Unsupervised Representation Learning and Anomaly Detection in ECG Sequences

João Pereira*

Instituto Superior Técnico,
University of Lisbon,
Lisbon, Portugal
joao.p.cardoso.pereira@tecnico.ulisboa.pt
*corresponding author

Margarida Silveira

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

Abstract: While the big data revolution takes place, large amounts of electronic health records, such as electrocardiograms (ECGs) and vital signs data, have become available. These signals are often recorded as time series of observations and are now easier to obtain. In particular, with the arise of smart devices that can perform ECG, there is the quest for developing novel approaches that allow to monitor these signals efficiently, and quickly detect anomalies. However, since most data generated remains unlabelled, the task of anomaly detection is still very challenging.
Unsupervised representation learning using deep generative models (*e.g.*, variational autoencoders) has been used to learn expressive feature representations of sequences that can make downstream tasks, such as anomaly detection, easier to execute and more accurate.
We propose an approach for unsupervised representation learning of ECG sequences using a variational autoencoder parameterised by recurrent neural networks, and use the learned representations for anomaly detection using multiple detection strategies. We tested our approach on the `ECG5000` electrocardiogram dataset of the UCR time series classification archive. Our results show that the proposed approach is able to learn expressive representations of ECG sequences, and to detect anomalies with scores that outperform other both supervised and unsupervised methods.

Keywords: deep learning; representation learning; data mining; bioinformatics; variational autoencoders; recurrent neural networks; time series; anomaly detection; clustering; healthcare; electrocardiogram; unsupervised learning.

Reference to this paper should be made as follows: Pereira, J. and Silveira, M. (2019) 'Unsupervised Representation Learning and Anomaly Detection in ECG Sequences', *International Journal of Data Mining and Bioinformatics*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: João Pereira received the M.Sc. degree in Electrical and Computer Engineering from Instituto Superior Técnico, University of Lisbon,

Lisbon, Portugal, in 2018. His research interests are in the area of machine learning, with focus on deep learning. He has been working on deep learning applications in the fields of healthcare and energy, and in problems such as representation learning, time series modelling, and anomaly detection.

Margarida Silveira received the E.E. and Ph.D. degrees from the Technical University of Lisbon, Lisbon, Portugal, in 1994 and 2004, respectively. Currently, she is an Assistant Professor with the Electrical Engineering Department, Instituto Superior Técnico, Lisbon, Portugal, and a Researcher at the Institute for Systems and Robotics. Her research interests are in the areas of image processing, computer vision, and pattern recognition.

1 Introduction

In deep learning, learning good feature representations for the downstream problem to be solved is often of great importance (Bengio *et al.*, 2012). By doing so, one can find simpler, useful, and expressive, representations that can make the target task easier to perform.
Finding these representations often requires highly complex and non-linear mappings between the original space of the inputs and the representations space. The success of neural network models has been strongly motivated by their ability to perform these complex mappings.

One such downstream task that has been growing its importance and motivating a significant amount of work over the past few years is anomaly detection (AD). The objective is to find data points that *deviate* from the majority of the data or, in other words, that do not follow the normal behaviour of the data. While many approaches for detecting anomalies in the most various types of data have been proposed over time (Chandola *et al.*, 2009; Pimentel *et al.*, 2014), in application domains such as healthcare, energy, finance, robotics, and security, this problem still remains very challenging, both for researchers and practitioners. The challenges arise from the difficulty of obtaining large labelled datasets that are required to train traditional supervised machine learning models.
Motivated by the lack of labels in the context of applications, there has been in recent years strong efforts on improving unsupervised learning (LeCun *et al.*, 2015). The introduction of novel deep generative models, such as variational autoencoders (VAEs) (Kingma and Welling, 2013; Rezende *et al.*, 2014) and generative adversarial networks (GANs) (Goodfellow *et al.*, 2014), has established a new promising paradigm, specially in the context of representation learning.
In the broad healthcare domain, electrocardiograms (ECGs) are important sequences to be analysed. The collection of these signals is nowadays ubiquitous and easier to perform, with the arise of wearable devices such as smart watches that have the ability of executing one-lead ECG anywhere and anytime. Therefore, with the increase of users, monitoring and classifying these signals is becoming more and more important.

We propose an approach for unsupervised representation learning of ECG sequences, and use the learned representations for the task of anomaly detection. In particular, we train a variational autoencoder model parameterised with recurrent neural networks to first learn the representations of ECG sequences and, then, perform detection in the learned latent space using unsupervised and supervised methods, including clustering, *Wasserstein* similarity (Villani, 2009), and support vector machines.

We first provide background on variational autoencoders and recurrent neural networks. Then, we review related work on representation learning and anomaly detection in time series data (*e.g.*, ECG). Finally, we present our approach for both tasks and evaluate them on a publicly available ECG dataset.

2 Background

In this section, we provide background on variational autoencoders and recurrent neural networks (LSTM and Bi-LSTMs).

