Learning Representations from Healthcare Time Series Data for Unsupervised Anomaly Detection

João Pereira
Instituto Superior Técnico
University of Lisbon
Lisbon, Portugal
joao.p.cardoso.pereira@tecnico.ulisboa.pt

Margarida Silveira
Institute for Systems and Robotics, Instituto Superior Técnico
University of Lisbon
Lisbon, Portugal
msilveira@isr.tecnico.ulisboa.pt

***Abstract*—The amount of time series data generated in Healthcare is growing very fast and so is the need for methods that can analyse these data, detect anomalies and provide meaningful insights. However, most of the data available is unlabelled and, therefore, anomaly detection in this scenario has been a great challenge for researchers and practitioners.**
Recently, unsupervised representation learning with deep generative models has been applied to find representations of data, without the need for big labelled datasets. Motivated by their success, we propose an unsupervised framework for anomaly detection in time series data. In our method, both representation learning and anomaly detection are fully unsupervised. In addition, the training data may contain anomalous data. We first learn representations of time series using a Variational Recurrent Autoencoder. Afterwards, based on those representations, we detect anomalous time series using Clustering and the *Wasserstein* distance.
Our results on the publicly available `ECG5000` electrocardiogram dataset show the ability of the proposed approach to detect anomalous heartbeats in a fully unsupervised fashion, while providing structured and expressive data representations. Furthermore, our approach outperforms previous supervised and unsupervised methods on this dataset.

***Index Terms*—Variational Recurrent Autoencoder, Representation Learning, Clustering, Electrocardiogram.**

I. Introduction

Detecting anomalies in time series data is an important problem of interest in applications such as healthcare, energy and cyber-security. Many Anomaly Detection (AD) approaches have been proposed over time [1, 2]. However, most of these approaches are based on supervised machine learning models that require (big) labelled datasets to be trained. In applications like healthcare, labels are often difficult to obtain, while the annotation process is time-consuming and requires domain-knowledge from experts in the field. Hence, the application of supervised models is limited by this constraint.
Furthermore, some previous anomaly detection approaches do not take into account the sequential nature of data by assuming it is independent and identically distributed in time. When dealing with time series data it is crucial to consider the temporal dependencies of the data.
Recently, there is a renewed interest in unsupervised learning, which is more and more foreseen to play an important role in the future of machine learning [3].
In this work, we propose an unsupervised framework for anomaly detection in sequential data, based on representation learning using a Variational Recurrent Autoencoder and anomaly detection in the representation's space via Clustering and the *Wasserstein* distance [4].

This paper is organized as follows. We start by revising Autoencoders, Variational Autoencoders and Recurrent Neural Networks. Then, we present a summary of recent approaches to anomaly detection in time series data. Afterwards, we introduce our proposed representation learning model and detection methodology. Finally, we present and analyse the results obtained with our model in electrocardiogram (ECG) time series.
Our contributions in this work can be summarized as:

- Unsupervised representation learning of time series data through a Variational Recurrent Autoencoder;
- Latent space-based detection using Clustering and the Wasserstein distance.

II. Background

In this section, we revise Autoencoders, Variational Autoencoders and Recurrent Neural Networks, including Long Short-Term Memory Networks.

A. Autoencoder (AE)

Autoencoders [5, 6] are neural networks trained in an unsupervised fashion that aim to reconstruct their input. They consist of two parts: an *encoder* and a *decoder*. The encoder maps input data $\mathbf{x} \in \mathbb{R}^{d_\mathbf{x}}$ to a latent code/representation $\mathbf{z} \in \mathbb{R}^{d_\mathbf{z}}$ and the decoder maps back from latent code to input space.
Training is executed by minimizing a reconstruction loss and, thus, by making the output of the decoder $\hat{\mathbf{x}}$ as close as possible to the original input $\mathbf{x}$.
Very often autoencoders are undercomplete, i.e. their latent code $\mathbf{z}$ has a lower dimensionality than the input space $\mathbf{x}$ and, hence, they are forced to learn compressed representations of the input data. This characteristic makes them suitable for dimensionality reduction (DR) tasks, where they were proven to work much better than other DR techniques, such as Principal Component Analysis [7].

B. Variational Autoencoder

The interest in autoencoders, and unsupervised learning in general, was strongly revived by the introduction of the variational autoencoder (VAE) [8, 9].

The variational autoencoder is a deep generative model that adds a new constraint on the code $\mathbf{z}$ of the autoencoder. The VAE assumes that the latent code $\mathbf{z}$ is a random variable distributed according to a prior distribution $p_\theta(\mathbf{z})$, which is often defined as a standard Normal distribution, $\mathcal{N}(\mathbf{0}, \mathbf{I})$. However, the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$ is intractable for continuous latent variables $\mathbf{z}$. Therefore, variational inference is applied to find a deterministic approximation $q_\phi(\mathbf{z}|\mathbf{x})$ of the intractable true posterior. Hence, the inference problem is tackled by solving an optimization one. The approximate posterior is usually a multivariate Normal distribution, $\mathcal{N}(\boldsymbol{\mu}_\mathbf{z}, \boldsymbol{\Sigma}_\mathbf{z})$, whose parameters are derived using neural networks.

