-overlapping CNN layers to retain independent receptive fields. We compare the results using three different generated time-series datasets to show how the dataset difficulties effect our proposed AUROC evaluation metric. We present quantitative and qualitative results among various methods and datasets in our experiments section.

Index Terms— time-series, explainable AI

1. INTRODUCTION

Time-series represent any variable that changes over time. This type of data is important in various fields such as in healthcare [1, 2], finance [3], and audio [4, 5]. The expanding importance of time-series data across many domains promote many AI researchers to utilize this type of data for various downstream tasks such as classification, forecasting, and synthesis. However, most state-of-the-art methods in time-series are not interpretable. Therefore, a barrier exists in deploying these models because users lack trust and the models are prone to unexplainable errors for high-risk tasks. In order to mitigate this problem, previous works attempted to use state-of-the-art explainable AI methods in computer vision and natural language processing domains to enable interpretability in the time-series domain. Most of these methods are model-agnostic, which means the tool can be used in any machine-learning model, and it is not constrained to a specific single model.

There are four widely-used model-agnostic state-of-the-art methods, which will be used to determine the feature importance before calculating the AUROC score of our proposed evaluation method. The first method is Local Interpretable Model-agnostic Explanation (LIME) [6], which perturbs the feature values of each sample and observes the resulting impact on the output. The second method is Kernel SHAP [7], which is a combination between LIME and Shapley values. It is similar to LIME but uses shapley kernels instead of exponential kernels for linear regression weight calculation. The third method is Permutation Feature Importance (PFI) [8] and represents ranking based on the decrease in model score when a single feature value is randomly shuffled. Finally, Integrated Gradients [9] attribute the predictions of a classification model to its input features by computing the gradient of the output with respect to the inputs.

The strongest limitation of previous state-of-the-art XAI methods is that the models assume any given input dataset is explainable. In other words, even if a dataset is highly complex to be interpretable for human users, the models output explanation using perturbation-based or gradient-based methods. These outputs do not provide any insightful or consistent human-understandable explanations. Therefore, it is crucial to measure the interpretability difficulty of a dataset to assess its' possibility of explanation.

We propose an assessment method with AUROC scores and masked features utilizing feature importance rankings provided by the four model-agnostic state-of-the-art XAI methods. The raw time-series data are first quantized to encoding indices with VQ-VAE [10]. The benefit of using neural discrete representations is two fold: First, a small number of patterns are learned depending on the size of the codebook, which leads to a regularization effect on the time-series data that contains high noise. Second, the important discrete features can later be visualized with VQ-VAE decoder as a representative time-series segment of the input dataset. Thus, we incorporate non-overlapping convolutional layers to the VQ-VAE architecture by matching kernel size and stride, which enables the receptive field of the discrete features to be independent to each other.

In this paper, we present a framework to determine the interpretability difficulty of a time-series dataset. This framework contains three phases: In the first phase, raw time-series dataset is quantized and a blackbox classifier is trained using the discrete representations. When training the VQ-VAE, kernel and stride size are set equal to each other. In the second phase, previous model-agnostic state-of-the-art methods are utilized to provide feature importance rankings of the quantized inputs. Finally, in the third phase, our proposed AUROC-based evaluation method and feature importance rankings provided from the second phase outputs AUROC scores, which represents interpretability difficulty scores. We present experiments in three different datasets that represent various interpretability difficulty levels. Also, we compare the performance in the four different state-of-the-art XAI methods.