2.1 Variational Autoencoders

Autoencoders (Rumelhart *et al.*, 1986; Bourlard and Kamp, 1988) are unsupervised learning models that are trained to reconstruct their input. They consist of two components, an *encoder* and a *decoder*. The encoder maps input data $\mathbf{x} \in \mathbb{R}^{d_\mathbf{x}}$ into a latent representation $\mathbf{z} \in \mathbb{R}^{d_\mathbf{z}}$ and the decoder takes the latent representation and maps back to input space.
The variational autoencoder (Kingma and Welling, 2013; Rezende *et al.*, 2014) is a deep generative model that introduces a probabilistic framework for the conventional 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, $\mathcal{N}(\mathbf{0}, \mathbf{I})$. A sampling process takes place on this prior from which samples of latent codes $\mathbf{z}$ are drawn. Afterwards, a decoder network $p_\theta(\mathbf{x}|\mathbf{z})$, which is a generator model defined by a neural network with parameters θ, outputs a data point $\mathbf{x}$ in the original input space. However, the true posterior $p(\mathbf{z}|\mathbf{x})$ required during maximum likelihood training is generally difficult to compute for a continuous latent space. Instead, the VAE leverages the variational inference technique to find an approximation $q_\phi(\mathbf{z}|\mathbf{x})$ of the true, but intractable, posterior. The approximate posterior is defined by an encoder neural network, also referred to as recognition network, with parameters ϕ, that generates distributions over latent codes. Typically the code distribution is modelled as a multivariate Normal $\mathcal{N}(\boldsymbol{\mu}_\mathbf{z}, \boldsymbol{\Sigma}_\mathbf{z})$, for which the encoder network outputs the corresponding mean and variance.
Figure 1 shows an illustration of a VAE from a graphical model perspective.

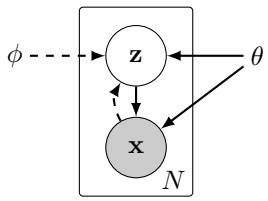


Figure 1 Representation of the VAE as a (directed) graphical model.

The encoder and decoder networks are jointly trained to maximise an evidence lower bound (ELBO) on the log-likelihood of the data, as given by equation 1.

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

where ϕ and θ are the encoder and decoder parameters, respectively. The first term can be seen as a reconstruction term that promotes good reconstructions, whereas the second term

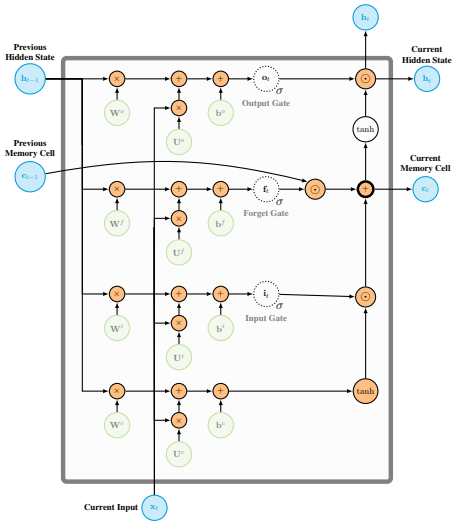
acts as a regularisation penalty on the latent representation $\mathbf{z}$. The later KL divergence term expresses the amount of information that the code $\mathbf{z}$ contains about the input data point $\mathbf{x}$. The decoding distribution is typically a multivariate Normal or Bernoulli, according to the type of data being real-valued or binary, respectively.

2.2 Recurrent Neural Networks

The conventional feed-forward neural networks assume data is independent and identically distributed in time and, hence, they are not specially designed for sequences, such as time series. In order to better model sequences of inputs, recurrent neural networks (RNNs) were proposed. A RNN takes as input a sequence $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T)$ and produces every timestep t a hidden state $\mathbf{h}_t$. A feedback connection is established across timesteps, so that the network keeps an internal memory about the inputs. One of the simplest RNN architectures is the *vanilla* RNN, in which the hidden states are updated in function of the current input and the hidden state at the previous timestep, according to Equation 2.

$$\mathbf{h}_t = f(\mathbf{U}\mathbf{x}_t + \mathbf{W}\mathbf{h}_{t-1}) \tag{2}$$

where the function f is typically a tanh or sigmoid and $\mathbf{U}$ and $\mathbf{W}$ are the weights matrices, shared across timesteps. The hidden states $\mathbf{h}_t$ of a RNN can be interpreted as a vector representation of the sequence of inputs read up to timestep t. While RNNs are more suited for sequential data, their training process is affected by the vanishing gradient problem, which appears specially in sequences with long-term dependencies. In order to tackle this issue, long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997; Graves, 2013), illustrated in Figure 3, were proposed. They have a memory cell and three gates that: control the proportion of the current input to include in the memory cell, the proportion of the previous memory cell to forget, and the information to output from the current memory cell. The computations in a LSTM can be summarised by the following set of equations:



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

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

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

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

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

where $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$, $\mathbf{c}_t$ and $\mathbf{h}_t$ denote the input gate, the forget gate, the output gate, the memory cell, and the hidden state. $\odot$ represents an element-wise product. The other parameters are weight matrices shared across timesteps.

In some applications, it is relevant to exploit the input sequence using both past and future features for a specific timestep. Bidirectional long-short term memory networks (Bi-LSTMs) (Graves *et al.*, 2005) do so by training two LSTMs on the input sequence $\mathbf{x}$. The first LSTM processes the input sequence in the forward direction, while the second LSTM executes a backward pass, producing the corresponding hidden states at each timestep. Then, the resulting hidden state outputs from the forward LSTM, $\overrightarrow{\mathbf{h}}_t$, and backward LSTM, $\overleftarrow{\mathbf{h}}_t$, are merged (*e.g.*, concatenated) at every timestep in order to encode information from past and future contexts, respectively, so that a global hidden state $\mathbf{h}_t$ is obtained.