The VAE training objective aims to maximize an evidence lower bound (ELBO) on the training data log-likelihood given by the following equation, where ϕ and θ are the encoder and decoder parameters, respectively.

$$\mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] - \mathcal{D}_{\mathrm{KL}}\big(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})\big) \quad (1)$$

The distribution for the likelihood term is often a multivariate Normal or $\mathrm{Bernoulli}$, depending on the type of data being continuous or discrete, respectively.

The expectation may be approximated using Monte Carlo integration by drawing L samples from the approximate posterior.

$$\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] \approx \frac{1}{L}\sum_{l=1}^{L} \log p_\theta\left(\mathbf{x}|\mathbf{z}_l\right) \quad (2)$$

C. Recurrent Neural Networks

Feed-forward Neural Networks assume data is independent in time. However, when dealing with sequential data such as time series this assumption does not hold and, thus, recurrent neural networks (RNNs) are often applied. Recurrent neural networks are powerful sequence learners designed to model the temporal dependencies of the data by introducing memory into the network. They receive a sequence of input vectors $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T)$ and, at each timestep t, they compute a hidden state $\mathbf{h}_t$. The key aspect about RNNs is a feedback connection that builds a recurrence mechanism. This mechanism decides how the hidden states $\mathbf{h}_t$ are updated. In simple "vanilla" RNNs, the hidden states are updated based on the current input and the hidden state at the previous timestep, $\mathbf{h}_t = f(\mathbf{U}\mathbf{x}_t + \mathbf{W}\mathbf{h}_{t-1})$. The function f is usually a $\tanh$ or $\mathrm{sigmoid}$ and $\mathbf{U}$ and $\mathbf{W}$ are weight matrices shared across timesteps, to be learned. The hidden state $\mathbf{h}_t$ acts like a summary of the sequence of inputs already seen up to timestep t. RNNs can (optionally) produce an output based on the hidden state $\mathbf{h}_t$ at every timestep or just a single output in the last timestep T.

However, when dealing with sequences with long term dependencies, RNNs suffer from the vanishing gradient problem. This happens when the output at timestep t depends on inputs much earlier in time. Long Short-Term Memory (LSTM) networks [10, 11] are a variant of RNN proposed to overcome this limitation.

LSTMs integrate a memory cell and three gates that control the proportion of the current input to include in the memory cell $\mathbf{i}_t$, the proportion of the previous memory cell to forget $\mathbf{f}_t$ and the information to output from the current memory cell, $\mathbf{o}_t$. The updates of the memory at each timestep t are computed as follows:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{h}_{t-1} + \mathbf{U}_i\mathbf{x}_t) \quad (3)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{h}_{t-1} + \mathbf{U}_f\mathbf{x}_t) \quad (4)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{h}_{t-1} + \mathbf{U}_o\mathbf{x}_t) \quad (5)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_c\mathbf{h}_{t-1} + \mathbf{U}_c\mathbf{x}_t) \quad (6)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (7)$$

In the previous equations, $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$, $\mathbf{c}_t$ and $\mathbf{h}_t$ denote the input gate, the forget gate, the output gate, the memory cell, and the hidden state. $\odot$ denotes an element-wise product.The other parameters are weight matrices to be learned, shared between all timesteps.

Despite the success of LSTMs for sequence modeling, they still can not integrate information from future timesteps. To solve this problem, Bidirectional Long Short-Term Memory networks (Bi-LSTMs) [12] were proposed. Bi-LSTMs exploit the input sequence in both directions by means of two LSTMs: one executes a forward pass and the other a backward pass. As a result, two hidden states are produced at each timestep t, one in each direction, $\overrightarrow{\mathbf{h}}_t$ and $\overleftarrow{\mathbf{h}}_t$. Each one of these states acts like a summary of the past and the future. By concatenating both of them, a global hidden state $\mathbf{h}_t$ that represents the whole context around timestep t is obtained.

III. Related Work

The problem of finding sequences (e.g., ECG heartbeats) that do not conform with the normal pattern is often framed as a time series anomaly detection (AD) task, which is a particular instance of a classification problem (two-class). The work on AD has increased significantly over the past few years and has benefited from the progress made in the framework of deep learning (DL). In healthcare applications dealing with time series data in particular, the work on anomaly detection has been mostly based on (supervised) deep neural network models using either recurrent neural networks or convolutional neural networks (CNNs). In this context, Ng *et al.* [13] applied a 34-layer convolutional neural network for classification of ECG time series. Vig *et al.* [14] used long short-term memory networks for anomaly detection in ECG data. Malhotra *et al.* [15] introduced TimeNet, a sequence to sequence autoencoder model for time series feature extraction, and performed classification using a supervised classifier trained on the extracted features. Other works try to mix different neural network models, such as Karim *et al.* [16] that proposed an architecture that integrates both RNNs and CNNs for time series classification. On the unsupervised learning side, the amount of work developed in the framework of anomaly detection in time series

data is less than the one exploiting supervised models and the proposed approaches still do not yield impressive results. However, recently, there has been an increasing interest in adopting unsupervised learning models for anomaly detection, mainly in the framework of representation learning. In this line, Lei *et al.* [17] proposed a representation learning approach that converts time series of possibly unequal lengths to a matrix form while preserving pair-wise similarities between them and apply it to time series clustering and classification tasks. Aytekin *et al.* [18] used a feed-forward autoencoder for extracting representations of images and performed anomaly detection using clustering.

