# UNIVERSITY OF TORINO

M.Sc. in Stochastics and Data Science

Final dissertation


**Irregular heartbeats detection using Deep Generative modelling**

Supervisor: Rossano Schifanella

Candidate: Simone Maggi

ACADEMIC YEAR 2020/2021

# Summary

Can we teach a machine the normal behaviour of the heart and then have it use this knowledge to assess whether something is wrong?
In this thesis, we will face the problem of recognition of abnormal heartbeats inside the ECG signal by using modern data-driven techniques. However, since the abnormal patterns are plentiful, rare and challenging to collect in a balanced dataset while the normal ones are pretty common, we will face this problem in a/an semi-supervised/unsupervised manner. The model will learn the normal heartbeats variability, and then it will use the knowledge acquired to infer the abnormality of new data. We propose AAECG a Deep Generative Model derived by an Adversarial Autoencoder(AAE) to capture the Sinus heartbeat distribution throughout a set of latent variables and additional patient gender information. The model is intended to monitor 24/7 patients in intensive care by alarming doctors only when abnormal heartbeats are detected. Tested on the MIT-BIH arrhythmia database, It reached 0.95 ROC-AUC and 0.92 PR-AUC outperforming the baselines and competing with the state-of-art.
Furthermore, the model shows to understand the sex differences between heartbeats, opening the possibility to study the effects of some conditions, drugs or other particular details on the normal heartbeat wave-form, opening a path towards more patient-specific diagnosis.

# Acknowledgements

Ammetto che è stato un lungo viaggio, pieno di sofferenze, sacrifici e qualche sporadica gioia. Ma più che la soddisfazione di averlo concluso mi rimarrà l'esperienza del viaggio stesso, che forse mi aiuterà a viaggiare meglio in futuro. Ringrazio il prof. Rossano Schifanella che mi ha seguito nel progetto, la famiglia e gli amici e tutti quelli che mi hanno accompagnato lungo il percorso e che continueranno a farlo nei tempi futuri.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Brief introduction to electrocardiography

It is beyond this section's scope to treat this topic extensively, but a brief introduction will help the reader understand future mathematical or algorithmic choices.

### 1.1.1 Sinus Rhythm

The electrocardiogram (ECG) is the electric signal obtained measuring heart activity through electrodes placed on the skin. The signal is composed of a sequence of heartbeats which duration is about 0.8 seconds. Cardiologists use it to spot misbehaviour at the heart level. In particular, they analyze the waves' shape in the signal, the frequency of the beats, and their regularity. The heart's normal behaviour is called in medical terms *Sinus rhythm*, and it is characterized by a regular sequence of sinus heartbeats with a rate between 60 and 100 beats per minute.

A precise sequence of waves composes the Sinus heartbeat. It starts with the so-called P wave generated by the sinoatrial (SA) node during the atrial depolarization. It triggers ventricular depolarization, which appears in the ECG as the biggest wave called the QRS complex. The last phase is ventricular repolarization, which produces the T wave and the U wave.

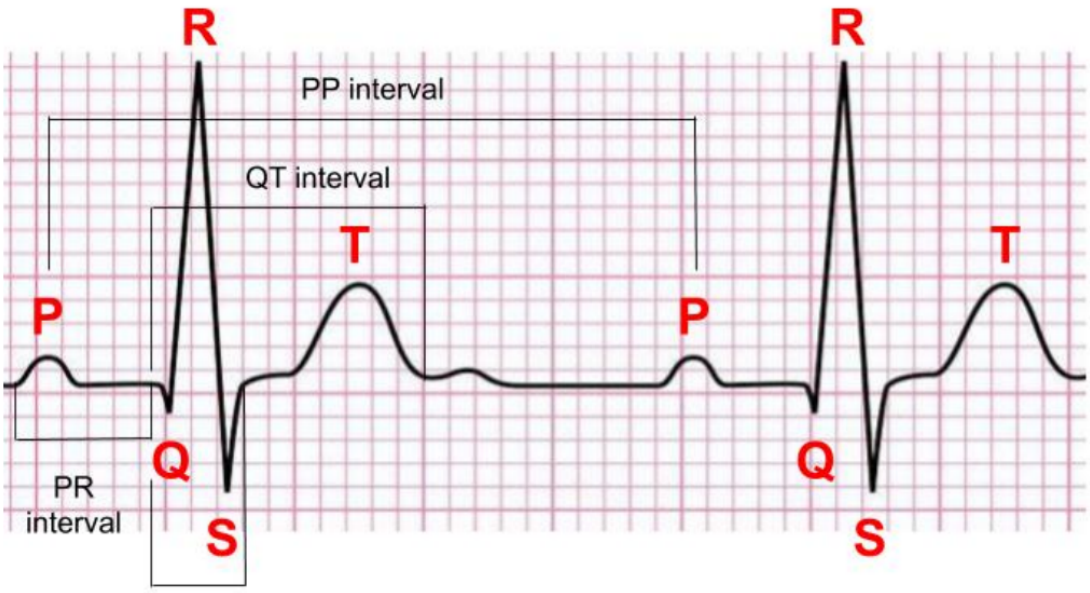



Figure 1.1: Two normal heartbeats and the corresponding notation[79].

### 1.1.2 Abnormal ECG

The ECG abnormalities alter the rate of observed beats and their morphology. When the bpm is too slow, and the heartbeats reflect the sinus behaviour, we face sinus bradycardia. Sinus bradycardia could imply that the heart is pumping too slowly. In the opposite case, we have Sinus tachycardia, with a heart-rate of over 100 bpm. When the hearth-rate appears irregular and fluctuating, the rhythm is called Sinus Arrhythmia. When the heartbeats do not follow the sinusal shape, we observe an irregular rhythm. The ectopic rhythms are the type of heartbeats that do not start from the sinoatrial node. For instance, the Premature Atrial Contraction, Wandering Atrial Pacemaker, Atrial Tachycardia, Atrial Flutter, Atrial Fibrillation are ectopic beats that start from the atrial. Some of them are innocent and happen under emotional stress like premature atrial contraction(PAC). However, when the PAC beat appears consecutively, the rhythm is called Atrial tachycardia, and it requires intervention since it could cause fainting[47]. Sometimes the heartbeat starts from the sinoatrial node but presents an irregular shape in the ST segment or the T wave amplitude. The most harmful case is the acute ST-segment elevation myocardial infarction (STEMI), which announces an incoming stroke and needs immediate hospitalization.

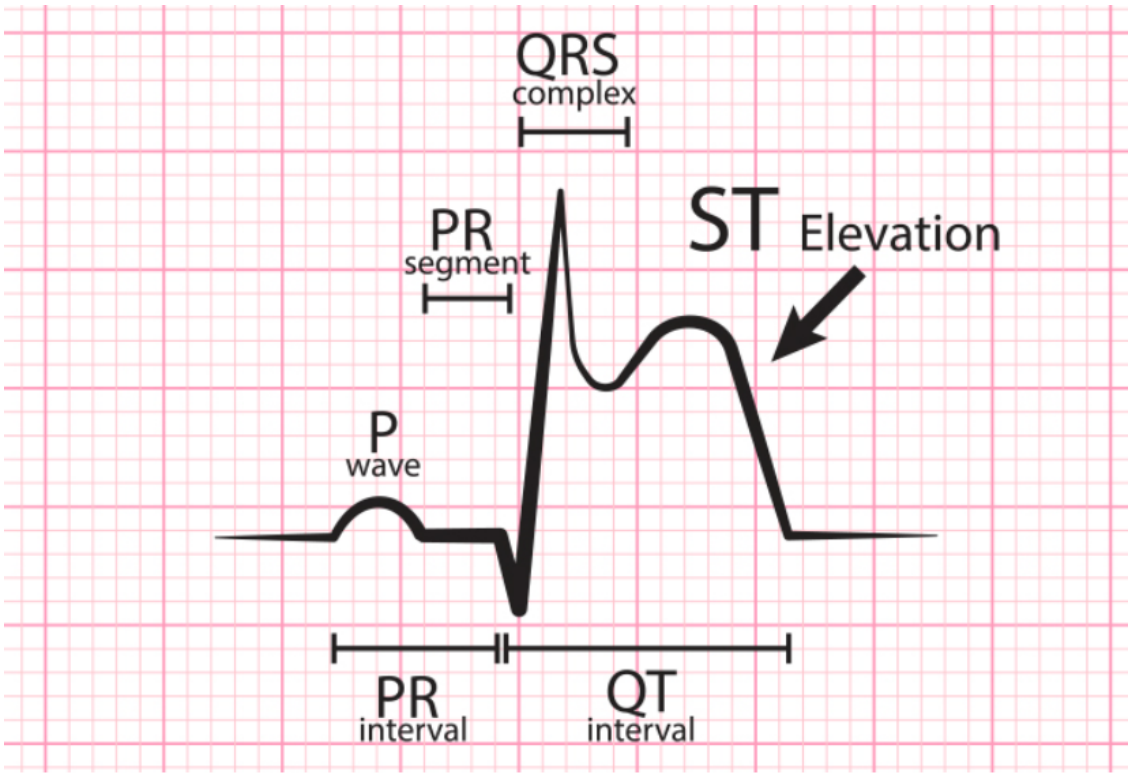



Figure 1.2: Elevetion of ST-segment [58].

### 1.1.3 ECG derivations

The recorded signal depends on the position of the electrodes with respect to the heart. From an electrical point of view, the heart can be regarded as a dipole that rotates during the cardiac cycle, and the two electrodes, positive and negative, register the variation of electrical potential along an axe traced by the positions of the electrodes. The theory behind electrocardiography is rooted in electromagnetic and can be explained in four points:

- depolarization of the heart towards the positive electrode produces a positive deflection
- depolarization of the heart away from the positive electrode produces a negative deflection
- repolarization of the heart towards the positive electrode produces a negative deflection
- repolarization of the heart away from the positive electrode produces a positive deflection.

The above four rules determine the waves in the sinus heartbeat showed in figure 1.1, observed by a specific electrodes configuration. The medical community has established a standard electrodes configuration: the 12-lead ECG. The standard specifies the position of ten electrodes which output twelve distinct leads. Each of them offers a peculiar perspective of the heart electrical activity from a different angle. The 12-lead ECG is considered the

| **Category** | **Leads** | **Activity** |
|---|---|---|
| Inferior leads | Leads II, III and aVF | diaphragmatic surface of heart |
| Lateral leads | I, aVL, V5 and V6 | lateral wall of left ventricle |
| Septal leads | V1 and V2 | septal surface of the heart |
| Anterior leads | V3 and V4 | anterior wall of the right and left ventricles |

Table 1.1: The ECG leads register the electrical activity from different angles [25].

most comprehensive report of the heart status, and it is commonly used in hospital as a diagnostic tool[30]. However, it is not suitable for long ECG recordings, such as continuous monitoring of intensive care patients. For this purpose, a subset of relevant leads is sometimes adopted. In most cases, the choice is not too restrictive. The research community has proposed different ways to reconstruct the 12-lead ECG records starting from a 3-leads record. Piotr Augustyniak et al. [10] showed that a three channels record could be used to reconstruct the 12-lead ECG, with some extent of distortion, using the Dower and Levkov transformations. An artificial neural network has shown the capability of learning a transformation from 3-lead recording to the 12-lead standard[8]. Those researches demonstrate that the 12-leads record contains a considerable amount of redundant information legitimizing the uses of a lower number of leads for longer ECG recording sessions. Another problem arises when a very long ECG record has to be analyzed by the medical personnel. It is impractical for human intelligence to scrutinize a 24-hour long record. In this context, a computerized algorithmic observer could inspect with the same attention and precision for an everlasting ECG record.

## 1.2 Computer-aided electrocardiography

Nowadays, the ECG is a widespread diagnostic tool to assess patient clinical status[41]. It is non-invasive, simple to obtain and can be used to monitor constantly patients in intensive care. The early detection and correct classification of abnormalities can increase the chance of a successful

treatment[1]. Nonetheless, manual interpretation of the ECG is a time-consuming task and requires specialized personnel[13]. For this purpose, since the late 1950s, the computerized interpretation of ECG started to take place to facilitate health care diagnosis and speed up the ECG interpretation reducing healthcare costs. A computer program interpreter has to take into account several technical aspects. First of all, the signal must be acquired, digitized and cleaned.
The ECG signal often presents a high amount of noise coming from non-physiological and physiological artefacts. The power-line interference is the primary non-physiological source of noise, together with other equipment problems. However, the physiological artefact is the most dangerous since they can significantly affect the signal's shape causing incorrect diagnosis. They are produced by the patient muscle contractions, skin inferences and baseline wanders linked to respiration. When the preprocessing stage is concluded, the algorithm must recognize the different waveforms, such as the P-wave or the QRS complex, and measure their duration and amplitudes. This procedure is often easy when the ECG is a normal sinus rhythm, but in the presence of arrhythmia, the waves mentioned above are difficult to identify. Finally, the algorithm outputs a diagnosis. Often they can handle only a restricted set of possible diagnosis. Despite the advances in technology, the computer interpretation of ECG struggles to achieve a cardiologist level of accuracy.
Moreover, the inter-variability between diagnosis made by different programs is very high [36]. Major diagnostic algorithms' limitation lies in the way they elaborate the ECG: a set of hard-coded rules identifies the waves, their amplitudes and duration, and produces the diagnosis. On the other hand, the innate human ability of visual pattern recognition yields more flexible analysis. Joseph S. Alpert[5] wrote in an editorial in the American journal of medicine: "What is the reason that the most sophisticated computer ECG interpreting software makes so many mistakes? I think the answer lies in the remarkable and extensive capacity of the human brain to recognize visual patterns.". The requirement of algorithms capable of emulating human pattern recognition ability encounters the recent progress in machine learning. The latest Deep Learning models based on the Convolutional Neural Networks are able to recognize a wide variety of objects demonstrating human-like abilities. Furthermore, computational Neuroscience has shown how the human and monkey's neurons react similarly to the receptive fields in the Convolutional Neural Networks[38].