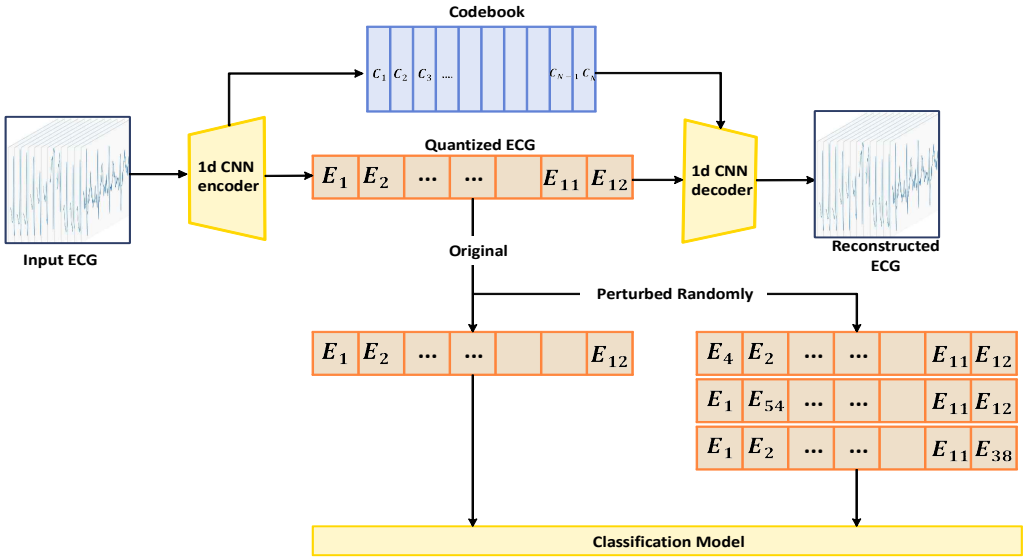


Fig. 1: The overall framework for VQ-VAE, Classification, and perturbation. The VQ-VAE quantizes the input time-series data (ex. ECG) and the quantized time-series tokens are used as input to train the classification model. Perturbations of the time-series tokens are utilized to determine feature importance rankings for area under AUROC calculations.

2. MODEL ARCHITECTURE

Our model architecture consists of VQ-VAE, classification model, and AUROC calculation based on feature masking. Fig.1 represents the overall architecture of our proposed framework. In the intermediate step before the AUROC evaluations, previous model-agnostic state-of-the-art XAI methods are used to determine the important features.

2.1. VQ-VAE

As mentioned earlier, there are two main advantages of using VQ-VAE to quantize raw time-series data: As a regularization effect on the noise, and interpretable visualizer to represent a dataset representative segment. For the first advantage, there has already been work done in various domains to understand and utilize VQ-VAE as regularization to various noise [11, 12]. In the process of quantizing the continous signals into discrete codebooks, unnecessary information such as jitters are reduced. For the second advantage, our main motivation for representing time-series data into discrete tokens was to explore the patterns contained within each token, and visualize them after determining the most important tokens. However, the tokens will be difficult to visualize if the receptive field of each token with respect to the original signal is overlapped by the convolutional layers. Therefore, we use non-overlapping convolutional layers in the VQ-VAE encoder and decoder.

VQ-VAE training process contains three parts, which are encoder, a codebook, and decoder. The encoder E consists of four convolutional layers to downsample the 12-lead ECG signal. The codebook C, also defined as the latent embedding space, consists of code vectors $c_k \in \mathbb{R}^{K\times d}$, where K represents the codebook size and d the dimension of each code vector $c_k, k \in 1, 2, ..., K$. Given a raw ECG signal $x \in \mathbb{R}^{L\times T}$ with L leads and T timesteps, x goes through the encoder to produce output $\hat{l} = E_c(x) \in \mathbb{R}^{T'\times d}$, where T' is the reduced time dimension after downsampling. The output after quantization $E_q(.)$ process is as follows:

$$E_q(\hat{l}) := (\underset{c_k \in C}{\operatorname{argmin}} \|\hat{l}_i - c_k\|_2^2 \text{ for all } i \text{ in } T')$$

Then, the decoder U reconstructs the input $\hat{x} = U(E_q(\hat{l}))$. The entire process is trained with the following loss function:

$$L_{VQ} = \|x - \hat{x}\|_2^2 + \|\text{sg}[E_c(x)] - E_q(\hat{l})\|_2^2 + \|\text{sg}[E_q(\hat{l})] - E_c(x)\|_2^2$$

where sg stands for stop-gradient, which is an identity during the forward pass and zero gradient during the backward propagation.