All in all, even though some of the aforementioned works attained state-of-the-art performances, the literature is still very focused on supervised learning models that heavily rely on good labels to be trained.

IV. Proposed Model

In this section, we present our proposed approach that is based on two fundamental steps: representation learning and detection. The main difference between our work and previous approaches is that both representation learning and anomaly detection are performed in an unsupervised fashion.

A. Representation Learning

Consider a dataset $\mathcal{X} = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$ composed of N observed sequences (e.g., time series), where each sequence n has length T, $\mathbf{x}^{(n)} = \left(\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)}, ..., \mathbf{x}_T^{(n)}\right)$, and each datapoint $\mathbf{x}_t^{(n)}$ is a $d_{\mathbf{x}}$-dimensional vector.

The proposed representation learning model is a Variational Recurrent Autoencoder that works as follows.

The model reads an input time series $\mathbf{x}^{(n)}$ with T timesteps. Afterwards, a local denoising criterion [19] is applied by adding noise to the inputs:

$$\tilde{\mathbf{x}} \sim p(\tilde{\mathbf{x}}|\mathbf{x}), \qquad p(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, \boldsymbol{\sigma}_{\mathbf{n}}^2\mathbf{I}) \tag{8}$$

This corruption process, at the input level, forces the model to reconstruct the original input, $\mathbf{x}$, from a corrupted version of it, $\tilde{\mathbf{x}}$.

The encoder is parametrized by a bidirectional long short-term memory network of parameter ϕ that processes the input time series and produces a sequence of hidden states in both directions. The final hidden states of the forward ($\rightarrow$) and the backward ($\leftarrow$) passes generated by the encoder Bi-LSTM are, then, concatenated in a single vector $\mathbf{h}_T^e = \left[\overrightarrow{\mathbf{h}}_T^e; \overleftarrow{\mathbf{h}}_T^e\right]$. This global hidden state $\mathbf{h}_T^e$ is a fixed-length vector representation/summary of the entire sequence $\mathbf{x}$.

Similarly to Park *et al.* [20] we simplified the denoising criterion by modelling the posterior distribution given a corruption distribution around $\mathbf{x}$ with a single Gaussian, $\tilde{q}_\phi(\mathbf{z}|\mathbf{x}) \approx q_\phi(\mathbf{z}|\tilde{\mathbf{x}})$.

The prior distribution over the latent variables, $p_\theta(\mathbf{z})$, is defined as an isotropic multivariate Normal distribution, $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The parameters $\boldsymbol{\mu}_{\mathbf{z}}$ and $\boldsymbol{\Sigma}_{\mathbf{z}}$ of the approximate posterior distribution $\tilde{q}_\phi(\mathbf{z}|\mathbf{x})$ are derived from the final encoder hidden state, $\mathbf{h}_T^e$, using two fully connected layers with Linear and $\mathrm{SoftPlus}$ activations, respectively. The $\mathrm{SoftPlus}$ function is used to ensure that the variance is parametrized as non-negative and activated by a smooth function.

The latent variables $\mathbf{z}$ are sampled from the approximate posterior and computed using the re-parametrization trick as follows,

$$\mathbf{z} = \boldsymbol{\mu}_{\mathbf{z}} + \boldsymbol{\sigma}_{\mathbf{z}} \odot \boldsymbol{\epsilon} \tag{9}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is an auxiliary (external) noise variable and $\odot$ denotes an element-wise product.

The decoder (generative model) is another Bi-LSTM that receives as input a sample $\mathbf{z}$ drawn from the approximate posterior and outputs, at each timestep t, the parameters of the reconstruction of the input variable $\mathbf{x}$. The decoding distribution $p_\theta(\mathbf{x}_t|\mathbf{z})$ is defined as a multivariate Normal with diagonal co-variance matrix, $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}_t}, \boldsymbol{\Sigma}_{\mathbf{x}_t})$.

Both the encoder and the decoder Bi-LSTMs are activated by a $\tanh$ function.

The training objective is to minimize:

$$\begin{aligned}\mathcal{L}(\theta, \phi; \mathbf{x}^{(n)}) = &-\mathbb{E}_{\tilde{q}_\phi(\mathbf{z}^{(n)}|\mathbf{x}^{(n)})}\left[\log p_\theta(\mathbf{x}^{(n)}|\mathbf{z}^{(n)})\right] \\ &+ \lambda_{\mathrm{KL}}\mathcal{D}_{\mathrm{KL}}\big(\tilde{q}_\phi(\mathbf{z}^{(n)}|\mathbf{x}^{(n)})\|p_\theta(\mathbf{z}^{(n)})\big)\end{aligned} \tag{10}$$

We included a weight parameter λ_{KL} in order to adjust the trade-off between the reconstruction term and the KL-divergence term.