## 1.3 Machine learning based systems

As soon as the diagnostic algorithms based on hard-coded rules started to show their limitations, the research has moved towards machine learning models. The advantages are evident: the algorithm can decide by itself the best set of rules which maximize the correct diagnostic rates. However, the designer has to collect enough data to allow the machine to learn statistically relevant rules. In other words, the designer delegates the development of the decision rules to the algorithm, but the problem translates into finding enough training data. Another crucial step that has remained under the designer's responsibility is the *features extraction*. The machine learning algorithms often do not understand the data directly out of the box. Hence, someone has to make it machine-readable. We will see that the deep learning models can take care of this problem by requiring additional data. A typical machine learning diagnostic system can identify anomalous behaviours at heartbeats level or rhythms level. In the first case, the algorithm processes only one heartbeat a time, assessing if it behaves as a Sinus beat. In the other case, the algorithm correlates information from the heart rate and the abnormalities in the single heartbeats for a longer period and classifies the ECG records in a rhythm category. In general, these algorithms follow these steps:

1. **Noise removal**. As we have previously seen, the ECG recordings are affected by different sources of noise, which can compromise the algorithm's correct functioning.

2. **Heartbeat detection and heart rate measure**. The heartbeats are often localized by detecting the QRS complex, which is the most prominent wave. Then, the beginning of the heartbeat, which is the onsite of the P wave, and the end, which is the T wave's offsite, are determined in relation to the R peak. However, in particularly abnormal heartbeats, the P wave and the T wave could not be present. The earliest algorithm for R peak detection is the Pan–Tompkins algorithm [61]. The structure is straightforward, and it is illustrated in 1.3. The detection of the R peak can also be used to compute the heart rate of the ECG by averaging the R-R distances.

3. **Heartbeat classification** or **rhythm classification**. If the problem consists in the classification of single heartbeats, then the detected heartbeats are extracted from the ECG, a set of features is derived and

inputted in the machine learning model, which computes the diagnosis. In the other case, the model needs information coming from the entire ECG.

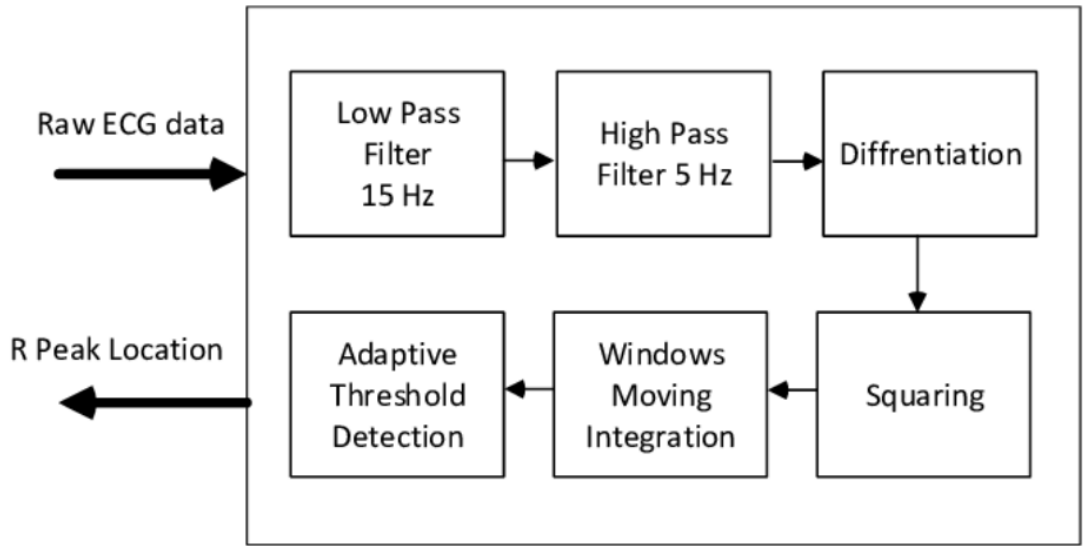



Figure 1.3: The Pan–Tompkins is one of the first algorithm used to detect R-peak locations[44].

The literature abounds of algorithms that try to classify heartbeats or rhythms. Hongzu Li et al. show in a recent review that most of them can classify the anomalies in a restricted number of classes [44]. The problem is that the possible anomalies that could affect the ECG are plenty; some are rare [15] and difficult to observe in a short ECG record. Hence, the classification algorithms available in the literature are limited by the data on which they are trained and tested. They lack generalization to other pathologies not included in the database. Furthermore, most of the ECG data contains Sinus rhythm, which is the most common to observe. The abnormal cases are rare and more valuable to detect. The imbalance between classes is another substantial limitation to common ECG diagnostic algorithm. For this purpose, it seems more logical to face the problem in an **unsupervised/semi-supervised anomaly detection** framework.

## 1.4 Thesis aim and objectives

How can a doctor understand the abnormality of an ECG? I am not a doctor, but presumably he knows the patterns of a Sinus heartbeat, he has

trained its pattern recognition ability after years of practice. Furthermore, he knows that a specific pattern coincides with the electrical activity of a part of the heart. Then, he connects his knowledge of the heart cycle pumping process to the patterns detected on the ECG signal and understand if it is all right. Is It possible to mimic such a behaviour with a machine? Probably not entirely, in particular the ability of causal reasoning is the biggest hurdle to face[37]. However, the outstanding pattern matching abilities exhibited by the most recent Deep learning models can at least reproduce the intuitive process which induce the doctor to notice anomalous pattern in the ECG.

The purpose of this thesis is exactly expressed by the latter sentence. In this thesis we will face the problem of recognition of abnormal patterns inside the ECG signal. But since there is a wide variety of possible abnormalities, it is limiting to face the problem as a classical supervised classification. Hence, the algorithm will be trained only on normal data capturing the normal ECG variability without using any prior knowledge of anomalous patterns. Then, it will use the knowledge acquired by normal data to infer the normality or abnormality of new data. The framework is often referred in literature as **semi-supervised**, or sometimes **unsupervised**, **anomaly detection** which defines the class of situation in which there is an abundance of normal data, but the abnormal one is rare, sometimes unknown or unobserved.

In practice, we will focus our attention on abnormalities at heartbeat level. We will use a Deep Generative Model derived by an Adversarial Autoencoder in order to capture the Sinus heartbeat distribution throughout a set of latent variables. When a new heartbeat is presented to the model, it will generate an anomaly score based on the reconstruction error which indicates the fit of the new heartbeat to the distribution learned. The model can also pinpoint the abnormal time ticks inside the beat providing an additional level of interpretability. In real application, the model could be used for continuous monitoring of patients in intensive care, by alarming doctors only when abnormal heartbeats are detected. Hence, its purpose is not directly deliver a diagnosis but help the medical personnel to focus their attention on relevant region on very long ECG records.

The model is tested on the heartbeats from MIT-BIH database, reaching 0.95 ROC-AUC and 0.92 PR-AUC outperforming other baselines and competing with state-of-the-art models.

## 1.5 General overview of anomaly detection

The section is intended to formalize the problem faced by the thesis. First, we provide a small digression on anomaly detection.

Anomaly detection is a broad class of problems whose fil rouge is to find anomalous observations in the data leveraging data-driven techniques. It has a vast range of application areas such as intrusion detection systems, bank fraud detection, health monitoring, outlier detection. Each application field has its challenges: for this reason, it is quite tricky solving the problem in its most general form. The primary challenge lies somehow hidden in the problem itself and consist in answer precisely to the question: **what is an anomaly?** The notion of anomaly is as clear and intuitive to a human as it is imprecise and fuzzy to a machine. Giving a general definition of what is an anomaly is challenging. The literature has described it as **a pattern that does not conform to expected normal behavior** [16]. However, normal behaviour depends on the specific application field.

### 1.5.1 Taxonomy of anomaly detection preoblems

During years of research, a precise taxonomy of possible anomalies was realized. In general, the kind of anomaly depends on the type of data used in the application(e.g. time series, single observations, images) and on the specific objectives.

- A **point anomaly** is a single observation of a set which is significantly different from the rest. An easy example is the case of detection of bank fraud transactions. Given a set $\mathcal{X}$ of transactions, detects if any of those is atypical or anomalous. In this case an abnormal point can correspond to a fraudulent operation.
- A **contextual** or **conditional anomaly** is an observation that should not appear in a specific context. They are common in problems where the notion of normal behaviour depends on some external information. Time series, spatial or graph-based anomalies are the most prominent example of this class.
- A **group** anomaly happens when more anomalous observations lie together in a cluster. It is important to point out that single anomalies could be regarded as normal in the context of the anomalous cluster.

- Lukas et al.[70] introduced the dichotomy between **low-level** and **high-level** or **semantic** anomalies. Deep learning showed impressive ability in learning complex semantical attributes of data thanks to a hierarchy of concepts. Therefore they are able to distinguish between dogs or cats, different shapes or different topics in a discussion. On the other hand, shallow networks can model only low-level features of data.

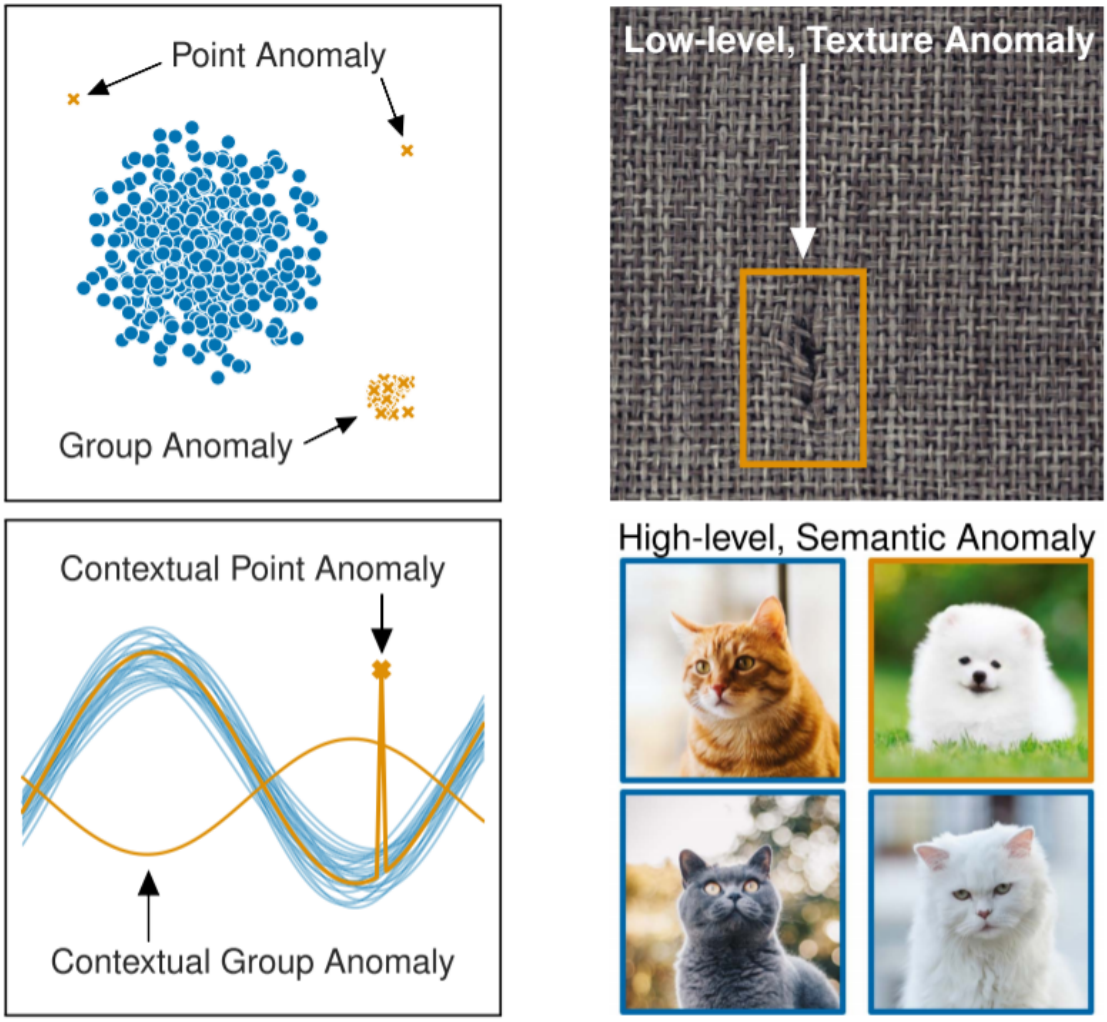



Figure 1.4: The types of anomalies. Figure from Lukas et al. [70]. The semantic anomaly(difference between cats and dogs) that they have introduced, can be captured only by deep hierarchical structure of concepts.

Another determinant aspect characterizing anomaly detection problems is the availability of labelled data. In the optimal case, we can access to a balanced quantity of normal and abnormal instances. In this case, the

anomaly detection problem simplifies in a supervised binary classification problem. However, in most cases, the anomalous data is rare, scarce, and sometimes unknown. Sometimes the anomaly detection model is supposed to find observation which brings new knowledge of the environment. The latter case is known as **novelty detection**, and it is fundamental in scientific applications to discover new, unexpected behaviours. When anomalous examples are rare or difficult to obtain, two possible settings open up.