3 Related Work

The literature on anomaly detection in sequences (*e.g.*, ECG) has been recently leveraging the successes obtained with deep learning models, predominantly in the supervised learning setting. In this framework, neural networks have been able to learn useful feature representations of sequences automatically, which serve as a basis for further tasks, such as anomaly detection. The architectures adopted are typically based on recurrent neural networks or convolutional neural networks (CNNs) that can captures the temporal dependencies of the data. In this context, Ng *et al.* (2017) used a deep convolutional neural network with 34 layers to classify ECG time series. Chauhan and Vig (2015) applied LSTMs for detecting anomalies in sequences and applied it to ECG data. Malhotra *et al.* (2017) proposed a sequence autoencoder model (TimeNet) to extract sequence features automatically, and use them for supervised classification. Other works leverage both neural network architectures, such as Karim *et al.* (2017) that proposed a combination of RNNs and CNNs for time series classification.

In what concerns unsupervised learning approaches, the literature is not so broad as for supervised models. Nevertheless, since the introduction of new deep generative models (*e.g.*, VAEs and GANs), which are unsupervised, there was a renewed interest in unsupervised anomaly detection. An and Cho (2015) proposed a novel detection approach based on a variational autoencoder and applied it to image data and Pereira and Silveira (2018) used a VAE to detect anomalies in solar photovoltaic generation sequences. Li *et al.* (2018) applied generative adversarial networks for anomaly detection in time series data.

Finally, even though the supervised setting continues to provide impressive results, the efforts on the unsupervised side are quickly improving the results and reveal a promising direction of work.

4 Proposed Approach

The proposed approach consists of two tasks. The first one is the task of unsupervised feature representation learning and the second is the downstream task of anomaly detection, for which we want to learn the representations. Unlike previous works, our approach is

unsupervised in both of these steps.
Let $\mathcal{X} = \left\{\mathbf{x}^{(n)}\right\}_{n=1}^{N}$ denote a dataset of N time series with length T and dimension $d_{\mathbf{x}}$:

$$\mathbf{x}^{(n)} = \left(\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)}, ..., \mathbf{x}_T^{(n)}\right)$$

4.1 Unsupervised feature learning by variational autoencoder

The model used for representation learning is based on a variational autoencoder with an encoder/decoder parameterised by recurrent neural networks to tackle the sequential structure of the data. The specific architecture can be described as follows:

4.1.1 Input layer

The input of the model is a time series $\mathbf{x}^{(n)}$. A denoising criterion (Bengio *et al.*, 2015) is, then, applied in order to promote better generalisation of the model to unseen data points. In this criterion, the input is corrupted with noise and the model has to learn how to reconstruct the original clean input $\mathbf{x}$ from its noisy version $\tilde{\mathbf{x}}$. In other words, noise with variance $\boldsymbol{\sigma}_{\mathbf{n}}^2$ is injected at the input level and noisy samples of $\mathbf{x}$ are drawn from a corruption distribution $p(\tilde{\mathbf{x}}|\mathbf{x})$, in this work, a Normal distribution.
The implementation of the denoising criterion was simplified as in the work of Park *et al.* (2017), by modelling the posterior distribution given a corruption distribution around $\mathbf{x}$ with a single Normal, $\tilde{q_\phi}(\mathbf{z}|\mathbf{x}) \approx q_\phi(\mathbf{z}|\tilde{\mathbf{x}})$.

$$\tilde{\mathbf{x}} \sim p(\tilde{\mathbf{x}}|\mathbf{x}) \tag{8}$$

$$p(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, \boldsymbol{\sigma}_{\mathbf{n}}^2\mathbf{I}) \tag{9}$$

4.1.2 Encoder network

The next layer of the model is the encoder, which plays the role of an inference network. The encoder is parameterised by a bidirectional long short-term memory network, with parameters ϕ. This Bi-LSTM reads in all the observations of the input time series $\mathbf{x}^{(n)}$ and generates a sequence of hidden states in both directions (from timestep 1 to T and vice versa). The final hidden states produced in each direction, which are vector representations of the whole sequence, are then concatenated into the vector $\mathbf{h}_T^{\mathrm{e}} = \left[\overrightarrow{\mathbf{h}}_T^{\mathrm{e}}; \overleftarrow{\mathbf{h}}_T^{\mathrm{e}}\right]$.
We defined the prior on the latent variables, $p_\theta(\mathbf{z})$, as an isotropic multivariate Normal distribution, $\mathcal{N}(\mathbf{0}, \mathbf{I})$. From the concatenated vector representation $\mathbf{h}_T^{\mathrm{e}}$ are derived the parameters of the variational approximate posterior $q_\phi(\mathbf{z}|\tilde{\mathbf{x}})$. For the approximate posterior we adopted a multivariate Normal distribution, with diagonal co-variance structure for the sake of simplicity. Therefore, the parameters derived are a mean $\boldsymbol{\mu}_{\mathbf{z}}$ and a variance $\boldsymbol{\sigma}_{\mathbf{z}}^2$, by means of two fully-connected layers with Linear and SoftPlus activation functions.
Afterwards, samples of latent codes are drawn from the variational approximate posterior, using the re-parameterisation trick:

$$\mathbf{z} \sim q_\phi(\mathbf{z}|\tilde{\mathbf{x}}) \tag{10}$$

$$\mathbf{z} = \boldsymbol{\mu}_{\mathbf{z}} + \boldsymbol{\sigma}_{\mathbf{z}} \odot \boldsymbol{\epsilon} \tag{11}$$

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

4.1.3 Decoder network

The decoder network, which is a generative model of time series, is a Bi-LSTM with tanh activation that takes as input a latent code sample from the approximate posterior distribution. The outputs of this decoder are the reconstruction parameters for every observation in the input sequence. The decoding distribution $p_\theta(\mathbf{x}_t|\mathbf{z})$ is a multivariate Normal with diagonal co-variance, $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}_t}, \boldsymbol{\Sigma}_{\mathbf{x}_t})$.