The expectation in the training objective is approximated by Monte Carlo integration. The log-likelihood of a particular sequence $\mathbf{x}^{(n)}$ decomposes across timesteps:

$$\log p_\theta\big(\mathbf{x}^{(n)}|\mathbf{z}^{(n)}\big) = \sum_{t=1}^{T} \log p_\theta\big(\mathbf{x}_t^{(n)}|\mathbf{z}^{(n)}\big) \tag{11}$$

Since the prior on the latent variables is defined as an isotropic multivariate Normal distribution, the KL-divergence term in the training objective has a closed form solution, given by equation 12, and does not require estimation.

$$\mathcal{D}_{\mathrm{KL}}\big(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})\big) \approx 1/2\left[\mathrm{tr}(\boldsymbol{\Sigma}_{\mathbf{z}}) - \boldsymbol{\mu}_{\mathbf{z}}^T\boldsymbol{\mu}_{\mathbf{z}} - d_{\mathbf{x}} - \log(|\boldsymbol{\Sigma}_{\mathbf{z}}|)\right] \tag{12}$$

Figure 1 illustrates the proposed representation learning model.

B. Anomaly Detection

In this work, anomaly detection is performed on the representations provided by the Variational Bi-LSTM Autoencoder model. The representation learning model learns to map input data sequences $\mathbf{x}$ with different patterns into different regions of the space and, therefore, it is straightforward to use those representations to distinguish between normal and anomalous samples.

Given a set of latent representations, the goal of anomaly detection is to find out whether a given representation is *normal* or *anomalous*. For this purpose, we consider three different methodologies: detection via Clustering in the $\boldsymbol{\mu}_{\mathbf{z}}$ space $\big(\boldsymbol{\mu}_{\mathbf{z}} = \mathbb{E}\big[q_\phi(\mathbf{z}|\mathbf{x})\big]\big)$, detection using a metric based on the Wasserstein distance and detection using a supervised Support

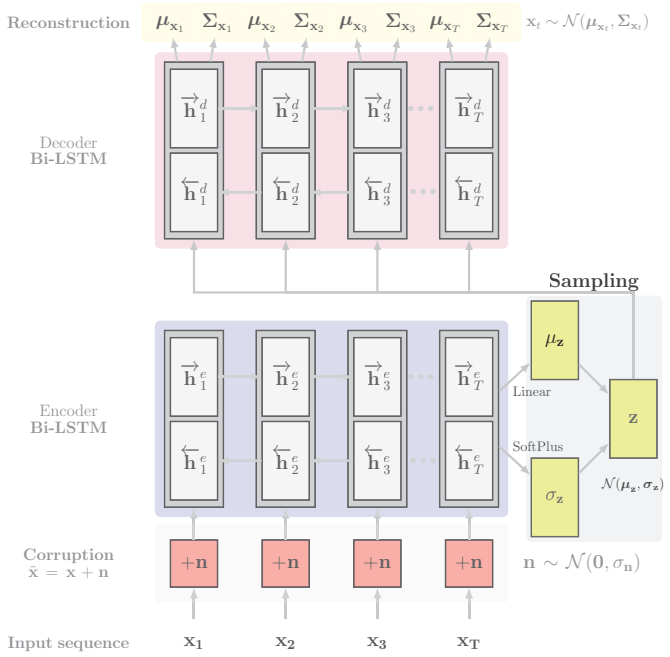

Fig. 1. Illustration of the proposed representation learning model: Variational Bi-LSTM Autoencoder.

Vector Machine (SVM) with linear kernel. The latter is used as a reference to compare the performance of unsupervised *vs* supervised anomaly detection. Note that, in this work, anomaly detection is approached as a two-class classification problem that is not focused on distinguishing between anomalies.

1) Clustering: The detection approach based on clustering consists on applying unsupervised clustering to the latent representations in the approximate posterior mean space ($\boldsymbol{\mu}_{\mathbf{z}}$) and aims to find the clusters that best describe the *normal* and *anomalous* classes of the data. The principle behind this technique in rooted on the assumption that most data used for training the representation learning model are *normal* and, therefore, the representations of anomalous samples will lie in a different region of the latent space. In other words, there will be a cluster containing the predominant (normal) examples and all the others will be represented far from those and assigned to the cluster of anomalous examples.

For this technique we applied three different clustering algorithms in the representations space: hierarchical clustering [21], spectral clustering [22] and k-means++ [23]. The clustering algorithms were set to find 2 clusters, one for each class (normal and anomalous). The output of these algorithms is, then, matched with the normal/anomalous classes by setting the cluster with higher number of data points assigned to be the *normal* one.