- We speak about **unsupervised** anomaly detection when the available data is unlabelled. This often happen when anomalous data is unknown, e.g. in an outlier or novelty detection framework. A common strategy in this case is to find most common data patterns and treat it like normal instances. On the other hands, rare patterns are classified as anomalies, outliers or novelties depending on the objectives of the problem.
- The **semi-supervised** is the intermediate case. In the literature this terminology is dubious: sometimes [17] is referred to the case when available data is both labelled and unlabelled, other times [59][3][6] is reported to the case when only normal behaviour data is available. However, in both cases, the data does not contain sufficient amount of abnormal observations, but there is still a good amount of labelled normal data.

The last central aspect relevant to point out is the nature of the anomaly score function. The anomaly score function is the way the anomaly detector can report the anomalies. In general, there exist three different options:

- the detector assigns a probability of normality to data instances, $p_{normal}(\mathbf{x})$. This is the preferred mode because it provides an understandable and quantitative information on how much new data can be regarded as an anomaly. Then a new observation $\mathbf{x}_0$ can be classified as anomalous with a decision function

$$C(\mathbf{x}_0) = \begin{cases} 1 & \text{if } p_{normal}(\mathbf{x}_0) \leq \alpha \\ -1 & \text{otherwise} \end{cases} \tag{1.1}$$

  where a value of 1 indicates that $\mathbf{x}_0$ is anomalous and $\alpha$ is a threshold which regulates the false positive rate.
- the anomaly score function is of the form $A(\mathbf{x}_0) : \mathcal{X} \mapsto \mathbb{R}^+$. The new observations are classified as anomalous with the same decision function

in 1.1, but this time the threshold is much more difficult to set because it has not a clear interpretation as in the last case.

- The algorithm directly assign a label, anomalous or normal, to data instances. This is the worst case since the detector hides information that quantify the abnormality of observations.

Concluding this general discussion, it is natural to talk more specifically about our anomaly detection problem on heartbeats. Our problem tackles two different types of anomalies. In first glance, we want to detect irregular heartbeats in the ECG. This can be regarded as a heartbeat collection, hence an irregular heartbeat is a point anomaly. Secondly, we aim to find anomalous time ticks inside the heartbeat. These types of anomalies are therefore contextual since the normal behaviour change with time. Finally, the heartbeats abnormalities are studied and classified by the medical community, but some of them appears rarely making very difficult the creation of a balanced dataset which contains all the heartbeat types. On the other hand, the sinus beat is very easy to observe. Therefore, the problem reflects the aforementioned aspects of a semi-supervised framework. The next section provides a formal definition of our specific anomaly detection problem.

### 1.5.2 Problem statement

Our problem can be summarized as:
Let $\mathcal{B}$ be a collection of multivariate time series $(\mathbf{x}_i)_{i=1,..,N}, \mathbf{x}_i \in \mathbb{R}^L$, representing the **normal** heartbeats, where $N$ is the length and $L$ is the number of leads. Given an unseen heartbeat $\mathbf{x}_0 \in \mathbb{R}^{L \times N}$:

- identify if $\mathbf{x}_0$ has a behaviour which deviate significantly from $\mathcal{B}$;
- spot the anomalous time ticks.

As it stands, the previous definition is vague because it lacks of a precise delineation of what it is anomaly, and what it is meant by normal behaviour. Hence the keystone passage is to define precisely these last concepts. As we have seen in previous section, the majority of heartbeats have a normal behaviour, so it does make sense to rephrase the problem in a statistical fashion. Following past authors[24][12], we define the normal behaviour as a probability measure on the observation space. The formal definition of the problem becomes straightforward.

**Definition 1.5.1. (Problem)** Assume there exist $p_{normal}(\mathbf{x})$, the unknown probability measure on $\mathbb{R}^{L\times N}$ of normal heartbeats. Let $\mathcal{B} = (\mathbf{x}_1, ..., \mathbf{x}_n) \sim p_{normal}(\mathbf{x})$ be a random sample from $p_{normal}(\mathbf{x})$. Estimate $p_{normal}(\mathbf{x})$ given $\mathcal{B}$. Then let $\mathbf{x}_0 \sim p_{heartbeats}(\mathbf{x})$ a sample from the distribution of all possible heartbeats. Asses if $\mathbf{x}_0$ comes from $p_{normal}(\mathbf{x})$. □

**Remark 1.5.2.** Once we obtain $\hat{p}_{normal}(\mathbf{x})$, i.e. the estimated probability distribution, we have to detect out of distribution samples. A new $\mathbf{x}_0$ is classified as anomaly (or out of distribution) by thresholding an anomaly score function $A(\mathbf{x}) : \mathbb{R}^{L\times N} \mapsto \mathbb{R}^+$. It is tempting to use directly the likelihood value $A(\mathbf{x}) = 1 - \hat{p}_{normal}(\mathbf{x}_0)$ as anomaly score, but we will see that this intuition is not straightforward to implement. The same can be done for the single time ticks $a(x_i) : \mathbb{R} \mapsto \mathbb{R}^+$. □

### 1.5.3 Why to go deep

We have seen that a step towards the problem's solution consists in finding the probability distribution of the normal heartbeats. Obviously, we can't use any parametric known form because assuming a specific structure for the ECG data could be too restrictive. Moreover using classical non-parametric density estimators, e.g. Parzen window estimator [63], is not feasible because they suffer from the well-known curse of dimensionality problem [42]. They require exponentially increasing amount of data, as the input dimension increase, to reach a reliable solution. This phenomenon happens because those kinds of algorithms often rely on a **symbolic representation** of the input. An algorithm based on a symbolic representation is characterized by the fact that the relationship between the number of parameters (or examples in the case of non-parametric density estimators) and the number of regions they can define on the input space is linear. However, as the dimension increases, the number of input space regions for which the algorithm has to make a decision (assign a probability in the case of density estimation) grows exponentially. On the other hand, algorithms which rely on a **distributed representation** can break the input space in $k^n$ regions, with n parameters that assumes k values [27]. This motivates using deep generative models: leveraging the ability of neural networks to build layer by layer a distributed representation of the input[33]. We will discuss better about them in the next chapter. Those algorithms are not without problems: the density estimated is often implicit and can't be evaluated directly.

## 1.6 Related works

A wide variety of anomaly detection models are proposed in the literature. As we have seen, traditional machine learning models applied to ECG data make use of label information for both normal and abnormal data in a supervised classification framework. Most of them extract a set of features representative for the heartbeat, then adopt a classification algorithm such as Support Vector Machines, Kth nearest neighbor(KNN) or Linear/quadratic Discriminant Analisys. For instance, Zhang et al. [84] adopt a Support Vector Machine to classify the heartbeats throughout a set of 46 features extracted from them. Those methods showed good performances, but they are restricted to the pathology covered by the database on which they are trained.
The thesis is more related to the works which make use only of normal labeled data during training in a semi-supervised/unsupervised anomaly detection fashion. This allows to generalize to any type of pathology, by sacrificing a precise diagnosis for the abnormalities detected. Some of them aim to understand the normal data distribution $p_{normal}(\mathbf{x})$, e.g. Kernel Density Estimator (KDE)[67], Histogram based models or Mahalanobis[66], Energy based models or and Flow models [26]. Other aim to estimate directly an $\alpha-$*density level set* $C_\alpha = \arg\inf_C \{C|p_{normal}(C) \geq 1-\alpha\}$,by complying the Vapnik's Principle[81] which suggests to avoid solving a more general problem (e.g. estimate $p_{normal}(x)$) as intermediate step to solve a simpler one (e.g. anomaly detection). They are often referred as One Class Classification methods and they are the One-Class Support Vector Machine[19], the Support Vector Data Descriptor [28], One-class Kernel Fisher Discriminants [68]. An additional class of methods are trained to encode and reconstruct only normal instances. The reconstruction error is then used as anomaly score. They are related with both density estimation and one-class classification methods, and they are the most strictly related to our case study. The Principal Component Analysis (PCA)[80] and the AutoEncoder are the most classical ones. Both models learn a transformation which maps the input data in a space usually smaller or with nicer properties, by minimizing the objective

$$\mathcal{L}(\theta, \phi) = ||\mathbf{x} - D_\theta \circ E_\phi(\mathbf{x})||^2, \tag{1.2}$$

where $E_\phi(\mathbf{x})$ is the transformation that encodes the data and $D_\theta(\mathbf{h})$ its inverse. PCA assumes both functions linear and orthogonal, while the autoencoder allows for non-linearities provided by neural networks[1]. Recently, the Generative Adversarial Networks have captured the interest of many researchers, who have tried to apply them in anomaly detection. Anogan[75], Ganomaly[3] are some examples which use a reconstruction error as anomaly score. Two works that make use of GAN[72] and autoencoder frameworks are applied to ECG data: the BeatGAN[85] and the AnoBeat[60]. All those models will be examined further after a general discussion on the Deep Generative Models.

### 1.6.1 Contributions

Our work is conceptually similar to the last two cited, but it uses an adversarial autoencoder to capture the normal heartbeat distribution. To the best of our knowledge, this is the first work, where an adversarial autoencoder is used for anomaly detection of heartbeats signals. Moreover, to the best of our knowledge, it is the first work that allows the introduction of additional patient related information in the modelling of the normal heartbeat distribution. Thanks to the regularization of the latent space achieved by imposing an independent Gaussian prior, the model is more robust in detecting point anomalies inside the heartbeat than other autoencoder-based methods. Finally the results obtained are competitive with other state-of-the-art method, confirming that an additional discriminator between reconstructed and real heartbeats is not needed, as done by BeatGAN and AnoBeat Framework.
Finally, the model trained on the MIT-BIH database[56] has shown to learn different heartbeats distributions depending on patient sex. This interesting property could help to discover how sex, or other additional information, impacts on the ECG morphology, letting more personalized diagnosis adaptable to specific patient's condition.

[1] A function obtained by composing consecutively affine non linear transformations $f(\mathbf{x}) = \sigma(W\mathbf{x}+b)$, with $\sigma(x)$ often referred as activation function and W and b trainable parameters.

# Chapter 2

# General framework

The last chapter underlined the importance of finding the probability distribution of the sinus beat. This chapter will introduce a class of generative models that estimate a probability distribution on high dimensional spaces. The first section provides a general overview of the deep generative models. The following sections will focus their attentions on the differentiable generator networks, a subclass of deep generative models which rely on the ability of neural networks to approximate any continuos and differentiable function[62] given enough capacity to the model.
In the last part, we discuss their uses in anomaly detection.

## 2.1 Deep generative models

Generative modelling aims to learn the probability distribution of the input. Given a set of observation $\mathbf{x}_1$ , ..., $\mathbf{x}_n \sim p(\mathbf{x})$ that comes from an unknown distribution we put our attention on estimate $p(\mathbf{x})$. Classical statistical method proceed assuming a known parametric form $p_\theta(\mathbf{x})$ and the problem becomes estimating the most likely $\theta$ given the observed data. Other methods do not assume any parametric form and estimate the data's density directly using the observed data. **Deep generative models** (DGM) keep the same aim of classical generative modelling, but they leverage the ability of deep hierarchical structures of latent variables to model complex interactions between visible variables. They assume that the input $\mathbf{x} \in R^N$ can be explained by a **hidden representation h** which usually lies in a lower-dimensional space $R^h$. Moreover, an advantage of using deeper models is the natural ability of multi-layered networks to learn a distributed representation of the input[33]. Deep generative modelling promises to scale

up classical machine learning to face more complex problems such us modelling natural language, images or high dimensional time series data(e.g. the ECG). The language often used to describe those models is borrowed by **graphical models**. Graphical modelling, also defined as structured probabilistic modelling, is a formalism to describes probability distributions using graphs. The graph is a mathematical structure studied in graph theory and is composed of vertices and edges. The latter model the relationship between the former. The formalism used by graphical models represents random variables as vertices and relation between them as edges. The advantage of using such a formalism is that we can model directly significant interaction between random variables, using a directed edge when there is a clear causal relationship, otherwise using an undirected edge. Picture 2.1 shows an example of a probability distribution expressed as a graphical model. The following sections will thoroughly discuss deep gen-


(a) directed (b) undirected

Figure 2.1: The directed graph represents a probability distribution $p(a, b, d, e) = p(d|a)p(e|a)p(b|a)p(a)$. The undirected graph represents a probability up to a normalizing constant,often called partition function, $p(a, b, d, e) \propto f_{a,d}(a, d)f_{a,b}(a, b)f_{b,e}(b, e)f_{b,c}(b, c)$

.

erative models attributable to directed graphical models. Models relying on undirected edges notable to mention are Boltzmann machines[31] and Deep belief networks[32]. In particular, Guido Mont´ufar has shown that the deep Boltzmann machines are universal approximators of probability distributions[55].