4.1.4 Loss function

The loss function of the proposed model is, then, given by equation 12.

$$\begin{aligned}\mathcal{L}(\theta, \phi; \mathbf{x}^{(n)}) = &- \mathbb{E}_{\tilde{q}_\phi(\mathbf{z}^{(n)}|\mathbf{x}^{(n)})}\left[\log p_\theta(\mathbf{x}^{(n)}|\mathbf{z}^{(n)})\right] \\ &+ \lambda_{\mathrm{KL}} \mathcal{D}_{\mathrm{KL}}\big(\tilde{q}_\phi(\mathbf{z}^{(n)}|\mathbf{x}^{(n)}) \| p_\theta(\mathbf{z}^{(n)})\big)\end{aligned} \tag{12}$$

Instead of adopting the classic variational lower bound objective of the VAE, we added a parameter λ_{KL} that sets the trade-off between the quality of the reconstructions and the simplicity of the representations.
The first term of the loss function is the expectation over the log-likelihood of a Normal distribution, approximated by Monte Carlo integration. The log-likelihood of a time series $\mathbf{x}^{(n)}$ can be decomposed across timesteps according to equation 13.

$$\log p_\theta\big(\mathbf{x}^{(n)}|\mathbf{z}^{(n)}\big) = \sum_{t=1}^{T} \log p_\theta\big(\mathbf{x}_t^{(n)}|\mathbf{z}^{(n)}\big) \tag{13}$$

The second term in the loss is the KL-divergence ($\mathcal{D}_{\mathrm{KL}}$) between the approximate posterior and the prior on the latent codes. In the case of a Normal, this KL term can be computed without estimation with the closed form given by equation 14.

$$\mathcal{D}_{\mathrm{KL}}\big(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z})\big) \approx \frac{1}{2}\big[\mathrm{tr}(\boldsymbol{\Sigma}_\mathbf{z}) - \boldsymbol{\mu}_\mathbf{z}^T \boldsymbol{\mu}_\mathbf{z} - d_\mathbf{z} - \log(|\boldsymbol{\Sigma}_\mathbf{z}|)\big] \tag{14}$$

Figure 2 illustrates a representation of the proposed model.

4.2 Anomaly Detection

We are interested in detecting anomalies in ECG time series data by leveraging the expressive power of the feature representations learned by the model described in section 4.1.
The philosophy behind autoencoder-based anomaly detection consists of training the model on time series with mostly normal patterns, so that it learns a manifold of normal data in its latent space. At test time, *normal* time series are likely to be mapped into a region of the latent space distinct from the time series with anomalous patterns. These representations in the latent space are, thus, useful feature vector representations for our downstream task. Under this strategy, the final end objective of anomaly detection is to identify whether a given representation in the latent space is *normal* or *anomalous*.
We propose three different strategies for detection. Motivated by the quest for unsupervised approaches that are suited to applications where the lack of labels is a key constraint, we propose two unsupervised detection methods. The third methodology is supervised on the anomaly labels and is intended to be used as a baseline for evaluating the unsupervised/supervised frameworks. These strategies are the following:

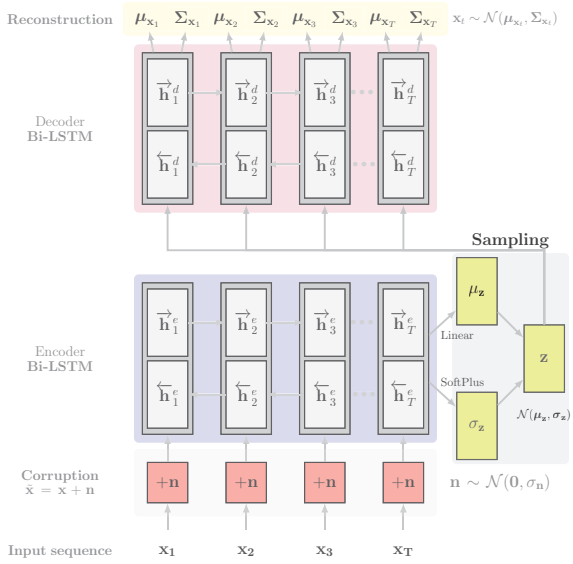


Figure 2 Representation of the proposed model.

- Clustering
- Wasserstein similarity
- Support vector machine (SVM)

4.2.1 Clustering

The first methodology consists on performing clustering over the representations predictive mean space $\left(\boldsymbol{\mu}_{\mathbf{z}} = \mathbb{E}\left[q_\phi(\mathbf{z}|\mathbf{x})\right]\right)$. The objective is to identify the two clusters that contain the latent codes of *normal* and *anomalous* ECG heartbeats. This strategy assumes that the majority of the heartbeats have normal pattern and, therefore, the anomalous heartbeats will be projected differently in the latent space, with respect to the normal ones.
We consider three clustering algorithms: hierarchical clustering (Zhao *et al.*, 2005), spectral clustering (Ng *et al.*, 2001) and k-means++ (Arthur and Vassilvitskii, 2007). The algorithms are set to find two clusters of heartbeat representations, that correspond to the normal and anomalous classes. The clusters are matched to these classes according to their size: the one with higher number of data points is assigned to the *normal* class, and the other one to the anomalous class. Note that this procedure does not require any supervision on the anomaly labels.