2.2. Classification Model

The classification model takes in encoding indices trained by VQ-VAE as input and outputs logits for two classes (binary classification). This classification model serves two purposes: First, as a blackbox model when locating important features with the four XAI methods. Second, as an evaluator for AUROC calculations.

LIME, SHAP, PFI, and IG all require a blackbox model for their interpretations. Although they are model-agnostic, the XAI methods still require a model to interpret. In our case we compare three different classification (blackbox) models: Convolutional Neural Network (CNN), Transformer Encoder, and CNN + Transformer models. The advantage of using model-agnostic XAI methods is that various classification models can be used and compared. We display the performance of each classification models in our AUROC evaluation method in the experiments section.

2.3. Model-agnostic XAI Methods

LIME examines the effect of giving variations of data into the black-box model by observing the outcome. The method generated a new dataset that consists of perturbed samples and corresponding predictions. Then, LIME trains a interpretable model (linear regression)

Classifier	Method	AUROC vs. Number of features		
		mitbih	flat	peak
CNN	LIME	0.877	1	0.973
	SHAP	0.921	1	0.997
	IG	0.885	1	0.960
	PFI	0.828	1	0.894
Transformer	LIME	0.873	1	1
	SHAP	0.937	1	1
	IG	0.923	0.9996	0.998
	PFI	0.940	1	1
CNN+Trans.	LIME	0.901	0.998	0.962
	SHAP	0.869	0.998	0.908
	IG	0.891	0.996	0.833
	PFI	0.887	0.998	0.930

Table 1: The table shows area under AUROC values depending on the number of unmasked features using positional rankings determined by four different model-agnostic XAI methods. Three different classifiers are compared.

Dataset	VQVAE reconstruction loss
hard_mitbih	0.00380
flat	0.0231
peak	0.0578

Table 2: Reconstruction loss of VQ-VAE

Classifier	Test accuracy		
	mitbih	flat	peak
CNN	0.930	1	1
Transformer	0.958	1	1
CNN+Trans.	0.971	1	1

Table 3: The test classification accuracy of three different classifier models in three different datasets.

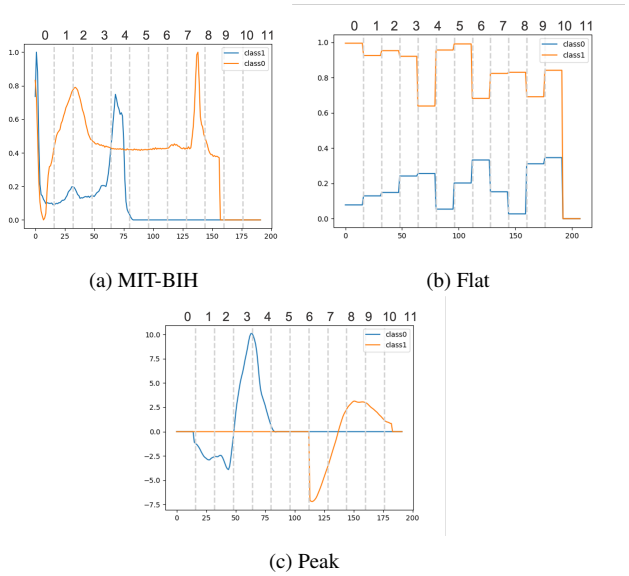
with weights representing the proximity between target sample and generated perturbations. The formula is as shown below:

$$explanation(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \quad (1)$$

where the explanation model for instance x is the linear regression model g. f is the original blackbox model, and $\Omega(g)$ denotes the complexity of the model. G is the family of possible explanations. In LIME, the exponential kernel is used to assign weights to the perturbed samples as follows:

$$K(x, x_i) = \exp\left(\frac{-d(x, x_i)}{\sigma^2}\right) \quad (2)$$

Here, x represents the instance of interest, x_i represents a perturbed sample, and $d(x, x_i)$ represents the distance with σ a parameter that controls the decay of the weights. The Kernel SHAP method is