However, those models are challenging to train. A deep generative model define a joint probability distribution $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{h})$ on the latent and visible variables which depends on some parameters $\boldsymbol{\theta}$. Those parameters are estimated following the maximum likelihood principle, hence maximizing with respect to $\boldsymbol{\theta}$ the log-likelihood of the data

$$log p_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}[log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{h}) - log(p_{\boldsymbol{\theta}}(\mathbf{h}|\mathbf{x}))]. \tag{2.1}$$

where the expectation in the right hand side is taken with respect to $p(\mathbf{h})$. Sometimes the log-likelihood normalizing constant, called *partition function*, is intractable. Hence, taking the gradient to maximize the log-likelihood requires to approximate the partition function

$$\nabla_{\boldsymbol{\theta}} log(\tilde{p}_{\boldsymbol{\theta}}(\mathbf{x})) - \nabla_{\boldsymbol{\theta}} log(Z(\theta)) \tag{2.2}$$

denoted by $Z(\boldsymbol{\theta})$ in equation 2.2.
Sometimes the problem lies in the intractability of the posterior distribution $p_{\boldsymbol{\theta}}(\mathbf{h}|\mathbf{x})$, which is required in order to obtain the marginal likelihood $p_{\boldsymbol{\theta}}(\mathbf{x})$. In this cases a common strategy is to maximize a lower bound $\mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, q)$ of the log-likelihood. This technique is known as *variational inference*. The lower bound is called *evidence lower bound* and is defined as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, q) = log(p_{\boldsymbol{\theta}}(\mathbf{x})) - \mathcal{D}_{KL}(q_{\boldsymbol{\theta}}(\mathbf{h}|\mathbf{x})||p(\mathbf{h}|\mathbf{x})) \tag{2.3}$$

where $\mathcal{D}_{KL}(p||q)$ is the Kullback–Leibler divergence between the intractable posterior distribution and an arbitrary $q(\mathbf{h}|\mathbf{x})$ posterior which approximate the desired one. The Kullback–Leibler divergence is defined as

$$\mathcal{D}_{KL}(p||q) = \int p(x) log\left(\frac{p(x)}{q(x)}\right) dx, \tag{2.4}$$

and it is an asymmetric measure of the difference between the two probability distribution.
This section will no further delve into the generalities regarding deep generative model. The concepts already presented are sufficient to deal with a specific subclass of deep generative models which nowadays has gained wide popularity: differentiable generator networks.

## 2.2 Differentiable Generator Networks

Differentiable generator networks are a subset of deep generative models which can be described using a directed graphical model. The main assumption of this kind of models is that the underlying sampling process

which generate the observed data is caused by a set of latent variables. They are based on the idea of *inverse transform sampling*[20].

**Proposition 2.2.1.** *Given an univariate random variable $X$ with invertible cumulative distribution function (CDF) $F(x) = \int_{-\infty}^{x} p(v)dv$, then the random variable $X' = F^{-1}(U)$, where $F^{-1}(x)$ is the inverse of the CDF and $U$ is a random variable such that $U \sim Unif([0,1])$, has the same distribution of $X$, i.e $X \simeq X'$.*

*Proof.* Note that the CDF of $U$ is $F_U(u) = \int_0^u dv = u$, hence we have that the CDF of $X'$ is

$$\begin{aligned} F_{X'}(x) &= \mathcal{P}(X' \leq x) \\ &= \mathcal{P}(F^{-1}(U) \leq x) \\ &= \mathcal{P}(U \leq F(x)) = F_U(F(x)) = F(x) \end{aligned}$$

□

**Remark 2.2.2.** If we can sample from the uniform distribution, we can sample from all random variables whose inverse CDF is known. □

The differentiable generator networks join together the insight proposed by 2.2.1 and the neural networks' ability to be a universal approximator[62] of continuos and differentiable functions. The neural network provides a nonlinear change of variable $\mathbf{g}(\mathbf{h}, \boldsymbol{\theta})$, which is able to transform the distribution over $\mathbf{h}$, which is usually set to be uniform or Gaussian, into the desired distribution over $\mathbf{x}$. In other words, the generator network $g(\mathbf{x}, \boldsymbol{\theta})$ imposes implicitly a probability distribution over $\mathbf{x}$ which could be theoretically computed using the change of variable formula,

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{p_{\mathbf{h}}(g^{-1}(\mathbf{x}, \boldsymbol{\theta}))}{|det(\frac{\delta \mathbf{g}}{\delta \mathbf{h}})|} \tag{2.5}$$

where at the denominator there is the determinant of the Jacobian of the generator network. The advantage of using such a procedure is that they can be trained using gradient descend. Experimentally, models optimized using gradient descend showed impressive performances in supervised problems. However, Goodfellow et al. cleverly pointed out that in supervised problems both input $\mathbf{h}$ and output $\mathbf{x}$ are specified in the data, meanwhile, in this case, the optimization procedure needs to find by itself the right association between the two spaces $\mathbf{h}$ and $\mathbf{x}$[27].

### 2.2.1 Variational Autoencoders

The Variational Autoencoder (VAE) [40] is the quintessential example of differentiable generator network trained entirely with variational bayes techinque. Also in this case the data is assumed to depend on a set of latent variables $\mathbf{h} \sim p(\mathbf{h})$, implying a conditional density on the visible data $p(\mathbf{x}|\mathbf{h})$. In the variational autoencoders these distributions are often assumed to be Gaussian

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{h}) = N(\mu_{\boldsymbol{\theta}}(\mathbf{h}), \sigma^2_{\boldsymbol{\theta}}(\mathbf{h})I) \tag{2.6}$$

$$p(\mathbf{h}) = N(0, I), \tag{2.7}$$

where $\mu_{\boldsymbol{\theta}}(\mathbf{h})$ and $\sigma_{\boldsymbol{\theta}}(\mathbf{h})$ are neural networks. Hence train a variational autoencoder means maximize the log-likelihood of the data $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ with respect to $\theta$

$$\max_{\boldsymbol{\theta}} \sum_{i=1}^{N} logp_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}), \tag{2.8}$$

where the log-likelihood for a single data point is

$$logp_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = \mathbb{E}_{\mathbf{h}\sim p(\mathbf{h})}\left[logp_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{h}) - logp_{\boldsymbol{\theta}}(\mathbf{h}|\mathbf{x}^{(i)})\right]. \tag{2.9}$$

The posterior distribution $p_{\boldsymbol{\theta}}(\mathbf{h}|\mathbf{x})$ is intractable because it is parametrized by a neural network. Therefore, we estimate an approximated posterior by maximizing the ELBO, previously seen in 2.3. The objective function becomes

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = logp_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - \mathcal{D}_{KL}(q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{h}|\mathbf{x}^{(i)}))) \tag{2.10}$$

$$= \mathcal{H}(q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x}^{(i)})) + \mathbb{E}_{\mathbf{h}\sim q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x}^{(i)})}\left[logp_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{h})\right] \tag{2.11}$$

$$= \mathbb{E}_{\mathbf{h}\sim q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x}^{(i)})}\left[logp_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{h})\right] - \mathcal{D}_{KL}(q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x}^{(i)})||p(\mathbf{h})) \tag{2.12}$$

where $q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x}^{(i)})$ is the approximated posterior distribution parametrized by a neural network, and $\mathcal{H}(q(x)) = \mathbb{E}_{x\sim q(x)}\left[-logq(x)\right]$ is the entropy of the distribution $q(x)$. In equation 2.12 the first term discloses an interesting connection with classical autoencoders. If we intepret the $p_{\boldsymbol{\phi}}(\mathbf{x}^{(i)}|\mathbf{h})$ as a decoder network, usually called the *generator* network, and $q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x}^{(i)})$ as the encoder network, called *inference network* or *recognition model*, maximizing the expected value in 2.12 means maximize the log-likelihood of reconstructed samples. It is the same objective of a probabilistic autoencoder which is trained to maximize

$$\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathbb{E}_{\mathbf{h}\sim q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x})}\left[logp_{\boldsymbol{\theta}}(\mathbf{x}|h)\right]. \tag{2.13}$$

If we choose the encoder and decoder to be deterministic, we recover the objective function characterized by the mean squared error between real and reconstructed samples

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \sum_{i=1}^{N} \left(\mathbf{x}^{(i)} - D_{\boldsymbol{\theta}}(E_{\boldsymbol{\phi}}(\mathbf{x}^{(i)}))\right)^2. \tag{2.14}$$

Indeed, a deterministic autoencoder could be thought to assume that the data can be expressed as

$$\mathbf{x} = D_{\boldsymbol{\theta}}(\mathbf{h}) + \boldsymbol{\epsilon} \tag{2.15}$$

$$\mathbf{h} = E_{\boldsymbol{\phi}}(\mathbf{x}), \tag{2.16}$$

where $\boldsymbol{\epsilon}$ is a random reconstruction error distributed as a isotropic multivariate Gaussian random vector. The latter distributional assumption implies a distribution also to the data.

$$\mathbf{x}|\mathbf{h} \sim N(D_{\boldsymbol{\theta}}(\mathbf{h}), I). \tag{2.17}$$

Maximizing the log-likelihood of the data is equivalent to minimize the mean square error

$$\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \sum_{i=1}^{N} logp(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \max_{\boldsymbol{\theta},\boldsymbol{\phi}} - \sum_{i=1}^{N} (\mathbf{x}^{(i)} - D_{\boldsymbol{\theta}}(E_{\boldsymbol{\phi}}(\mathbf{x}^{(i)})))^2 \tag{2.18}$$

$$= \min_{\boldsymbol{\theta},\boldsymbol{\phi}} \sum_{i=1}^{N} (\mathbf{x}^{(i)} - D_{\boldsymbol{\theta}}(E_{\boldsymbol{\phi}}(\mathbf{x}^{(i)})))^2. \tag{2.19}$$

The second term in 2.12 encourage the approximate posterior to approach the latent variables distribution. This term can be computed analitically if the approximate posterior is assumed to have a Gaussian form

$$q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x}^{(i)}) = N(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}^{(i)}), \boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{x}^{(i)})I). \tag{2.20}$$

Under this assumption the Kull-back Leiber divergence in 2.12 becomes[40]

$$\mathcal{D}_{KL}(q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x}^{(i)})||p(\mathbf{h})) = \int q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x}^{(i)}) \left(logp(\mathbf{h} - q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x}^{(i)}))\right) d\mathbf{h} \tag{2.21}$$

$$= -\frac{1}{2} \sum_{i=1}^{n_h} \left(1 + log(\sigma_j^2(\mathbf{x}^{(i)})) - \mu_j^2(\mathbf{x}^{(i)}) - \sigma_j^2(\mathbf{x}^{(i)})\right) \tag{2.22}$$

where $n_h$ is the latent space dimension.

At this point the objective function is clearly defined, we now need a practical differentiable estimator for the variational lower bound. Differentiability

**Algorithm 1:** Minibatch version of the Auto-Encoding VB (AEVB) algorithm.

Set parameters $L$
**Initialise**
$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialise
**For** $j = 0, \ldots, J$
$\mathbf{X}^M \leftarrow$ Random minibatch of M datapoints
$\boldsymbol{\epsilon} \leftarrow$ Random sample from $p(\boldsymbol{\epsilon})$
$\mathbf{g} \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathcal{L}^B(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}^M, \boldsymbol{\epsilon})$ Compute gradients of ELBO estimator
$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Update with $\mathbf{g}$ using SGD
**Return** $\boldsymbol{\theta}, \boldsymbol{\phi}$

is required to update the parameters of the neural networks through gradient descend algorithms. Diederik et al. [40] proposed the Stochastic Gradient Variational Bayes estimator (SGVB) for the ELBO and the Auto-Encoding Variational Bayes(AEVB) alghoritm 1 to update the weight of the generator and inference networks. These netwroks are stochastics, hence it is not clear how to differentiate their outputs. One way to solve this problem is to reparametrize the stochastic network as a function of a source of random noise, which does not depend on the parameters, and the input. In the literature, this procedure is known as **reparametrization trick** or **stochastic back-propagation**. In our case, we have assumed both generator and inference network to follow a Gaussian distribution, hence the reparametrization of the inference network

$$\mathbf{h}|\mathbf{x}^{(i)} \sim N(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}^{(i)}), \boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{x}^{(i)})I) \tag{2.23}$$

is

$$\mathbf{h} = \mathbf{g}_{\boldsymbol{\phi}}(\mathbf{x}^{(i)}, \boldsymbol{\epsilon}) = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}^{(i)}) + \boldsymbol{\sigma}^2_{\boldsymbol{\phi}} \odot \boldsymbol{\epsilon} \tag{2.24}$$

where $\boldsymbol{\epsilon} \sim N(0, I)$ is the source of stochasticity which does not depend on the weights $\boldsymbol{\phi}$, and $\odot$ denotes the element-wise product. Thanks to the reparametrization we can obtain the gradients of the output $\mathbf{h}$ with respect to the parameters $\boldsymbol{\phi}$. The SGVB estimator of the ELBO is

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) \approx \hat{\mathcal{L}}^B(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \tag{2.25}$$

$$\frac{1}{L}\sum_{l=1}^{L}\left(logp_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)})\right) - \frac{1}{2}\sum_{i=1}^{n_h}\left(1 + log(\sigma_j^2(\mathbf{x}^{(i)})) + \mu_j^2(\mathbf{x}^{(i)}) - \sigma_j^2(\mathbf{x}^{(i)})\right) \tag{2.26}$$

$$where \qquad \mathbf{z}^{(i,l)} = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}^{(i)}) + \boldsymbol{\sigma}^2_{\boldsymbol{\phi}} \odot \boldsymbol{\epsilon}^{(l)}\,, \quad \boldsymbol{\epsilon}^{(l)} \sim N(0, I). \tag{2.27}$$

At this point, we have a precise objective function, an estimator and an algorithm to train the two networks.