4.2.2 Wasserstein similarity

The variational autoencoder introduced a probabilistic framework for the latent code of the conventional autoencoder. The encoder neural network, which parameterises an approximate posterior, derives the expectation and variance of the distribution of latent codes. Since the anomalous heartbeat sequences are likely to have a distinct latent code compared to the normal ones, both the predictive mean and variance of the code can serve as a basis for detection.

We propose to adopt the similarity between the code distribution of a test sample and a representative set of other samples as anomaly score. The intuition behind this strategy is that provided that most of the heartbeats are normal, the anomalous ones will have codes that are far from the normal codes, in terms of similarity of their approximate posterior distributions. Figure 3 illustrates this detection strategy.

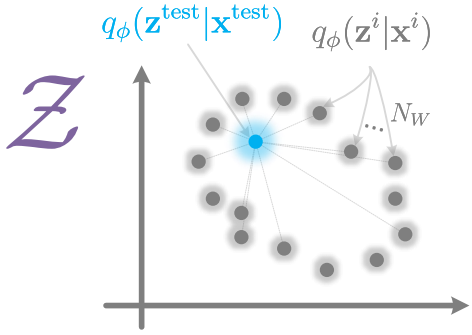


Figure 3 Illustration of the *Wasserstein* similarity-based detection approach.

The anomaly score itself is computed in terms of the median *Wasserstein* similarity between the test code $\mathbf{z}^{\text{test}}$ and a set of N_W other codes, as shown in equations 15 and 16.

$$W(\mathbf{z}^{\text{test}}, \mathbf{z}^i)^2 = \|\boldsymbol{\mu}_{\mathbf{z}^{\text{test}}} - \boldsymbol{\mu}_{\mathbf{z}^i}\|_2^2 + \|\boldsymbol{\Sigma}_{\mathbf{z}^{\text{test}}}^{1/2} - \boldsymbol{\Sigma}_{\mathbf{z}^i}^{1/2}\|_F^2 \tag{15}$$

$$\text{score}(\mathbf{z}^{\text{test}}) = \text{median}\{W(\mathbf{z}^{\text{test}}, \mathbf{z}^i)^2\}_{i=1}^{N_W} \tag{16}$$

where the subscript 2 and F refer to the ℓ_2-norm and the *Frobenius* norm, respectively.

4.2.3 Support vector machine

Support vector machines (Vapnik, 1998) are supervised learning models that try to separate two data classes by finding an optimal hyper-plane that maximises the separating margin. Given a dataset $\mathcal{X} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ with N training examples, where $\mathbf{x}_i \in \mathbb{R}^{d\times}$ is an example in the input space and $\mathbf{y}_i \in \{-1, 1\}$ is the corresponding class label, the optimal hyper-plane can be obtained by solving the optimisation problem formulated in Equation 17.

$$\begin{aligned} &\min_{\mathbf{w},b,\xi_i} J(\mathbf{w},\xi) = \frac{1}{2}\|\mathbf{w}\| + C\sum_{i=1}^{N}\xi_i \\ &\text{subject to} \quad \mathbf{y}_i(\mathbf{w}^T\varphi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, N \end{aligned} \tag{17}$$

where $\mathbf{w}$ and b are a weight vector and a bias, respectively. ξ_i is a slack variable used to allow for data points to be on the wrong side of the hyper-plane, provided that they suffer a penalty C. φ is non-linear a function that maps the original input space into a high-dimensional feature space that may allow a better separation of the classes. In this binary classification formulation, the two data classes correspond to the *normal* and *anomalous* classes.

5 Experiments

5.1 Data

We tested the proposed approach on the `ECG5000` electrocardiogram dataset, released by Eamonn Keogh and Yanping Chen in the UCR Time Series Classification archive (Chen *et al.*, 2015). This dataset contains 5000 one-dimensional sequences ($d_{\mathbf{x}} = 1$) of length 140 and each sequence corresponds to one heartbeat. Figure 4 illustrates a few examples of heartbeats randomly extracted from the dataset.

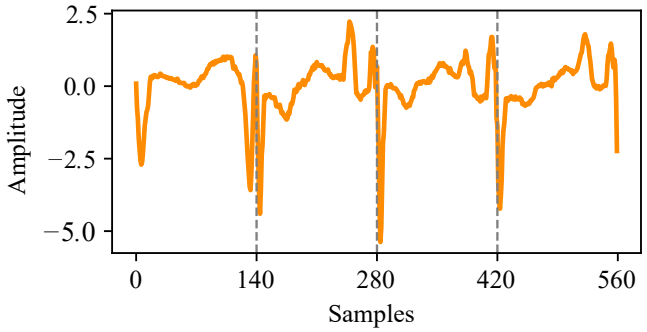


Figure 4 Examples of ECG sequences from the ECG5000 dataset.

The `ECG5000` dataset includes five class labels. One corresponds to normal ECG sequences and the remaining ones are anomalous classes, as described in Table 1.

Table 1 Description of the class labels of the ECG5000 dataset.

Abbreviation	Description
N	Normal
R-on-T PVC	R-on-T Premature Ventricular Contraction
PVC	Premature Ventricular Contraction
SP or EB	Supra-ventricular Premature or Ectopic Beat
UB	Unclassified Beat

The dataset is provided with a division into a training and test sets with size 500 and 4500, respectively. We further split the original training set into two subsets, one for training (80%) and the other for validation (20%). These sets include both normal and anomalous examples and these classes are highly imbalanced as show in Figure 5, being the normal class predominant, followed by the R-on-T premature ventricular contraction anomalous class.