2) Wasserstein Distance: Since the model parametrizes either the mean $\boldsymbol{\mu}_{\mathbf{z}}$ and variance $\boldsymbol{\sigma}^2_{\mathbf{z}}$ of the latent variables (approximate posterior parameters), in the framework of anomaly detection, it makes sense to take into account the variability of the latent representations, $\boldsymbol{\sigma}^2_{\mathbf{z}}$, instead of just their expectation, $\boldsymbol{\mu}_{\mathbf{z}}$. This idea is motivated by the fact that even though the representations of normal and anomalous samples in the latent space might share the same mean, $\boldsymbol{\mu}_{\mathbf{z}}$, the variability of anomalous samples relatively to normal ones is likely to be higher, as pointed out by Cho *et al.* [24]. For obtaining an anomaly score, we compute the median *Wasserstein* distance between a test sample $\mathbf{z}^{\text{test}}$ and N_W other samples within the test set of latent representations, so that the similarity between the posterior distribution of a given sample and subset of other samples is used as anomaly score. This methodology works under the assumption often made in anomaly detection problems that most data are normal. The process can be described by equations 13 and 14.

$$W(\mathbf{z}^{\text{test}}, \mathbf{z}^i)^2 = \|\boldsymbol{\mu}_{\mathbf{z}^{\text{test}}} - \boldsymbol{\mu}_{\mathbf{z}^i}\|_2^2 + \|\boldsymbol{\Sigma}^{1/2}_{\mathbf{z}^{\text{test}}} - \boldsymbol{\Sigma}^{1/2}_{\mathbf{z}^i}\|_F^2 \quad (13)$$

$$\text{score}(\mathbf{z}^{\text{test}}) = \text{median}\{W(\mathbf{z}^{\text{test}}, \mathbf{z}^i)^2\}_{i=1}^{N_W} \quad (14)$$

In equations 13 and 14, W denotes the *Wasserstein* distance and the subscript 2 and F denote the ℓ_2-norm and the *Frobenius* norm, respectively.

V. Experiments

A. Data

We applied the proposed model to electrocardiogram (ECG) time series data. The dataset is the `ECG5000`, which was donated by Eamonn Keogh and Yanping Chen and is publicly available in the UCR Time Series Classification archive [25]. This dataset contains a set of $N = 5000$ univariate time series ($d_{\mathbf{x}} = 1$) with 140 timesteps ($T = 140$). Each sequence corresponds to one heartbeat. Five classes are annotated, corresponding to the following labels: *Normal* (N), *R-on-T Premature Ventricular Contraction* (R-on-T PVC), *Premature Ventricular Contraction* (PVC), *Supra-ventricular Premature or Ectopic Beat* (SP or EB) and *Unclassified Beat* (UB). In the original data source, the dataset is provided with a splitting into two subsets: a training set with $N_{\text{train}} = 500$ sequences and a test set with $N_{\text{test}} = 4500$ sequences. Both the training and test sets contain all classes of data, meaning that the training set contains both normal and anomalous data. Moreover, the classes are highly imbalanced: the normal class is the predominant one followed by the class with label R-on-T PVC. For validation purposes, we divided the original training dataset into two subsets - one for training the model ($\mathcal{X}_{\text{train}}$) and one for validation ($\mathcal{X}_{\text{val}}$) - with a splitting ratio of $80/20$, respectively. No further pre-processing was executed. Figure 2 shows the density of each class per set.

B. Training Setup

All the models were implemented using the Keras deep learning library [26], with TensorFlow backend.

Training was performed using *AMS-Grad* [27] optimiser, a variant of *Adam* [28], with a learning rate of 0.001. Gradient computation and weight updates are performed in mini-batches of size 500 during 1500 epochs. We set the latent space dimensionality, $d_{\mathbf{z}}$, to 5, corresponding to an encoding compression ratio of 28. The encoder and the decoder Bi-LSTM both have 256 units in total, 128 in each direction. The noise added at

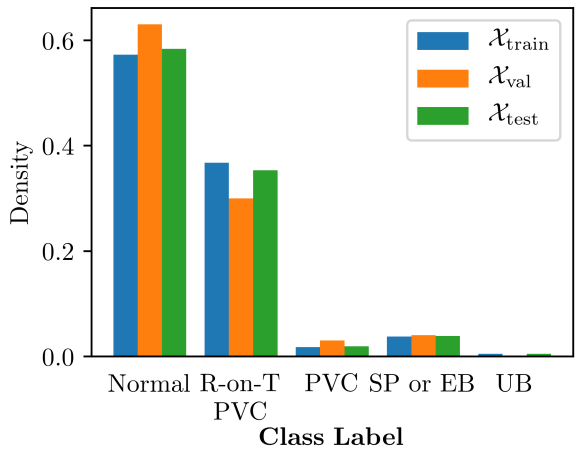

Fig. 2. Class densities per set.

the input level has a standard deviation $\sigma_{\mathbf{n}} = 0.8\sigma_{\mathbf{x}}$. We set the number of Monte Carlo samples L to 1 during training, following the work of Kingma and Welling [8]. To compute the Wasserstein anomaly score we use $N_W = 4000$. To promote stability during training, the gradients were clipped by value with a limit on their magnitude of 5.0. To prevent the KL-divergence term vanishing problem [29], we adopted a KL-annealing strategy in order to vary the weight λ_{KL} of the KL-divergence term in the loss function (equation 10). By doing so, the weight λ_{KL} is initially close to zero - to promote accurate reconstructions of $\mathbf{x}$ in the early stages of training - and gradually increased to encourage smooth encodings and diversity.