### 2.2.2 Generative Adversarial Networks

Generative adversarial networks, or GAN [72] learn the target distribution thanks to a procedure based on a zero sum game between two players. The **generator** network plays against the **discriminator** network. The former produces samples $\hat{x} = G_{\boldsymbol{\theta}}(\mathbf{h})$ from the latent space $\mathbf{h} \sim p(\mathbf{h})$ and it is trained to fool the latter which, like a binary classifier, learns to distinguish between real data points $\mathbf{x}$ and the generated ones $\hat{\mathbf{x}}$. In an ideal setting, where the neural networks used have enough capacity to represent any possible function, the GAN objective function presents a global minimum where the generator distribution $p_g$ reaches the data distribution $p_r$, i.e. $p_g = p_r$.

**Definition 2.2.3.** (Idealised GAN)(Goodfellow et al. 2014). Let $\mathcal{H} \subset R^{n_h}$ be the latent space, and $\mathcal{X} \subset R^N$ be the observation space. Let $p_h$ be the distribution over the latent space and $p_r$ be the distribution of data points. The idealised GAN objective is

$$\min_G \max_D V(G, D), \tag{2.28}$$

where $G : \mathcal{H} \mapsto \mathcal{X}$ and $D : \mathcal{X} \mapsto [0,1]$, and

$$V(G, D) = \mathbb{E}_{x \sim p_r} [log(D(x))] + \mathbb{E}_{h \sim p_h} [log(1 - D(G(h)))] \tag{2.29}$$

□

**Remark 2.2.4.** We can see that the discriminator has to assign values near 1 to real samples and the opposite for fake ones to maximize this objective. On the contrary, the generator tries to minimize the objective fooling the discriminator into assigning values near 1 to fake samples. □

**Proposition 2.2.5.** *(Goodfellow et al. 2014). For fixed G the optimal discriminator is*

$$D^*_G(x) = \frac{p_r(x)}{p_g(x) + p_r(x)} \tag{2.30}$$

*Proof.* The training criterion for the discriminator D, given any generator G, is to maximize the quantity $V(G, D)$

$$V(G, D) = \int_{\mathcal{X}} p_r(x) log(D(x)) dx + \int_{\mathcal{H}} p_h(h) log(1 - D(G(h))) dh \tag{2.31}$$

$$= \int_{\mathcal{X}} p_r(x) log(D(x)) + p_g(x) log(1 - D(x)) dx \tag{2.32}$$

For any $(a, b) \in \mathcal{R}^2 \setminus (0,0)$, the function $y \mapsto alog(y) + blog(1-y)$ achieves its maximum in [0,1] at $\frac{a}{a+b}$. The discriminator does not need to be defined outside of $Supp(p_r) \cup Supp(p_g)$ concluding the proof. □

**Remark 2.2.6.** (Goodfellow et al. 2014) Considering only perfect discriminator, the minimax game in 2.2.3 can be reformulated as:

$$C(G) = \max_D V(G, D) \tag{2.33}$$
$$= \mathbb{E}_{x\sim p_r}\left[logD^*_G(x)\right] + \mathbb{E}_{h\sim p_h}\left[log(1 - D^*_G(G(h)))\right] \tag{2.34}$$
$$= \mathbb{E}_{x\sim p_r}\left[logD^*_G(x)\right] + \mathbb{E}_{x\sim p_g}\left[log(1 - D^*_G(x))\right] \tag{2.35}$$
$$= \mathbb{E}_{x\sim p_r}\left[log\left(\frac{p_r(x)}{p_r(x) + p_g(x)}\right)\right] + \mathbb{E}_{x\sim p_g}\left[log\left(\frac{p_g(x)}{p_r(x) + p_g(x)}\right)\right] \tag{2.36}$$

□

**Theorem 2.2.7.** *(Goodfellow et al. 2014). The global minimum of the ideal training criterior C(G) is achieved if and only if $p_g = p_r$. At that point, C(G) achieves the value $-log(4)$.*

*Proof.* For $p_g = p_r$, $D^*_G(x) = \frac{1}{2}$. Hence by inspecting 2.35, we find $C(G) = log\left(\frac{1}{2}\right) + log\left(\frac{1}{2}\right) = -log(4)$. To see that this is the best possible value of C(G), reached only if $p_g = p_r$, observe that

$$C(G) = -log(4) + D_{KL}\left(p_r||\frac{p_r + p_g}{2}\right) + D_{KL}\left(p_g||\frac{p_r + p_g}{2}\right) \tag{2.37}$$

where $D_{KL}(p||g)$ is the Kullback–Leibler divergence. Hence

$$C(G) = -log(4) + 2\dot{D}_{JS}(p_r||p_g) \tag{2.38}$$

where $D_{JS}(p||g)$ is the Jensen–Shannon divergence. Since it is always non-negative and zero if and only if $p_g = p_r$, the gloabl minimum of C(G) is -log(4). □

Despite the theoretical properties, the GAN presents in practice some convergence problems:

- **Failure to improve**. Even if the solution exists, sometimes the dinamics of gradient descend alghoritms used to train the networks prevents the neural networks to reach the optimal solution. The causes of this problem are matter of research. Goodfellow et al.[72] proposes that a perfect discriminator which classifies with high confidence fake and real samples, saturates the gradients which should be used to train the generator.

- **Mode collapse**. This problem shows up when $p_r$ is multimodal. Under the circumstances the generator may learn to produce samples only for one mode of the data distribution. Metz et al.[54] shows experimentally that a GAN may fail to learn a simple mixture of 2D Gaussain distributions.

Research has proposed a variety of solutions to those problems. Some of them[53] aims to ameliorate the objective function keeping the adversarial strategy framework. Next section will focus on the argument proposed by Arjovsky et al.[7] that shows how minimizing the Wasserstein distance between $p_r$ and $p_g$ is a preferable choice than the Jensen-Shannon divergence.

### 2.2.3 Wasserstein GAN

As we have seen in previous section, the classical GAN approach results in the minimization of the Jensen-Shannon divergence between the distribution of real data $p_r$ and generated one $p_g$. Liu et al. [46] generalizes this approach introducing the concept of **adversarial divergence**.

**Definition 2.2.8.** (Liu et al. (2017). Let $\mathcal{X}$ be a topoligical space and $\mathcal{F} \subset C_b(\mathcal{X})$, $\mathcal{F} \neq$, where $C_b(\mathcal{X}^2)$ is the set of bounded continous functions on $\mathcal{X}^2$. An adversarial divergence $\tau$ over $\mathcal{X}$ is a function

$$\mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X}) \mapsto \mathcal{R} \cup \{\infty\} \tag{2.39}$$

$$(\mu, \zeta) \mapsto \tau(\mu||\zeta) = \sup_{f \in \mathcal{F}} \mathbb{E}_{\mu \otimes \zeta}[f], \tag{2.40}$$

where $\mathcal{P}(\mathcal{X})$ denotes the space of probability measures on $\mathcal{X}$. □

**Example 2.2.9.** *(Liu et al. (2017) Choosing*

$$\mathcal{F} = \{x, y \mapsto log(D(x)) + log(1 - D(y))|D \in \zeta\}, \tag{2.41}$$

$$\zeta = \{f : \mathcal{X} \mapsto [0,1]|f continous\} \tag{2.42}$$

*recovers the objective function of the Idealised GAN in 2.35.*

The Wasserstein GAN arises from a particular choice of adversarial divergence, where

$$\mathcal{F} = \{x, y \mapsto v(x) - v(y)|v \in \zeta\}, \tag{2.43}$$

$$\zeta = \left\{ f \in C_b(\mathcal{X})|\frac{||f(x) - f(y)||}{||x - y||} \leq K \right\} \tag{2.44}$$

, where $K > 0$ is a constant and $\mathcal{X}$ is assumed to be a compact metric space. the theoretical advantages for this alternative choice relies in the nicer convergence properties characterizing this alternative divergence. Minimizing the divergence obtained in 2.43 is equivalent to minimize the **Heart mover** or **Wasserstein-1** distance between $p_r$ and $p_g$.

**Definition 2.2.10.** Let $\mu$ and $\zeta$ be two probability measures on a compact metric space $(\mathcal{X}, d)$. The **Heart mover** or **Wasserstein-1** distance is given by

$$d_W(\mu, \zeta) := \inf_{\gamma \in \prod(\mu,\zeta)} \mathbb{E}_{(x,y)\sim\gamma} \left[ ||x - y|| \right], \tag{2.45}$$

where $\prod(\mu, \zeta)$ denotes the set of all joint distributions $\gamma(x, y)$ such that, for all $A \in \mathcal{B}(\mathcal{X})$,

$$\gamma(A, \mathcal{X}) = \mu(A)$$
$$\gamma(\mathcal{X}, A) = \zeta(A).$$

□

The above form of Wasserstein distance is quite difficult to compute, the following theorem derives a more tractable formula.

**Theorem 2.2.11.** *(The Kantorovich-Rubinstein Duality). Let $(\mathcal{X}, d)$ be a compact metric space, and let $Lip_1(\mathcal{X})$ be the set of functions $f : \mathcal{X} \mapsto \mathcal{R}$ such that*

$$||f||_1 = sup \left\{ \frac{||f(x) - f(y)||}{||x - y||} | x, y \in \mathcal{X}, x \neq y \right\} \leq 1. \tag{2.46}$$

*Then $f \in Lip_1(\mathcal{X})$ is Lebesgue integrable with respect to any probability measures on $\mathcal{X}$, and*

$$d_W(\mu||\zeta) = \sup_{f \in Lip_1(\mathcal{X})} \left( \mathbb{E}_{x\sim\mu} \left[ f(x) \right] - \mathbb{E}_{x\sim\zeta} \left[ f(x) \right] \right) \tag{2.47}$$

*Proof.* The result comes from optimal transport theory. See Villani [82] for a proof. □

This result shows clearly that the Wasserstein distance is an adversarial divergence with the choice 2.43. Moreover it implies also the following result which will be used to understand the theoretical advantages in using the Wasserstein distance as substitute of the Jensen-Shannon divergence.

**Corollary 2.2.12.** *(Basso (2015), Corollary 1.4). Let $(\mathcal{X}, d)$ be a compact metric space. Then $d_W$ defines a metric on $\mathcal{P}(\mathcal{X})$.*

*Proof.* See Basso et al. [11]. □

The discussion so far has rigorously dealed with the general framework of adversarial divergences. In this framework, the GAN learns the data distribution $p_r$ by minimizing a specific divergence. The convergence of $p_g$ to $p_r$ depends on the kind of topology induced by the divergence choosen. The following results shows that the topology induced by the Wasserstein distance on the space of probability measures $\mathcal{P}(\mathcal{X})$ is **weaker** than the topology induced by the Jensen-Shannon divergence. If a topology is weaker than another topology, the set of convergent sequences of the latter is a subset of former's one.

**Theorem 2.2.13.** *((Arjovsky et al. (2017), theorem 2). Let $\mathcal{X}$ be compact, and let $\mu, (\mu_n) \subset \mathcal{P}(\mathcal{X})$. Then, as $n \mapsto \infty$,*

- *$d_{TV}(\mu||\mu_n) \to 0$ if and only if $d_{JS}(\mu||\mu_n) \to 0$.*
- *$d_W(\mu||\mu_n) \xrightarrow{d} 0$ if and only if $\mu_n \xrightarrow{d} \mu$, where $\xrightarrow{d}$ denotes the convergence in distribution.*
- *if either $d_{KL}(\mu||\mu_n) \to 0$ or $d_{KL}(\mu_n||\mu) \to 0$, then $d_{JS}(\mu_n||\mu) \to 0$.*
- *if $d_{TV}(\mu||\mu_n) \to 0$ then $d_W(\mu||\mu_n) \xrightarrow{d} 0$.*

*Proof.* proof in appendix A. □

**Remark 2.2.14.** $d_{KL}$ is the Kull-back leiber divergence and

$$d_{TV}(\mu, \nu) = \sup_{A \in \mathcal{B}(\mathcal{X})} |\mu(A) - \nu(A)| \tag{2.48}$$

is the total variation distance. □

The theorem shows that the convergence under $d_W$ is implied by the stronger topology induced by $d_{JS}$ but not conversely. This concept is showed by the following example.

**Example 2.2.15.** *(Arjovsky et al. (2017), Example 1). Let $Z \sim U[0,1]$ be uniformly distributed over the unit interval. Let $\mathbb{P}_0$ be the distribution of $(0, Z) \in \mathbb{R}^2$. Now let $g_\theta(z) = (\theta, z)$, with $\theta \in \mathbb{R}$ and $\mathbb{P}_\theta$ the distribution for $g_\theta(Z)$. In this case:*

- $d_W(\mathbb{P}_0||\mathbb{P}_\theta) \rightarrow |\theta|$
- $d_{JS}(\mathbb{P}_0||\mathbb{P}_\theta) = \begin{cases} log2 & if\theta \neq 0 \\ 0 & if\theta = 0 \end{cases}$
- $d_{KL}(\mathbb{P}_0||\mathbb{P}_\theta) = d_{KL}(\mathbb{P}_\theta||\mathbb{P}_0) = \begin{cases} \infty & if\theta \neq 0 \\ 0 & if\theta = 0 \end{cases}$

*Hence, when $\theta_n \rightarrow 0$, the sequence $\mathbb{P}_{\theta_n}$ converges to $\mathbb{P}_0$ only under the Wasserstein distance.*

We conclude that the topology induced by $d_W$ is weaker than $d_{JS}$ and then more suitable for the GAN training. When the Wasserstein GAN is implemented in practice, the critic function $v$ in the objective 2.43 is parametrized by a neural network. Hence it is difficult to respect the Lipschitz constraint required by the Wasserstein distance. The original paper presenting the WGAN framework proposed to clamp the weight of the critic in a bounded box. Obviuosly, this is very restrictive and it could limit too much the capacity of the neural network. Ishaan Gulrajani et al.[29] proposed to add a penalty on the gradients of the critic to the objective function, which becomes

$$\mathbb{E}_{\mathbf{x}\sim p_r}\left[D_\omega(\mathbf{x})\right] - \mathbb{E}_{\mathbf{x}\sim p_g}\left[D_\omega(\mathbf{x})\right] + \lambda\,\mathbb{E}_{\hat{\mathbf{x}}\sim p_{\hat{x}}}\left[\left(||\nabla_{\hat{\mathbf{x}}} D_\omega(\mathbf{x}) - 1||_2\right)^2\right], \quad (2.49)$$

where $\hat{\mathbf{x}} = t\mathbf{x} + (1-t)\mathbf{y}$, with $t \sim U([0,1])$ and $\mathbf{x} \sim p_r$, $\mathbf{y} \sim p_g$. They showed experimentally that in this way the critic could learn more complex functions while preserving the good property of the WGAN. The algorithm 2 is one of the state-of-the-art method to train GAN.