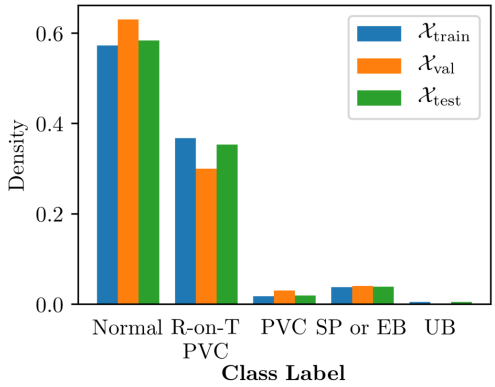


Figure 5 Class densities per set.

5.2 Training framework

The proposed model was implemented using the Keras deep learning library for Python (Chollet, 2015), running on top of TensorFlow (Abadi *et al.*, 2015).
Training was performed using the stochastic gradient descent optimiser *AMS-Grad* (Reddi *et al.*, 2018), in mini-batch mode, with a learning rate of 0.001, during 1500 training epochs. We used a latent space of dimension five, that corresponds to an encoding compression ratio of 28. The encoder and decoder neural networks are Bi-LSTMs with 256 units in total (128 in each direction). The denoising autoencoding criterion is implemented by adding noise to the input sequences with standard deviation $\sigma_{\mathbf{n}}/\sigma_{\mathbf{x}} = 0.8$. The sampling process in the stochastic layer of the variational autoencoder is performed using a single Monte Carlo sample ($L = 1$) during training, in similar setting to the work of Kingma and Welling (2013).
For the computation of the *Wasserstein* similarity anomaly score we used $N_W = 4000$. To promote stability during training, the gradients were clipped by value with a limit on their magnitude of 5. In order to mitigate the KL "collapse" problem in the VAE training, we adopted the strategy proposed by Bowman *et al.* (2015) (KL-annealing) in order to vary the weight λ_{KL} of the KL-divergence term in the loss function (equation 12). In this training scheme, the weight λ_{KL} is first approximately zero in order to generate good reconstructions of the inputs and it is progressively increased to promote smooth encodings and diversity. We further regularised the model by applying a sparsity penalty in the encoder Bi-LSTM (ℓ_1-norm of the activations) (Arpit *et al.*, 2016), with a weight 10^{-7}. The total number of parameters of the model is 273420. The model was trained on a NVIDIA GTX 1080TI GPU, with 11GB of memory, in a machine with an 8^{th} generation i7 processor, and 16GB of DDR4 RAM.

6 Results

This work focuses on representation learning and anomaly detection of ECG sequences. This section analyses the latent space representations produced by the proposed model and

the detection scores obtained on the test set containing 4500 sequences. Finally, an efficiency analysis is performed on the proposed methods.

6.1 Latent Space Representations

To inspect the learned representations we projected the five dimensional latent feature space into a two dimensional space by applying Principal Component Analysis and t-Distributed Stochastic Neighbour Embedding (t-SNE) (van der Maaten and Hinton, 2008) on the test set codes. The embeddings obtained are shown in Figure 6.

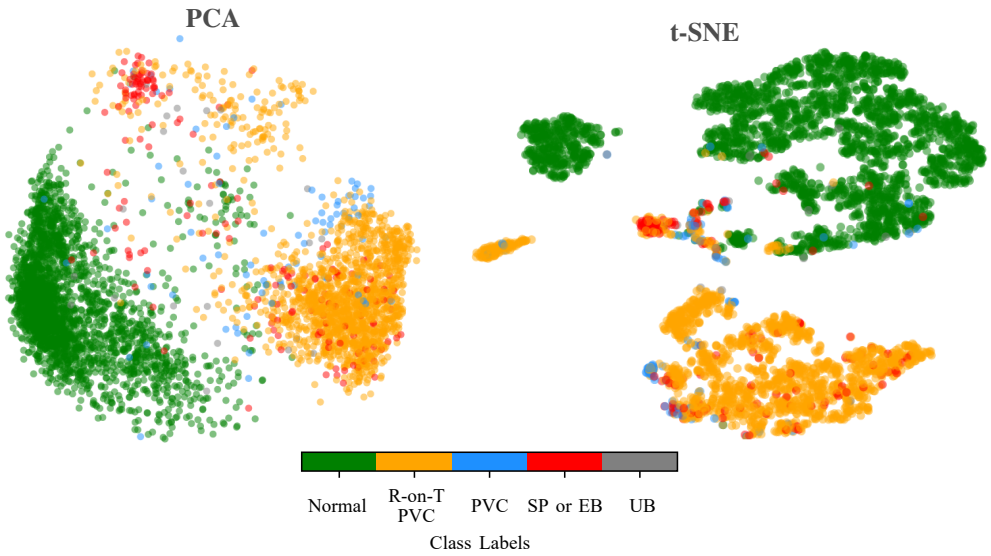


Figure 6 Visualisation of the learned representations for the test set in 2D through PCA and t-SNE. Each point corresponds to a heartbeat sequence and the color is the class label according to Table 1. For the t-SNE dimensionality reduction technique, we chose a perplexity of 50 and 2000 iterations.