(a) MIT-BIH (b) Flat

(c) Peak

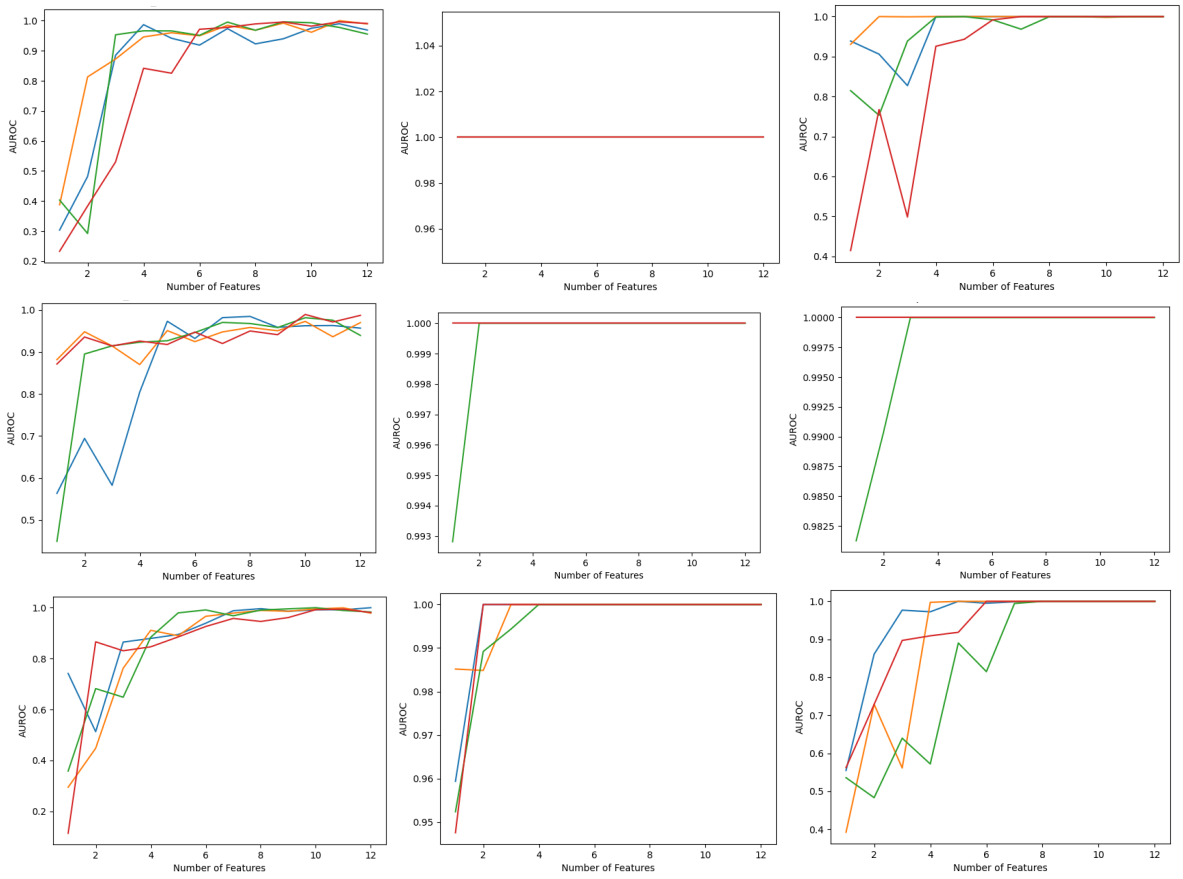
Fig. 2: An example of a sample in each of the three datasets.

same as the LIME method except shapley kernels utilized for weights instead of exponential kernels.