### 2.2.4 Adversarial Auto Encoder

Alireza Makhzani et al. [52] proposed the Adversarial Auto Encoder (AAE) framework which adapts a simple autoencoder structure in a deep generative model. The model is trained as a normal autoencoder by minimizing the objective function

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \sum_{i=1}^{N} (\mathbf{x}^{(i)} - G_{\boldsymbol{\theta}}(E_{\boldsymbol{\phi}}(\mathbf{x}^i)))^2, \quad (2.50)$$

but involves an additional regularization term which forces the **aggregated posterior** distribution $q(\mathbf{h})$ to match the prior latent distribution $p(\mathbf{h})$. The

**Algorithm 2:** WGAN with gradient penalty. (Ishaan Gulrajani et al., 2017)

Set gradient penalty coefficient $\lambda$, the number of critic per generator iteration $n_{critic}$, the batch size $m$, Adam hyperparameters $\alpha, \beta_1 \beta_2$.

**Initialise**

  $\boldsymbol{\theta}, \boldsymbol{\omega} \leftarrow$ Initialise

**while** $\theta$ **has not converged do**

  **for** $t = 1, .., n_{critic}$ **do**

    **for** $i = 1, ..., m$ **do**

      Sample real data $\mathbf{x} \sim p_r$, latent variable $\mathbf{z} \sim p(\mathbf{z})$, and $\epsilon \sim U[0,1]$.

      $\tilde{\mathbf{x}} \leftarrow G_\theta(\mathbf{z})$

      $\hat{\mathbf{x}} \leftarrow \epsilon \mathbf{x} + (1 - \epsilon)\tilde{\mathbf{x}}$

      $L^{(i)} \leftarrow D_{\boldsymbol{\omega}}(\tilde{\mathbf{x}}) - D_{\boldsymbol{\omega}}(\mathbf{x}) + \lambda \left(||\nabla_{\hat{\mathbf{x}}} D_{\boldsymbol{\omega}}(\hat{\mathbf{x}}) - 1||_2\right)^2$

    $\boldsymbol{\omega} \leftarrow \text{Adam}(\nabla_{\boldsymbol{\omega}} \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, \alpha, \beta_1, \beta_2)$

  Sample batch of latent variables $\{\mathbf{z}^{(i)}\}_{i=1}^{m} \sim p(\mathbf{z})$

  $\boldsymbol{\theta} \leftarrow \text{Adam}(\nabla_{\boldsymbol{\theta}} \frac{1}{m} \sum_{i=1}^{m} -D_{\boldsymbol{\omega}}(G_{\boldsymbol{\theta}}(\mathbf{z})), \boldsymbol{\theta}, \alpha, \beta_1, \beta_2)$

posterior aggregate distribution is the distribution induced by the encoder over the latent space $\mathcal{H} \subset \mathbb{R}^{n_h}$, defined as

$$q(\mathbf{h}) = \int q(\mathbf{h}|\mathbf{x}) p_r(\mathbf{x}) d\mathbf{x}, \tag{2.51}$$

where $p_r(\mathbf{x})$ is the data distribution on the observation space $\mathcal{X} \subset \mathbb{R}^N$. In order to match the prior distribution, the Encoder network is guided by an adversarial strategy borrowed by the aforementioned GAN framework. While a discriminator network tries to distinguish between samples generated by the real prior $p(\mathbf{h})$ and samples generated by the encoding distribution $q(\mathbf{h}|\mathbf{x})$, the encoder learns to fool the discriminator by generating samples as similar as possible to the prior's ones. The training procedure described in algorithm 3 computes the objective function in 2.52 which make use of a stochastic encoder $q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x})$ and a deterministic decoder $D_{\boldsymbol{\theta}}(\mathbf{h})$

$$\min_{\boldsymbol{\theta}, \boldsymbol{\phi}} \max_{\boldsymbol{\omega}} [\, \mathbb{E}_{\mathbf{h} \sim p(\mathbf{h})}[log(D_{\boldsymbol{\omega}}(\mathbf{h}))] + \mathbb{E}_{\mathbf{h} \sim q_{\boldsymbol{\phi}}(\mathbf{h})}[log(1 - D_{\boldsymbol{\omega}}(\mathbf{h}))] \tag{2.52}$$

$$- \mathbb{E}_{\mathbf{h} \sim q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x})}[log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{h})]], \tag{2.53}$$

where the first two expectation are the adversarial loss for the encoder and discriminator $D_{\boldsymbol{\omega}}(\mathbf{h})$ expressed with the classical GAN framework. Obviously, the same objective function can be rewrite using the WGAN objective. The AAE is strictly related with the VAE: while the latter minimize the Kullback–Leibler divergence between the latent distribution and the encoder aggregated posterior, the AAE minimize an adversarial divergence.

**Algorithm 3:** AAE training algorithm

Set gradient penalty coefficient $\lambda$, the number of critic per generator iteration $n_{critic}$, the batch size $m$, Adam hyperparameters $\alpha, \beta_1\beta_2$.

**Initialise**

$\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{\phi} \leftarrow$ Initialise

**for** $i = 1, ..., N_{iter}$ **do**

Sample real data batch $\mathbf{X}^M = (\mathbf{x}_1, ..., \mathbf{x}_M)$, $\mathbf{x} \sim p_r$

**Reconstruction phase**

$g \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \frac{1}{M} \sum_{m=1}^{M} \frac{1}{L} \sum_{l=1}^{L} \left( log p_{\boldsymbol{\theta}}(\mathbf{x}^{(m)}|\mathbf{h}^{(m,l)}) \right)$, where $\mathbf{h}^{(m,l)} \sim q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x}^{(m)})$

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow \text{Adam}(g, \alpha, \beta_1\beta_2)$

**Regularization phase**

Same GAN training 2 considering the encoder as generator.

This allows the modeler more freedom in the choice of the latent prior distribution to impose at the encoder, without having to compute analytically the Kullback–Leibler divergence between $q(\mathbf{h})$ and $p(\mathbf{h})$. Moreover Alireza Makhzani et al. [52] showed that the AAE is able to capture better the data manifold structure than the VAE.

## 2.3 Generative modelling in anomaly detection

In the previous chapter, we have defined the problem of anomaly detection in our specific framework. We have cast it in a statistical framework while showing that find the probability distribution of normal data is crucial. As we have seen the generative models are capable of modelling the sampling process through a set of latent variables and they seems to accomplish our aim. However, evaluate the density is sometimes not possible. In the case of GAN evaluating explicitly the likelihood requires computing the change of variable formula in equation 2.5, hence take the inverse of a neural network. In the case of VAE we can evaluate explicitly the conditional density $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{h})$ but obtain the marginal $p_{\boldsymbol{\theta}}(\mathbf{x})$ requires integrating over the latent space $\mathbf{h}$ computing

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_{\sim p(\mathbf{h}}[p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{h})]. \tag{2.54}$$

This could be estimated using a Markov chain Monte Carlo estimator, but since $\mathbf{h}$ is often high dimensional a massive amount of sample is required to obtain sufficient coverage of the latent space. Some other techniques were proposed to estimate the marginal likelihood [40], but in our specific framework of anomaly detection, this integral should be estimated for each

new observation, resulting in a procedure which may be computationally too expensive for a real time application. Moreover, a recent study showed that the likelihood estimated with deep generative models are unreliable [57]. The researchers have showed that the state-of-the-art models trained on a dataset assign higher likelihood values to sample from other datasets. Moreover the study shows that the estimated likelihood values depend on low-level statistics of the observations, and demonstrate how changing the color of the image inputs results in higher likelihood values. Lars Maaløe et al.[49] proposed BIVA, a deep generative model which make use of skip-connection to enhance flow of information between layer of latent variables and avoid inactive units. In the study, they showed that using latent variables in higher hierarchical layers produces likelihood values more suitable for anomaly detection applications. Hyunsun Choi et al.[18] advocate that the likelihood value should not be used at all in anomaly detection problems with high dimensional data. They argue that the simple one-tailed test made in lower dimension could be inappropriate in higher dimensional spaces. The intuition that in-distribution-samples should have higher likelihoods does not hold in higher dimensions. The fact is explained by the difference between the typical set and high likelihood values in high dimensional spaces. As example, they showed that in an isotropic Gaussian of 784 dimensions, the origin has the highest likelihood value, but it is high atypical because the majority of probability mass lies in the annulus of radius $\sqrt{784}$. Hence, despite Bishop et al. [14] suggested their application in anomaly detection, their implementation is not straightforward. For these reasons using the likelihood value as anomaly score, as suggested by 1.5.2, is not an easy task. When the model follows an autoencoder-like structure, such as the VAE or AAE and some models derivated by GAN, the **reconstruction error** is often used as surrogate. Next section motivates the use of this alternative metric, underlying its relationship with the likelihood-based anomaly score.

### 2.3.1 The reconstruction error as anomaly score

The reconstruction error is often used as anomaly score when the model has a *encoder - decoder* structure. Let $G_{\mathbf{g}} : R^h \mapsto R^N$ be the *decoder* or *generator*, and $E_{\mathbf{e}} : R^N \mapsto R^h$ the *encoder* or *inference network*, where $h$ and $N$ are respectively the dimension of the latent representation and of the input space. The encoder-decoder models are trained to minimize the

reconstruction loss

$$\min_{\mathbf{e},\mathbf{g}} d(\mathbf{X}, (G_{\mathbf{g}} \circ E_{\mathbf{e}})(\mathbf{X})) + \mathcal{R}, \tag{2.55}$$

where $\mathcal{R}$ is a regularization, and $d(\mathbf{x}, \mathbf{y})$ is measure of dissimilarity between the original observation and its reconstruction. The regularization in this problem is of critical importance since it avoids learning the trivial solution given by the identity mapping. Therefore, the anomaly score is defined directly as

$$A(\mathbf{X}_0) = d(\mathbf{X}_0, (G_{\mathbf{g}} \circ E_{\mathbf{e}})(\mathbf{X}_0)). \tag{2.56}$$

Usually $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$ the Euclidean distance. The heuristic justification to the use of reconstruction error is attributable to the basic idea that an encoder-decoder model trained to minimize the reconstruction loss on some specific data will reconstruct effectively only new observations that share similar characteristics with the ones already seen. This intuition is perfectly reasonable and takes its foundations on a geometrical perspective related to *manifold learning*. Manifold learning is a vast topic in which autoencoders shines. This section is not the right place to deal extensively about this topic, but we can mention some of the essential concepts inherent to our discussion.

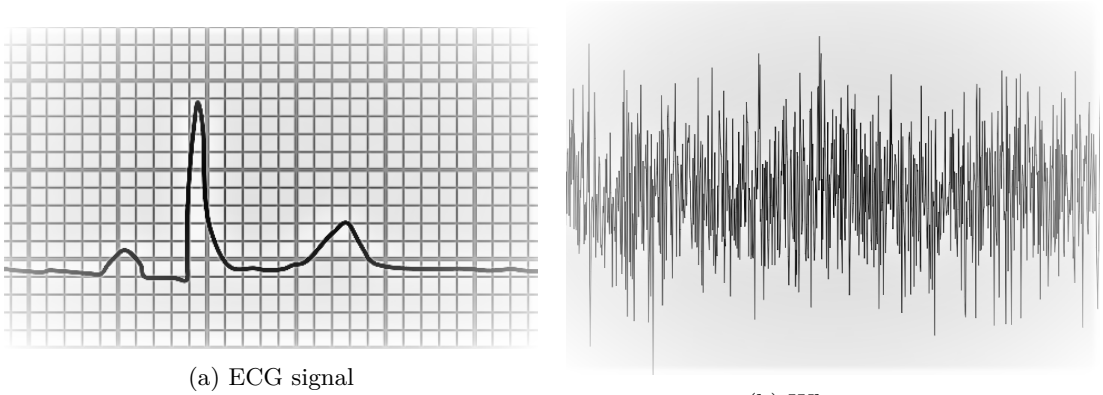
#### geometrical interpretation of the reconstruction error

First of all we give a definition of manifold.