Figure 6 shows that the representations of sequences with distinct class labels are spread over different regions of the latent space, meaning that the learned feature representations have captured relevant information about the heartbeat sequences. It is clear a large cluster containing mostly normal sequences (green) and several other smaller ones associated with the other anomalous classes. Such anomalous heartbeats themselves are also mapped into different regions. In addition, it is also possible to visualise two subgroups of representations associated with the R-on-T PVC class (orange).

These results show that the variational autoencoder parameterised by recurrent neural networks is capable of encoding meaningful information about the ECG sequences and their characteristics. More importantly, these characteristics lead to diverse representations of *normal* and *anomalous* heartbeat sequences, which supports the usefulness of these features for further tasks, such as anomaly detection.

6.2 Anomaly Detection

We evaluate the results of the detection task using standard metrics such as accuracy, precision, recall, and F_1-score. We also analyse the results in terms of the receiver operating

characteristic (ROC) curve and the area under the curve (AUC). The scores were weighted according to the class frequency and computed based on the following guidelines:

- **Clustering** —since most data is assumed to be normal, we assign the cluster with higher number of data points to the normal class and the other one to the anomalous class. The AUC is computed for a ROC curve with the corresponding true positive and false positive rates.
- **Wasserstein similarity** —the area under the curve score is computed by, first, finding the false positive and true positive rates for all detection thresholds on the Wasserstein score and, then, drawing the corresponding receiver operating characteristic curve. The remaining metrics are taken considering the detection threshold with best score.
- **SVM** —The SVM was set to output class membership probabilities and the decision threshold varied in order to produce a ROC curve.

The anomaly detection scores obtained on the test set are summarised in Table 2.

Table 2 Anomaly detection results obtained on the test set. We emphasise in bold the best scores produced by *unsupervised* detection methods. All the scores reported were averaged over 10 runs of the proposed approach.

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

For analysing the results it is important to consider if the detection strategy requires supervision on anomaly labels. The clustering strategy is unsupervised and attained similar scores for the three algorithms tested. The method based on the *Wasserstein* similarity achieves generally the same performance as the clustering approach, although it outperforms it in terms of area under the curve. This strategy, in addition to the predictive mean of the code ($\boldsymbol{\mu}_{\mathbf{z}}$), takes into account its variance, what may explain the better result in terms of AUC. Finally, it can be seen that the results of both clustering and Wasserstein similarity are very close to those of SVM, which is a supervised approach that strongly relies on the existence of anomaly labels.
In Figure 7, we represent the receiver operating characteristic curve for all the detection strategies proposed.

The ECG5000 dataset was previously used in other works. However, they are generally focused on the multi-class classification setting, which differs from anomaly detection, that is a two-class problem. Nevertheless, since the dataset is imbalanced, and most samples refer to the normal and one of the anomalous classes, it is still relevant to compare the results with other works that proposed different methodologies. In Table 3 we summarise the best results reported in recent works that applied supervised and unsupervised models.

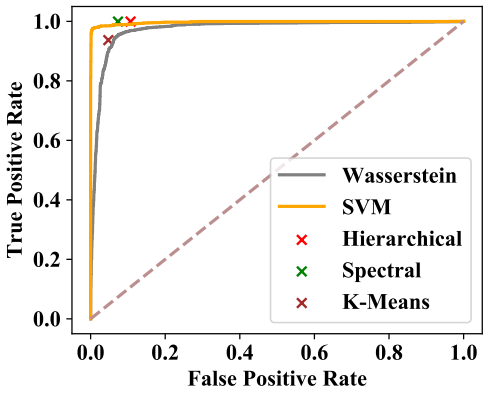


Figure 7 Receiver operating characteristic curve for all detection strategies. The three clustering algorithms are represented with a single point since they provide an output class, whereas the Wasserstein and the SVM ROC curves depend on a detection threshold that produces multiple false positive and true positive rates.

Table 3 Results obtained on the $ECG5000$ dataset.

Source	S/U[1]	Model	AUC	Accuracy	F_1-score
Ours	S	VRAE+SVM	**0.9836**	**0.9843**	**0.9844**
	U	VRAE+Clust/W	**0.9819**	**0.9596**	**0.9522**
Lei *et al.* (2017)	S	SPIRAL-XGB	0.9100	–	–
Karim *et al.* (2017)	S	F-t ALSTM-FCN	–	0.9496	–
Malhotra *et al.* (2016)	S	SAE-C	–	0.9340	–
Liu *et al.* (2018)	U	oFCMdd	–	–	0.8084

[1] S ≡ Supervised, U ≡ Unsupervised.
[2] – score not reported in the cited work.

In these other works the approaches are mostly based on supervised methods, which require anomaly labels, whereas just one follows an unsupervised method, up to the authors best knowledge. Under the aforementioned two-class approximation, our unsupervised approach outperforms the other supervised and unsupervised methods in every score reported.

6.3 Computational Efficiency

We analyse the computational efficiency of our approach in terms of the training, inference and anomaly scores computation times, shown in Table 4.

The model can infer and produce an anomaly score within a few dozens of milliseconds, which is a suitable time period for the purpose of ECG monitoring. The computation of the anomaly score itself is the most expensive step.

Table 4 Computational efficiency of training, inference and anomaly scores computation. We report the higher time for detection (*Wasserstein*, with $N_W = 4000$ representations).