Integrated Gradients interprets blackbox models by assigning importance scores to input features. It constructs a path from a baseline input to the actual input, computing gradients of the model's output with respect to the input features along this path, and integrating these gradients to determine feature importance. The resulting importance scores highlight the features that significantly influence the model's prediction for a specific instance. Finally, the Permutated Feature Importance (PFI) randomly permutes the values of a single feature while keeping the rest of the features unchanged, and then measures the resulting decrease in the model's performance. The larger the decrease in performance, the more important the permuted feature is considered to be.

2.4. AUROC Evaluation Method

Our AUROC method first determines the feature importance of encoded codebook tokens. The encoded indices are used as inputs to the classifier model, and are encoded by the pre-trained VQ-VAE model. The feature importance ranking of the encoded codebook tokens represent the order in which the features will be unmasked to determine the AUROC value. For example, if the feature importance output from LIME using CNN is $[5, 1, 4, 2...]$, then the 5th position of the test sample quantized by VQ-VAE will be unmasked and the rest of the positions will be masked. In the next step, feature positions 5 and 1 will be unmasked to determine the AUROC value. These steps are taken iteratively until the last position of the quantized tokens is unmasked, in which a plot is displayed with y-axis representing AUROC values and the x-axis representing the number of unmasked features used. We compare the area under these curves to determine the interpretability difficulty of various datasets.



(a) MIT-BIH (b) Flat (c) Peak

Fig. 3: AUROC result; First row represents result of CNN classification task, Second row is result of Transformer classifier, and the Last row represents the result of CNN+Transformer classifier. Each column is the result in hard-mitbih, flat, and peak dataset from left to right. Model by line color are as follows. Blue;LIME, Orange;SHAP, Green;IG, Red;PFI

3. EXPERIMENTS

3.1. Experimental Settings

We conduct experiments on two classes of MIT-BIH dataset [13] and generated Flat and Peak datasets. The MIT-BIH dataset is sampled in 360Hz, and contains two classes chosen for their classification difficulty. The dataset represents hard interpretability difficulty, because classification of the two classes is not relatively simple compared to the two generated datasets. We use 400 samples of each class and a total of 800 samples for training, validation, and testing. The Flat dataset represents easy interpretability difficulty, with each of the two classes denoting random points within the range $0 - 0.4$ and $0.6 - 1$ with respect to the y-axis, and each of the 16 timesteps have the same values with respect to the x-axis. The characteristic of this dataset is that all positions can be considered important features because utilizing any one position should enable the model to distinguish between the two classes. We also use 400 samples per class, and a total of 800 samples. Finally, we generate the peak dataset from the ECGtorso dataset [14]. We use two classes and two different regions from the dataset where there is high variance, and zero-out all other regions except those two regions. This dataset also represents easy interpretability difficulty, but is different to Flat dataset in that only a few features are considered important in classification. Similarly, 400 samples are generated for each class. Example of samples in each dataset is shown in Figure 2.

3.2. Training Setup

We trained all models with a batch size of 64. The VQ-VAE and classifier models were optimized using Adam [15] optimizer, and learning rate of 2×10^{-4}. The MIT-BIH dataset samples contained varying time lengths, therefore the samples were zero-padded to the longest existing time length which was 192. Flat and peak datasets were generated to be 192 in time length. After quantization with receptive field of 16, there were 12 quantized tokens representing each sample. we randomly split the datasets into train (80%), validation (10%) and test (10%) sets. All experiments were conducted with 1 **RTX 3090** GPU.

3.3. Quantitative Results

We first conduct VQ-VAE and classification training, and then use four different XAI methods to calculate the AUROC scores. For the VQ-VAE, time-series samples of each datasets are used to train a model for each dataset, independently. Afterwards, the pre-trained VQ-VAE is utilized to quantize the time-series samples to train classi-

fication models based on CNNs, transformers, and CNN + transformers. The VQ-VAE architecture consists of four CNN layers in both encoder and decoder, and the CNN classification model architecture consists of three layers, transformers consists of three transformer encoder layers, and CNN + transformer consists of three CNN layers and one transformer encoder layer. The VQ-VAE loss with respect to each dataset is shown in Table 2, and the classification accuracy with respect to each dataset and model is shown in Table 3. As shown in the tables, the VQ-VAE loss converges well for all datasets and the classification accuracy show perfect scores for Flat and Peak datasets, and slightly lower but almost perfect scores for MIT-BIH dataset.