**Definition 2.3.1.** A n-dimensional manifold is a *topological space* such that for each point there exist a neighborhood that is *homeomorphic* to the Euclidean space of dimension n. □

**Definition 2.3.2.** A neighborhood of a point is homeomorpich to a n-dimensional Euclidean space if there exist a continuos and invertible mapping $f : \mathcal{M} \mapsto R^n$ between the neighboorhood and the Euclidean space. □

The basic idea that motivates manifold learning is that the high dimensional data such as images often lies in a tiny subset of the whole Euclidean space hosting them. Figure 2.2 explain this latter concept visually. Hence manifold learning aims to capture the structure of the manifold which embrace the data. The principal component analysis[80] is a renowned statistical procedure which can be interpreted as a simple manifold learning



(a) ECG signal

(b) White noise

Figure 2.2: The ECG signal can be represented by a vector in $\mathbb{R}^N$, where $N$, the length of the signal, could be very high. However, the subset of all possibles ECG in $\mathbb{R}^N$ is very restricted. By sampling randomly a vector in $\mathbb{R}^N$ the probability to obtain something similar to an ECG is infinitely small. We will probably obtain something similar to the signal on the right. Therefore, the idea of manifold learning is to derive the low-dimensional subset of $\mathbb{R}^N$ which contains the data.

algorithm. The PCA aims to find a new coordinate system applying a linear transformation to the data that captures most of the data variability. The hope is that only a few dimensions are sufficient to explain the data in the new coordinate system. Therefore, the new coordinate system is the manifold that accounts most data information, and the linear transformation is the continuous and invertible function that maps the data from the initial space to the manifold. The autoencoder can be interpreted as an extension of the PCA with non-linear mappings. The reconstruction error from a geometrical perspective can be considered the amount of data variability orthogonal to the manifold, i.e. the information lost during projection into the manifold. Hence it can be regarded as a greyscale indicator of "membership" to a manifold. In an anomaly detection framework, an autoencoder trained to capture the normal data manifold will reconstruct poorly anomalous data.

#### Relations with likelihood-based anomaly score

A deep generative model learns the input probability distribution thorough a set of hidden variables $\mathbf{h}$. It can be adapted in anomaly detection by mapping the new observation in the hidden space and then reconstructing it. The distance between the generated data and the original , i.e. reconstruction error, gives an idea on how well the new observation has fitted the

model. We now investigate a possible relation between the reconstruction error and the likelihood value associated to the new observation.
Let $\mathbf{x} \sim p(\mathbf{x})$ be a random vector from the unknown target distribution, $\mathbf{x} \in R^N$. A deep generative model with deterministic generator in an anomaly detection framework can be thought to assume that there exist a latent representation $\mathbf{h} \in R^h$ such that

$$\mathbf{x} = G_{\mathbf{g}}(\mathbf{h}) + \boldsymbol{\epsilon} \tag{2.57}$$

where $G_{\mathbf{g}} : R^h \mapsto R^N$ is the deterministic generator, which is a continuous and differentiable mapping from the latent space to the visible space, and $\boldsymbol{\epsilon} \in R^N$ is a random additive noise which can be seen as the leftover information not learned by the model. Assume that is distributed as a multivariate normal $\boldsymbol{\epsilon} \sim N(0, \Sigma)$. Assume also a distribution for the latent representation $\mathbf{h} \sim P(\mathbf{h})$. These assumptions implies a conditional distribution

$$\mathbf{x}|\mathbf{h} \sim N(G_{\mathbf{g}}(\mathbf{h}), \Sigma). \tag{2.58}$$

Given the above we can rewrite the probability distribution as

$$p(\mathbf{x}) = \mathbb{E}_{\mathbf{h}\sim P(\mathbf{h})}[p(\mathbf{x}|\mathbf{h})] \tag{2.59}$$

$$= \frac{1}{\sqrt{(2\pi)^h \, det(\Sigma)}} \int_{R^h} e^{-\frac{1}{2}(\mathbf{x}-G_{\mathbf{g}}(\mathbf{h}))^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-G_{\mathbf{g}}(\mathbf{h}))} P(\mathbf{h}) d\mathbf{h} \tag{2.60}$$

$$= \frac{1}{\sqrt{(2\pi)^h \, det(\Sigma)}} \int_{R^h} e^{-\frac{1}{2}NR(\mathbf{x},\mathbf{h},\boldsymbol{\Sigma})} P(\mathbf{h}) d\mathbf{h} d\mathbf{h} \tag{2.61}$$

where $R(\mathbf{x}, \mathbf{h}, \boldsymbol{\Sigma}) = \frac{1}{N} D_M^2(\mathbf{x}, G_{\mathbf{g}}(\mathbf{h}))$ can be interpreted as a reconstruction error between $\mathbf{x}$ and $G_{\mathbf{g}}(\mathbf{h})$ based on the Mahalanobis distance[51] $D_M(\mathbf{x}, \mathbf{y})$ with weight matrix $\Sigma$. When $\boldsymbol{\Sigma} = \mathbf{I}$ the reconstruction error $R(\mathbf{x}, \mathbf{h}, \boldsymbol{\Sigma})$ becomes the reconstruction error based on the Euclidean distance. Now let $\mathbf{x}_0$ be a new observation from $p(\mathbf{x})$, and $\hat{\mathbf{h}} = \underset{\mathbf{h}}{argmax}\, p(\mathbf{h}|\mathbf{x}) = E(\mathbf{x}_0)$ be the best encoding for $\mathbf{x}_0$ given by the encoder $E_{\mathbf{e}}(\mathbf{x})$. We expect that the reconstruction error is the minimum using $\hat{\mathbf{h}}$, i.e. $R(\mathbf{x}_0, \hat{\mathbf{h}}, \boldsymbol{\Sigma}) = \underset{\mathbf{h}}{min} R(\mathbf{x}_0, \mathbf{h}, \boldsymbol{\Sigma})$. Under some regularities conditions, the well-known result of the multivariate Laplace approximation[34][77] let to approximate equation 2.61 obtaining

$$p(\mathbf{x}^{(i)}) \simeq \frac{e^{-\frac{1}{2}NA(\mathbf{x}_0)} P(\mathbf{x}_0)}{\sqrt{det(\mathbf{H})}} \left( \frac{2\pi}{\frac{1}{2}N} \right)^{\frac{h}{2}} \tag{2.62}$$

where $A(\mathbf{x}_0) = R(\mathbf{x}_0, E_{\mathbf{e}}(\mathbf{x}_0), \boldsymbol{\Sigma})$ is the anomaly score as defined in 2.56, and $\mathbf{H}$ is its hessian matrix of $A(\mathbf{x})$ in $\mathbf{x}_0$.

The discussion is valid only if we assume Gaussian distributed errors, deterministic generator and an anomaly score function based on the Mahalonobis distance. These assumptions will be satisfied by the model presented in the next chapter, but in general they are quite restrictive. Moreover, equation 2.62 is an approximation which gets worse if $\hat{\mathbf{h}}$ is not the unique minimizer of $R(\mathbf{x}_0, \mathbf{h}, \boldsymbol{\Sigma})$. Hence, in general the reconstruction error has not a direct relation with the likelihood value but it provides an indication on how the observation $\mathbf{x}_0$ is fitted by the model.

The following section describes some deep generative model implementation used in anomaly detection, which use as anomaly score the reconstruction error.

### 2.3.2 Reconstruction error based AD with DGM

The literature is rich of DGM which use the reconstruction error for detect anomalous observations. It is not the purpose of this section to deliver an extensive review of the researches in this field. Therefore, we will discuss some of the most renown models which make use of the DGMs covered in the previous sections.
One of the first works which uses GAN in an anomaly detection framework was proposed by Thomas Schlegl et al. in 2017[75]. The model was applied to medical imaging data, a very useful instrument for monitoring the disease progression. The approach followed is straightforward. In a first step the GAN is used to learn the distribution of normal anatomical data. Secondly, the test data is mapped in the latent space and reconstructed to compute the anomaly score. The GAN by itself does not present a way to map an observation $\mathbf{X}$ in the hidden space. Hence, in the paper was proposed a novel approach to obtain the inverse map based on a stochastic gradient search of the best hidden vector $\mathbf{h}$ which generates the image $\mathbf{X}$. The process is based on the fact that the latent space has smooth transitions, hence sampling images from two point close in the latent space results in two images visually similar. The process is iterative and follows this steps:

1. Generate an image $\hat{\mathbf{X}}$ from $\mathbf{h}_i$ by a forward pass through the generator $G(\mathbf{h})$.

2. Compute a loss between the generated image and the objective

$$\mathcal{L}(\hat{\mathbf{X}}, \mathbf{X}) = \sum \left(\mathbf{X} - \hat{\mathbf{X}}\right) + \sum \left(f(\mathbf{X}) - f(\hat{\mathbf{X}})\right), \tag{2.63}$$

   where $f(\mathbf{x})$ is a function in an intermediate layer of the discriminator, that should capture high semantic features of the images.

3. Obtain the next hidden vector $\mathbf{h}_{i+1}$ by computing the gradient of the loss with respect to $\mathbf{h}_i$.

When the search converges to a $\hat{\mathbf{h}}$, the anomaly score is the value of the loss 2.63 at the last iteration. In the experiments, the models correctly classifies anomalous images and correctly pinpoint anomalous regions inside the images by compute the pixel-wise absolute error between the original image and the reconstructed ones. However, the iterative process needed to obtain the anomaly score is costly and it is not suitable for real time applications. In this regards, Thomas Schlegl et al. in 2019 developed the fast anoGAN [74], which adds an additional training step after the GAN training. The step consists in training an encoder network to learn the map from the observations $\mathbf{X}$ to the latent space by minimizing the reconstruction loss 2.63. During the encoder training, the generator is fixed. They denotes this architecture as $\mathbf{ihi}_f$, which stands for image-latent-image, as opposed to the **hih** architecture. The latter was introduced by [22] and differs by the previous because the learned encoder $E(\mathbf{X})$ is trained by mapping generated images $\hat{\mathbf{X}}$ into the latent space minimizing

$$\mathcal{L}_{hih}(\mathbf{h}) = \frac{1}{d}||\mathbf{h} - E(G(\mathbf{h}))||^2, \tag{2.64}$$

the mean square error between the latent vector which has generated $\hat{\mathbf{X}}$ and the one computed by the encoder. The drawback is that the encoder learns only by mapping generated images, however we are sure that the target $\mathbf{h}$ exists. With the same spirit, many other works were proposed which try to incorporate the training of the encoder during GAN training [3]. A recent review [21] provides a general ovwrview of the state-of-art architectures which involves GAN in anomaly detection.
The VAE are used in anomaly detection too [6]. As anomaly score It is often used the reconstruction probability

$$\mathbb{E}_{\mathbf{h}\sim q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x})}\left[log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{h})\right]$$

instead of the marginal likelihood value for the problems discussed in a previous section. Haowen Xu et al. in 2018 applied successfully VAE for

monitoring KPIs for a web application[83].
Two works are applied to the detection of irregular heartbeats, and they are strictly related to the thesis. The first, the BeatGAN[85], was proposed by Bin Zhou et al. in 2019, and connects the autoencoder framework with the adversarial strategy. It is a convolutional autoencoder which learns to encode and reconstruct the normal heartbeats. Additionally, the decoder is regularized in order to fools a discriminator which has to distinguish between real and reconstructed heartbeats. The objective function that the BeatGAN try to minimize is then

$$\mathcal{L}_G = ||\mathbf{x} - \hat{\mathbf{x}}||^2 + \lambda||f_D(\mathbf{x}) - f_D(\hat{\mathbf{x}})||_2, \tag{2.65}$$

where $f_d(\mathbf{x})$ is a function on an intermediate layer of the discriminator and $\hat{\mathbf{x}}$ is the reconstructed heartbeat. The technique of using a intermediate function is known as feature matching technique, and aims to avoid the renown GAN convergence problems[71]. The discriminator minimize the original GAN objective

$$\mathcal{L}_D = log(D(\mathbf{x})) + log(1 - D(\hat{\mathbf{x}})). \tag{2.66}$$

The second model called AnoBeat[60] was proposed by Yingzi Ou et al. one year later. It try to improve the performance of the previous by adding an additional adversarial regularization to the encoder and a noise control for the heartbeat discriminator. The basic structure remains the autoencoder adversarially regularized by a "visual" critic, i.e. the discriminator between reconstructed and original heartbeats. The anoBeat adds a latent critic between the codes obtained encoding the original signals and the ones obtained by encoding the reconstructed. Hence, the encoder is regularized by fooling the latent discriminator, by using the aforementioned feature matching technique. The autoencoder objective function becomes

$$\mathcal{L}_{E-D} = \lambda_1||\mathbf{x} - \hat{\mathbf{x}}||^2 + \lambda_2||g_D(\mathbf{x}) - g_D(\hat{\mathbf{x}})||_2 \tag{2.67}$$

$$= \lambda_1 L_{rec} + \lambda_2 L_{advz}, \tag{2.68}$$

where $g_D(\mathbf{x})$ is an intermediate layer of the latent discriminator. The decoder is trained separately, while keeping the encoder fixed, to minimize

$$\mathcal{L}_{E-D} = \lambda_1 L_{rec} + \lambda_3 L_{advx} - \lambda_4 L_{advn}, \tag{2.69}$$

where

$$L_{advx} = ||f_D(\mathbf{x}) - f_D(\hat{\mathbf{x}})||^2 \tag{2.70}$$

$$L_{advn} = ||f_D(\tilde{\mathbf{x}} - f_D(\hat{\mathbf{x}}), \tag{2.71}$$

where $\tilde{\mathbf{x}} = \boldsymbol{\epsilon} + \mathbf{x}$ is the original signal with added noise $\boldsymbol{\epsilon} \sim N(0, I)$. The $L_{advn}$ regularization is intended to keep the features of the synthesized signal $\hat{\mathbf{x}}$ far from the noise that the discriminator could eventually learn during training. Both models learns the normal heartbeats distribution implicitly, they are not even intended to generate samples. Their only aim is focused to anomaly detection.
The next chapter presents AAECG, an anomaly detection model related to those mentioned above. The skeleton of this model follows the adversarial auto encoder, but some refinements are introduced to proficiently face the ECG data.