Furthermore, we adopted a sparse regularisation criterion to promote sparsity in the hidden layer of the Bi-LSTM encoder [30], by applying a penalty on the ℓ_1-norm of the activations, with a weight parameter of 10^{-7}. The total number of parameters to optimize is 273.420.

Training was executed on a NVIDIA GTX 1080TI graphics processing unit with 11GB of memory, in a machine with an 8^{th} generation i7 processor and 16GB of DDR4 RAM.

VI. Results

In this section, we present the results obtained with the proposed approach. We analyse the representations learned by the model and evaluate the anomaly detection results. All the results reported are evaluated on the test set $\mathcal{X}_{\text{test}}$ composed of 4500 sequences.

A. Latent Space Analysis

Figure 3 shows the latent space of the entire test set ($\mathcal{X}_{\text{test}}$) with 4500 sequences. Each datapoint is labelled with one of the five possible classes annotated. For visualization purposes, we reduced the dimensionality of the latent space from 5 to 2 dimensions using Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbour Embedding (t-SNE) [31]. For the t-SNE embedding, we set the perplexity parameter to 50.0 and the number of iterations to 2000.

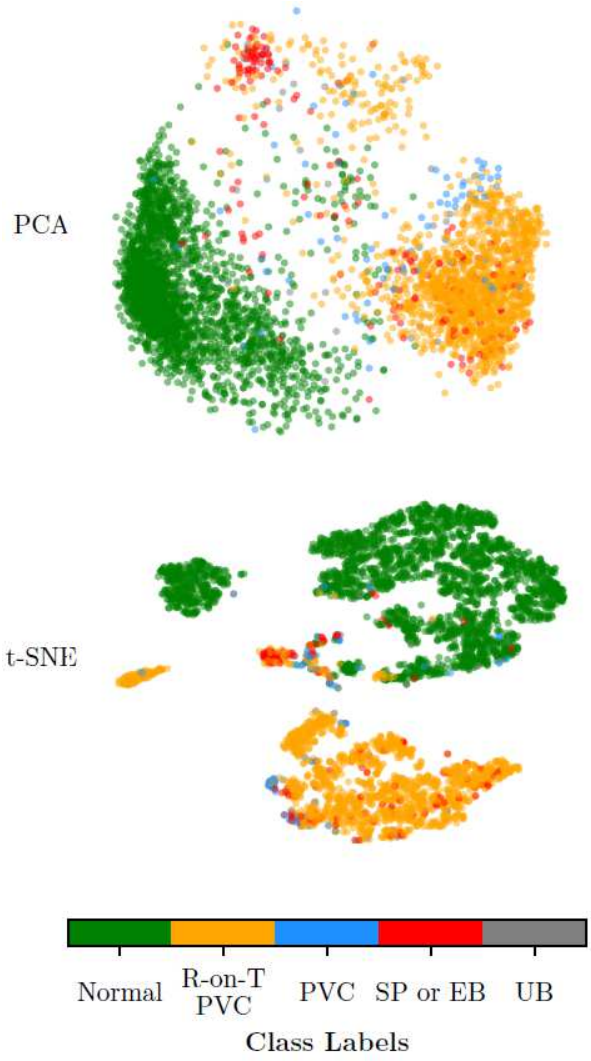

Fig. 3. Latent space visualization of $\mathcal{X}_{\text{test}}$ in 2D via PCA and t-SNE.

Figure 3 reveals a structured and expressive latent space. The sequences (heartbeats) of the *normal* class, represented in green, lie in a region of the latent space different from the *anomalous* ones, while similar heartbeats are mapped onto the same region of the space. Moreover, it is also clear that different anomalies are represented in distinct regions of the space. The anomalous heartbeats in blue and orange, which refer to Premature Ventricular Contractions, are represented close to each other. Interestingly, the anomaly with label "R-on-T PVC", represented in orange, has a smaller cluster apart from the larger one (top of the figure). This might be an interesting result to be analysed by experts.

B. Anomaly Detection

The anomaly detection results are evaluated using Area Under the Curve (AUC), Accuracy, Precision, Recall and F_1-score. These scores are weighted per-class. The process of computing the scores for the different detection methods proposed makes use of the anomaly labels available, but those are employed only for evaluation purposes. Since the output of a clustering algorithm might provide permuted labels, i.e. the cluster assignments may be permuted between the normal and anomalous classes, the assignment can be executed under the assumption that most data are normal, by matching the cluster with higher number of data points with the normal class.

In the methodology based on the *Wasserstein* distance, the AUC is computed by building the receiver operating characteristic (ROC) curve based on the false positive (FP) and true positive (TP) rates obtained for all possible detection thresholds, whereas the other metrics are computed for the detection threshold that leads to the higher scores. For the clustering approach, since it provides a hard classification result rather than an anomaly score, the AUC is computed for a ROC curve with the corresponding TP and FP rate.