After determining important positional features with LIME, SHAP, IG, and PFI, we calculate the area under AUROCs with different number of features to determine the interpretability difficulty of datasets. The area under AUROC scores are shown in Table 1. Overall, the values in Flat dataset is higher than both Peak and MIT-BIH datasets because AUROC values reach near perfect score regardless of number of features used. The values in Peak dataset is similar but slightly lower than Flat dataset, and the values of MIT-BIH is the lowest, denoting hardest interpretability difficulty.

3.4. Qualitative Analysis

As visualized in Figure 3, the area under AUROC plots are similar for all three different classification models. For the Flat dataset, the AUROC value is high from start to end because all features are important in the Flat dataset. For the Peak dataset, The AUROC score drastically improves after unmasking more than 2 or 3 features, as expected when a few positions in the dataset contain class discriminative features such as peak points. An exception is the transformer model, where all features are shown to be important. Finally, the MIT-BIH dataset represents difficult interpretability, with the area under AUROC scores less than both Flat and Peak datasets, and the AUROC value reaching top scores after several features unmasked.

4. CONCLUSION

In this work, we propose a novel AUROC-based evaluation method to determine the interpretability difficulty of time-series datasets. The evaluation method successfully represent the difficulty level of each of our three datasets in a consistent manner regardless of the classifier model. For future work we will utilize the VQ-VAE decoder to represent dataset discriminative time-series representations, and improve upon our evaluation method.

5. REFERENCES

[1] Saira Aziz, Sajid Ahmed, and Mohamed-Slim Alouini, "Ecg-based machine-learning algorithms for heartbeat classification," *Scientific reports*, vol. 11, no. 1, pp. 18738, 2021.

[2] Shruti Kaushik, Abhinav Choudhury, Pankaj Kumar Sheron, Nataraj Dasgupta, Sayee Natarajan, Larry A Pickett, and Varun Dutt, "Ai in healthcare: time-series forecasting using statistical, neural, and ensemble architectures," *Frontiers in big data*, vol. 3, pp. 4, 2020.

[3] Warren Freeborough and Terence van Zyl, "Investigating explainability methods in recurrent neural network architectures for financial time series data," *Applied Sciences*, vol. 12, no. 3, pp. 1427, 2022.

[4] Hyunseung Chung, Sang-Hoon Lee, and Seong-Whan Lee, "Reinforce-aligner: Reinforcement alignment search for robust end-to-end text-to-speech," *arXiv preprint arXiv:2106.02830*, 2021.

[5] Sang-Hoon Lee, Ji-Hoon Kim, Hyunseung Chung, and Seong-Whan Lee, "Voicemixer: Adversarial voice style mixup," *Advances in Neural Information Processing Systems*, vol. 34, pp. 294–308, 2021.

[6] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, ""why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[7] Scott M Lundberg and Su-In Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[8] Aaron Fisher, Cynthia Rudin, and Francesca Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously.," *J. Mach. Learn. Res.*, vol. 20, no. 177, pp. 1–81, 2019.

[9] Mukund Sundararajan, Ankur Taly, and Qiqi Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 3319–3328.

[10] Aaron Van Den Oord, Oriol Vinyals, et al., "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[11] Aurko Roy, Ashish Vaswani, Niki Parmar, and Arvind Neelakantan, "Towards a better understanding of vector quantized autoencoders," 2018.

[12] Hanwei Wu and Markus Flierl, "Vector quantization-based regularization for autoencoders," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, pp. 6380–6387.

[13] George B Moody and Roger G Mark, "The impact of the mit-bih arrhythmia database," *IEEE engineering in medicine and biology magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[14] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data mining and knowledge discovery*, vol. 31, pp. 606–660, 2017.

[15] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.