# parameters	# timesteps	# sequences	**Training Time** [ms/seq]	**Inference Time** [ms/seq]	**Anomaly Scores** [ms/seq]
273.420	140	400	2.00	2.42	31.34

7 Conclusions and Future Work

We proposed an unsupervised approach to learn representations of ECG sequences and used the learned representations for the downstream task of anomaly detection. For representation learning we trained a variational autoencoder model with encoder and decoder parameterised with Bi-LSTMs. In addition, we proposed novel methodologies for detecting anomalies in the latent space in an unsupervised way, dismissing anomaly labels that are difficult to obtain. On the other hand, the model can be trained with data containing also anomalous ECG examples and does not requires further preprocessing steps. Also important, from the computational complexity perspective, is that the proposed method is efficient and produces anomaly scores within a few dozens of milliseconds.
The representations learned by the model are structured and can be used for multiple downstream tasks, such as anomaly detection that was the focus of this work. Furthermore, the results obtained with the proposed detection methods are very promising, since unsupervised strategies achieved competitive performance to other supervised methods, in terms of standard detection metrics and receiver operating characteristic.
For future work, we would like to further exploit novel detection strategies that leverage the latent space representations and test the proposed approach on more datasets, characterised by different anomaly ratios. In addition, we would like to apply the proposed methods to multivariate data.

References

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, and et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `http://tensorflow.org/`. Software available from tensorflow.org.

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

D. Arpit, Y. Zhou, H. Ngo, and V. Govindaraju. Why Regularized Auto-Encoders learn Sparse Representation? In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 136–144, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `http://proceedings.mlr.press/v48/arpita16.html`.

D. Arthur and S. Vassilvitskii. K-means++: The Advantages of Careful Seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*,

SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 978-0-898716-24-5. URL http://dl.acm.org/citation.cfm?id=1283383.1283494.

Y. Bengio, A. C. Courville, and P. Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR*, abs/1206.5538, 2012. URL http://arxiv.org/abs/1206.5538.

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

H. Bourlard and Y. Kamp. Auto-Association by Multilayer Perceptrons and Singular Value Decomposition. *Biological Cybernetics*, 59(4):291–294, Sep 1988. ISSN 1432-0770. doi: 10.1007/BF00332918. URL https://doi.org/10.1007/BF00332918.

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

V. Chandola, A. Banerjee, and V. Kumar. Anomaly Detection: A Survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009. ISSN 0360-0300. doi: 10.1145/1541880.1541882. URL http://doi.acm.org/10.1145/1541880.1541882.

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)*, pages 1–7, 2015.

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/.

F. Chollet. Keras. https://keras.io, 2015.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

A. Graves. Generating Sequences With Recurrent Neural Networks. *CoRR*, abs/1308.0850, 2013.

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*, ICANN'05, pages 799–804, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3-540-28755-8, 978-3-540-28755-1. URL http://dl.acm.org/citation.cfm?id=1986079.1986220.

S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

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

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *CoRR*, abs/1312.6114, 2013. URL http://arxiv.org/abs/1312.6114.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 5 2015. ISSN 0028-0836. doi: 10.1038/nature14539.

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

D. Li, D. Chen, J. Goh, and S. Ng. Anomaly detection with generative adversarial networks for multivariate time series. *CoRR*, abs/1809.04758, 2018. URL http://arxiv.org/abs/1809.04758.

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*, 13(5):1–25, 05 2018. doi: 10.1371/journal.pone.0197499. URL https://doi.org/10.1371/journal.pone.0197499.

P. Malhotra, A. Ramakrishnan, G. Anand, and L. Vig. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. *CoRR*, abs/1607.00148, 2016. URL http://arxiv.org/abs/1607.00148.

P. Malhotra, V. TV, L. Vig, P. Agarwal, and G. Shroff. TimeNet: Pre-trained deep recurrent neural network for time series classification. *CoRR*, abs/1706.08838, 2017.

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*, NIPS'01, pages 849–856, Cambridge, MA, USA, 2001. MIT Press. URL http://dl.acm.org/citation.cfm?id=2980539.2980649.

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

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

J. Pereira. Unsupervised Anomaly Detection in Time Series Data Using Deep Learning. Master's thesis, Instituto Superior Técnico, University of Lisbon, 2018.

J. Pereira and M. Silveira. Unsupervised Anomaly Detection in Energy Time Series Data Using Variational Recurrent Autoencoders with Attention. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1275–1282, Dec 2018. doi: 10.1109/ICMLA.2018.00207.

J. Pereira and M. Silveira. Learning representations from healthcare time series data for unsupervised anomaly detection. In *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 1–7, Feb 2019. doi: 10.1109/BIGCOMP.2019.8679157.

M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215 – 249, 2014. ISSN 0165-1684. doi: https://doi.org/10.1016/j.sigpro.2013.12.026. URL http://www.sciencedirect.com/science/article/pii/S016516841300515X.

S. J. Reddi, S. Kale, and S. Kumar. On the Convergence of Adam and Beyond. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ryQu7f-RZ.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR. URL http://proceedings.mlr.press/v32/rezende14.html.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning Representations by Back-propagating Errors. *Nature*, 323:533–536, 1986.

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

V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

C. Villani. *The Wasserstein distances*, pages 93–111. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-540-71050-9. doi: 10.1007/978-3-540-71050-9_6. URL https://doi.org/10.1007/978-3-540-71050-9_6.

J. Xie, R. B. Girshick, and A. Farhadi. Unsupervised Deep Embedding for Clustering Analysis. *CoRR*, abs/1511.06335, 2015. URL http://arxiv.org/abs/1511.06335.

Y. Zhao, G. Karypis, and U. Fayyad. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10(2):141–168, Mar 2005. ISSN 1573-756X. doi: 10.1007/s10618-005-0361-3. URL https://doi.org/10.1007/s10618-005-0361-3.