In Table I we present the detection results evaluated on the test set, $\mathcal{X}_{\text{test}}$, using different clustering algorithms and a linear SVM. All results reported were averaged over 10 runs of both the representation learning and detection models.

TABLE I
SUMMARY OF THE RESULTS OBTAINED WITH THE PROPOSED MODEL.

Metric	Hierarchical	Spectral	*k*-Means	Wasserstein	SVM
AUC	0.9569	0.9591	0.9591	**0.9819**	0.9836
Accuracy	0.9554	0.9581	**0.9596**	0.9510	0.9843
Precision	**0.9585**	0.9470	0.9544	0.9469	0.9847
Recall	0.9463	0.9516	**0.9538**	0.9465	0.9843
F1-score	0.9465	0.9474	**0.9522**	0.9461	0.9844

The best *unsupervised* anomaly detection scores are emphasized in bold.

The *Wasserstein* distance-based anomaly metric yields the best unsupervised anomaly detection score in terms of AUC. The results obtained for the three clustering algorithms are roughly identical. This fact supports the idea that the key challenge in unsupervised anomaly detection is to learn good (expressive) representations of data. This is the reason why this work is strongly focused on representation learning.

Furthermore, the *Wasserstein* distance-based score outperforms clustering-based detection in terms of AUC and is similar in terms of the other metrics. This result is expected since this score is taking into account the variability of the representations in the latent space, rather than just their mean. The supervised Support Vector Machine performs very well, while the unsupervised detection methods stay roughly competitive. Anyway, all detection strategies attained relatively high detection scores.

Other works have used the same dataset mainly in a supervised multi-class classification framework, instead of anomaly detection that is a two-class problem. Even though both schemes can not be compared in general, since the dataset is highly imbalanced, with a large predominance of the normal and one of the anomalous classes (Figure 2), the multi-class classification problem is almost degenerated in a two-class one. Therefore, it is interesting to compare our method with the results reported in other works that considered different techniques. Table II summarizes the best scores obtained using both supervised and unsupervised learning models in several recent works and the best results for each metric are emphasized in bold.

TABLE II
RESULTS OBTAINED ON THE $ECG5000$ DATASET.

Source	S/U[a]	Model	AUC	Acc	F_1
Ours	S	VRAE+SVM	**0.9836**	**0.9843**	**0.9844**
	U	VRAE+Clust/W	**0.9819**	**0.9596**	**0.9522**
Lei *et al.* [17]	S	SPIRAL-XGB	0.9100	–	–
Karim *et al.* [16]	S	F-t ALSTM-FCN	–	0.9496	–
Malhotra *et al.* [33]	S	SAE-C	–	0.9340	–
Liu *et al.* [34]	U	oFCMdd	–	–	0.8084

[a]Supervised/Unsupervised; $-\equiv$ score not reported in the cited paper.

Most of the previous works that considered the same dataset use supervised machine learning models, while just one follows an unsupervised approach, up to the authors best knowledge. Under the two-class approximation made above, our unsupervised approach outperforms previous supervised learning models in every score reported.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an unsupervised approach to anomaly detection based on representation learning and latent space-based detection. Not only does the proposed representation learning model does not require labels to be trained but also the training data might contain anomalous data. The ratio of anomalous *vs* normal data used for training can be diverse, provided that most data are normal. The method can even deal with ratios of about 40% anomalous data, as was the case in this work. Since it does not depend on the existence of anomaly labels, the proposed approach is suitable for a wide range of applications where time series data are unlabelled, such as healthcare.

Even though the proposed model is generic to be applied to other types of sequential data, both univariate and multivariate, in this work, we focused on healthcare time series data, since it is an important field of application where the methodologies are still very focused on supervised machine learning models.

The results obtained are very encouraging, showing that it is possible to perform anomaly detection when no labels are available. In fact, our fully unsupervised approach attained results that compete with a conventional supervised learning model (the SVM) and outperforms supervised and unsupervised models recently proposed in other works. Nevertheless,

we think much work is still to be done to make unsupervised learning better in anomaly detection.

Finally, in this work, we tackled anomaly detection from the point of view of classifying normal and anomalous data. We plan to extend this framework to the multi-class case, to allow distinguishing between anomalies. We think the representations learned are structured and expressive enough to allow for such a scenario.

ACKNOWLEDGEMENT

This work was funded by FCT project UID/EEA/50009/2019.

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009. [Online]. Available: http://doi.acm.org/10.1145/1541880.1541882

[2] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A Review of Novelty Detection," *Signal Processing*, vol. 99, pp. 215 – 249, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016516841300515X

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 5 2015.

[4] C. Villani, *The Wasserstein distances*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 93–111. [Online]. Available: https://doi.org/10.1007/978-3-540-71050-9_6

[5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Representations by Back-propagating Errors," *Nature*, vol. 323, pp. 533–536, 1986.

[6] H. Bourlard and Y. Kamp, "Auto-Association by Multilayer Perceptrons and Singular Value Decomposition," *Biological Cybernetics*, vol. 59, no. 4, pp. 291–294, Sep 1988. [Online]. Available: https://doi.org/10.1007/BF00332918

[7] G. Hinton and R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504 – 507, 2006.

[8] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *CoRR*, vol. abs/1312.6114, 2013. [Online]. Available: http://arxiv.org/abs/1312.6114

[9] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic Backpropagation and Approximate Inference in Deep Generative Models," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1278–1286. [Online]. Available: http://proceedings.mlr.press/v32/rezende14.html

[10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[11] A. Graves, "Generating Sequences With Recurrent Neural Networks," *CoRR*, vol. abs/1308.0850, 2013.

[12] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition," in *Proceedings of the 15th International Conference on Artificial Neural Networks: Formal Models and Their Applications - Volume Part II*, ser. ICANN'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 799–804. [Online]. Available: http://dl.acm.org/citation.cfm?id=1986079.1986220

[13] A. Y. Ng, P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, and C. Bourn, "Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks," *CoRR*, vol. abs/1707.01836, 2017.

[14] S. Chauhan and L. Vig, "Anomaly Detection in ECG Time Signals via Deep Long Short-term Memory Networks," *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–7, 2015.

[15] P. Malhotra, V. TV, L. Vig, P. Agarwal, and G. Shroff, "TimeNet: Pretrained Deep Recurrent Neural Network for Time Series Classification," *CoRR*, vol. abs/1706.08838, 2017.

[16] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM Fully Convolutional Networks for Time Series Classification," *CoRR*, vol. abs/1709.05206, 2017.

[17] Q. Lei, J. Yi, R. Vaculín, L. Wu, and I. S. Dhillon, "Similarity Preserving Representation Learning for Time Series Analysis," *CoRR*, vol. abs/1702.03584, 2017.

[18] Ç. Aytekin, X. Ni, F. Cricri, and E. Aksu, "Clustering and Unsupervised Anomaly Detection with L2 Normalized Deep Auto-Encoder Representations," *CoRR*, vol. abs/1802.00187, 2018.

[19] Y. Bengio, D. J. Im, S. Ahn, and R. Memisevic, "Denoising Criterion for Variational Auto-Encoding Framework," *CoRR*, vol. abs/1511.06406, 2015.

[20] D. Park, Y. Hoshi, and C. C. Kemp, "A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-based Variational Autoencoder," *CoRR*, vol. abs/1711.00614, 2017.

[21] Y. Zhao, G. Karypis, and U. Fayyad, "Hierarchical Clustering Algorithms for Document Datasets," *Data Mining and Knowledge Discovery*, vol. 10, no. 2, pp. 141–168, Mar 2005. [Online]. Available: https://doi.org/10.1007/s10618-005-0361-3

[22] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, ser. NIPS'01. Cambridge, MA, USA: MIT Press, 2001, pp. 849–856. [Online]. Available: http://dl.acm.org/citation.cfm?id=2980539.2980649

[23] D. Arthur and S. Vassilvitskii, "K-means++: The Advantages of Careful Seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '07. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. [Online]. Available: http://dl.acm.org/citation.cfm?id=1283383.1283494

[24] J. An and S. Cho, "Variational Autoencoder based Anomaly Detection using Reconstruction Probability," *CoRR*, vol. 2015-2, 2015.

[25] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The UCR Time Series Classification Archive," July 2015, www.cs.ucr.edu/~eamonn/time_series_data/.

[26] F. Chollet, "Keras," https://keras.io, 2015.

[27] S. J. Reddi, S. Kale, and S. Kumar, "On the Convergence of Adam and Beyond," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=ryQu7f-RZ

[28] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[29] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio, "Generating Sentences from a Continuous Space," *CoRR*, vol. abs/1511.06349, 2015.

[30] D. Arpit, Y. Zhou, H. Ngo, and V. Govindaraju, "Why Regularized Auto-Encoders learn Sparse Representation?" in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 136–144. [Online]. Available: http://proceedings.mlr.press/v48/arpita16.html

[31] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE ," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. [Online]. Available: http://www.jmlr.org/papers/v9/vandermaaten08a.html

[32] J. Xie, R. B. Girshick, and A. Farhadi, "Unsupervised Deep Embedding for Clustering Analysis," *CoRR*, vol. abs/1511.06335, 2015. [Online]. Available: http://arxiv.org/abs/1511.06335

[33] P. Malhotra, A. Ramakrishnan, G. Anand, and L. Vig, "LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection," *CoRR*, vol. abs/1607.00148, 2016. [Online]. Available: http://arxiv.org/abs/1607.00148

[34] Y. Liu, J. Chen, S. Wu, Z. Liu, and H. Chao, "Incremental Fuzzy C Medoids Clustering of Time Series Data using Dynamic Time Warping Distance," *PLOS ONE*, vol. 13, no. 5, pp. 1–25, 05 2018. [Online]. Available: https://doi.org/10.1371/journal.pone.0197499