# Chapter 3

# Proposed framework

At this stage, we have underlined the importance of the problem, formalized it in a statistical framework and we have extensively dealt with the general framework of Deep Generative models. Now, we are ready to introduce AAECG, an Adversarial AutoEncoder which capture the normal heartbeats variability by using also additional gender patient information.

## 3.1 Model overview

The AAECG applies the framework of Differentiable Generator networks in a anomalous heartbeat detection problem. The aim is to capture the distribution of normal behaviour heartbeats $p_r(\mathbf{x})$ by modeling it through a set of latent variables $\mathbf{h}$ and some additional information related to the patient to whom the ECG was registered. It is known by the cardiologist community that the morphological features of the Sinus cardiac cycle are influenced by external factors such as the age, sex or race[78][50] of the patient, the medical treatment under he/she is subjected, its psycho-physiological status. A recent study showed that a neural network is capable of understand the age and sex of a patient through its ECG record [9]. While the sex prediction was accurate in 90.5% of patients, the age was less accurate. The researchers claim that the algorithm rather than predict the chronological age, predict the physiological one. To sustain this hypothesis they showed a strong correlation between the wrong predictions biased towards older ages and the presence of heart diseases.
In this thesis only information about sex will be used. The heartbeat is modeled as a vector $\mathbf{x} \in \mathbb{R}^{L \times N}$, where $L$ is the number of lead channels and $N$ the heartbeat samples number. The model assumes that the normal

heartbeat distribution $p_r$ can be expressed as

$$p_r(\mathbf{x}) = p_{\hat{\boldsymbol{\theta}}}(\mathbf{x}) = \frac{p_h(G^{-1}(\mathbf{x}, \hat{\boldsymbol{\theta}}))}{|\frac{\delta \mathbf{G}}{\delta \mathbf{h}}|}, \tag{3.1}$$

for some $\hat{\boldsymbol{\theta}}$, where $G^{-1}(\mathbf{x}, \boldsymbol{\theta})$ is the inverse of $G_{\boldsymbol{\theta}}(\mathbf{h}, S_x)$ which is a continuous differentiable mapping from the latent space $\mathcal{H} \subset \mathbb{R}^{n_h}$ distributed as $\mathbf{h} \sim p_h(\mathbf{h})$ and the sex information $S_x$, to the observation space $\mathcal{X} \subset \mathbb{R}^{L \times N}$. The expression is derived by the change of variable formula which has already been discussed in 2.5. In order to learn the distribution on the observation space, an Adversarial Auto Encoder is adopted. The general architecture of the model is depicted in 3.1. The model is composed by three networks.

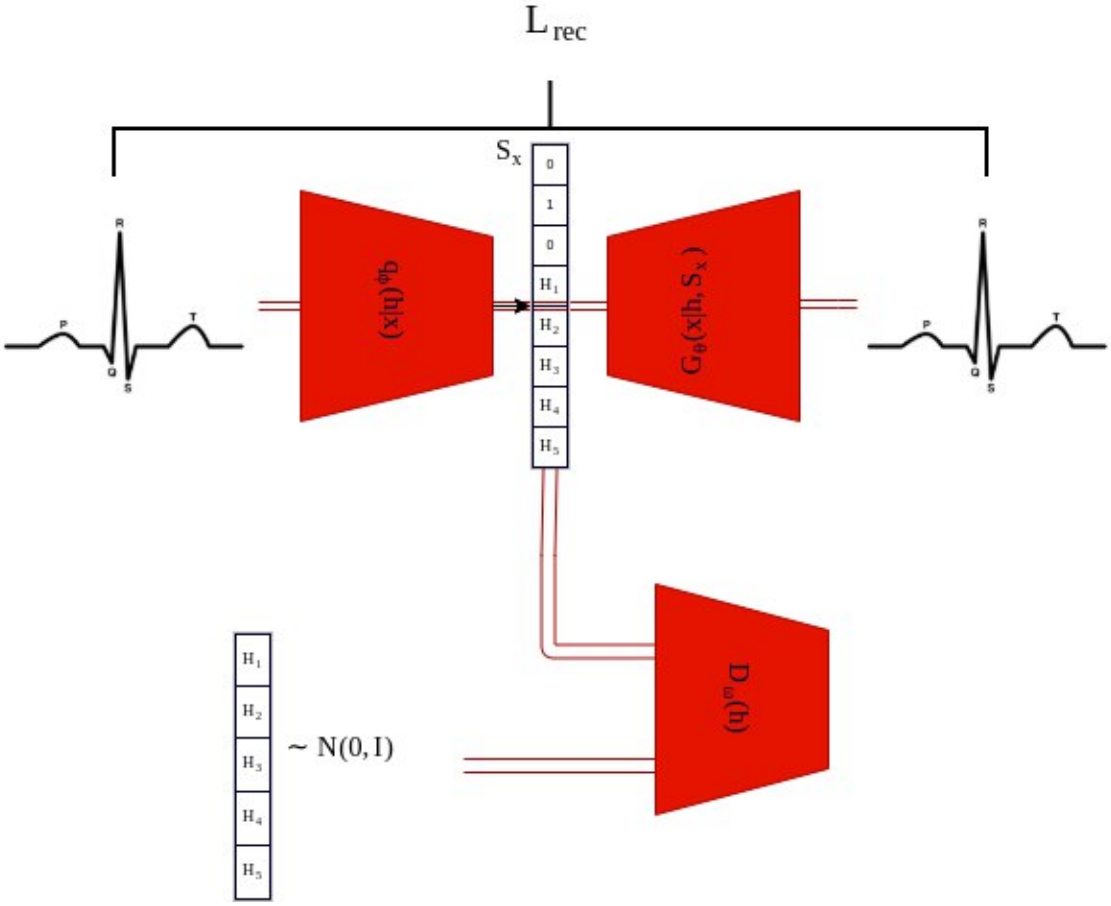



Figure 3.1: AAECG is composed by a stochastic encoder, which maps an ECG into a 5 dimensional vector. A decoder which reconstruct the ECG with the sex information joint to the hidden vector. A discriminator distinguishes between samples generated from the encoder and from an isotropic Gaussian.

A deterministic generator network learns to reconstruct the heartbeats by

minimizing

$$\lambda_{REC}\,\mathbb{E}_{\mathbf{h}\sim q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x})}\left[\sum_{i=0}^{N\times L}\left(\mathbf{x}_{(i)}-G_{\boldsymbol{\theta}}(\mathbf{h}|S_{\mathbf{x}})_{(i)}\right)^2\right]+\lambda_{TV}\,\mathbb{E}_{\mathbf{h}\sim q_{\boldsymbol{\phi}}}\left[TV(G_{\boldsymbol{\theta}}(\mathbf{h}|S_{\mathbf{x}})\right], \tag{3.2}$$

which is the sum between the expected reconstruction error and the expected total variation of the reconstruction. The total variation is defined as

$$TV(\mathbf{x})=\frac{1}{L}\sum_{l=1}^{L}\sum_{i=1}^{N}|x_{(l,i+1)}-x_{(l,i)}|, \tag{3.3}$$

i.e. the mean total variation of each ECG lead. The total variation penalty added to the objective function induce the generator to produce cleaner signals. This strategy was previously adopted in the literature [69]. Minimize the objective in 3.2, without considering the total variation penalty, from a statistical point of view means assuming that the conditional distribution of the data is Gaussian

$$\mathbf{x}|\mathbf{h},S_{\mathbf{x}}\sim N(G_{\boldsymbol{\theta}}(\mathbf{h}|S_{\mathbf{x}}),I). \tag{3.4}$$

Minimizing the expected mean square error is equivalent to maximize the expected log-likelihood of the data

$$\max_{\boldsymbol{\theta}} log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{h},S_{\mathbf{x}})= \tag{3.5}$$

$$\max_{\boldsymbol{\theta}} log\left[(2\pi)^{-\frac{(L\times N)}{2}}e^{-\frac{1}{2}(\mathbf{x}-G_{\boldsymbol{\theta}}(\mathbf{h}|S_{\mathbf{x}}))^T(\mathbf{x}-G_{\boldsymbol{\theta}}(\mathbf{h}|S_{\mathbf{x}}))}\right] \tag{3.6}$$

$$=\max_{\boldsymbol{\theta}}-\frac{1}{2}(\mathbf{x}-G_{\boldsymbol{\theta}}(\mathbf{h}|S_{\mathbf{x}}))^T(\mathbf{x}-G_{\boldsymbol{\theta}}(\mathbf{h}|S_{\mathbf{x}})) \tag{3.7}$$

$$=\min_{\boldsymbol{\theta}}\sum_{i=0,l=0}^{N,L}\left(\mathbf{x}_{(l,i)}-G_{\boldsymbol{\theta}}(\mathbf{h}|S_{\mathbf{x}})_{(l,i)}\right)^2. \tag{3.8}$$

The encoder ,or inference network, learns to infer the best hidden code which generates the heartbeat. While the generator network is assumed to be deterministic, the model uses a stochastic encoder $q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x})$. The encoder in an adversarial auto encoder is trained to match the aggregated distribution $\int q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x})p_r(\mathbf{x})d\mathbf{x}$ to the latent distribution $p_h(\mathbf{h})$. It can be of three types[52]:

- **deterministic**. $q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x})$ can access only to the source of stochasticity in the data distribution $p_r$ in order to match the latent distribution.

- **Gaussian**. We inject Gaussian noise in the output of the encoder by parametrize the mean and the variance of a multivariate Gaussian with the encoder neural network,

$$\mathbf{h}|\mathbf{x} \sim N(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\mathbf{x})I). \tag{3.9}$$

  The neural network parameters $\boldsymbol{\phi}$ can be trained with the reparametrization trick [39], already mentioned in previous chapter. In this case the encoder network can access to an additional source of noise.

- **Universal approximator**. The encoder is a function of the input and a random noise $\psi$, $f(\mathbf{x}, \psi)$. In this case the encoder has not a Gaussian structure, and has more freedom in assuming different distributional shapes. The encoder network can sample from an arbitrary distribution $q(\mathbf{h}|\mathbf{x})$.

During the experiments, the randomness of the data distribution $p_r$ proved to be not sufficient for the encoder to match the imposed prior $p_h(\mathbf{h})$. Hence, a Gaussian shape was imposed to the output of the encoder. Therefore the encoder network $q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x})$ is trained by minimizing the objective function

$$\lambda_{REC}\, \mathbb{E}_{\mathbf{h}\sim q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x})} \left[ \sum_{i=0,l=0}^{N,L} \left( \mathbf{x}_{(l,i)} - G_{\boldsymbol{\theta}}(\mathbf{h}|S_{\mathbf{x}})_{(l,i)} \right)^2 \right] \tag{3.10}$$

$$- \lambda_{ADV}\, \mathbb{E}_{\mathbf{h}\sim q_{\boldsymbol{\phi}}(\mathbf{h})} \left[ D_{\boldsymbol{\omega}}(\mathbf{h}) \right], \tag{3.11}$$

where $D_{\boldsymbol{\omega}}(\mathbf{h})$ is the discriminator in a Wasserstein GAN framework with gradient penalty. Its objective is an estimation of the Wasserstein-1 distance between the aggregated posterior $q_{\boldsymbol{\phi}}(\mathbf{h})$ and the imposed prior $p_h(\mathbf{h})$, obtained by maximizing

$$\mathbb{E}_{\mathbf{h}\sim p_h(\mathbf{h})} \left[ D_{\boldsymbol{\omega}}(\mathbf{h}) \right] - \mathbb{E}_{\mathbf{h}\sim q_{\boldsymbol{\phi}}(\mathbf{h})} \left[ D_{\boldsymbol{\omega}}(\mathbf{h}) \right] - \lambda_{GP}\, \mathbb{E}_{\hat{\mathbf{x}}\sim p_{\hat{x}}} \left[ \left( ||\nabla_{\hat{\mathbf{x}}} D_{\omega}(\mathbf{x})||_2 - 1 \right)^2 \right]. \tag{3.12}$$

Summarizing, the objective of the whole network is

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}; \mathbf{x}) = \tag{3.13}$$

$$\lambda_{REC}\, \mathbb{E}_{\mathbf{h}\sim q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x})} \left[ \sum_{i=0,l=0}^{N,L} \left( \mathbf{x}_{(l,i)} - G_{\boldsymbol{\theta}}(\mathbf{h}|S_{\mathbf{x}})_{(l,i)} \right)^2 \right] \tag{3.14}$$

$$+ \lambda_{TV}\, \mathbb{E}_{\mathbf{h}\sim q_{\boldsymbol{\phi}}} \left[ TV(G_{\boldsymbol{\theta}}(\mathbf{h}|S_{\mathbf{x}}) \right] \tag{3.15}$$

$$+ \mathbb{E}_{\mathbf{h}\sim p_h(\mathbf{h})} \left[ D_{\boldsymbol{\omega}}(\mathbf{h}) \right] - \lambda_{ADV}\, \mathbb{E}_{\mathbf{h}\sim q_{\boldsymbol{\phi}}(\mathbf{h})} \left[ D_{\boldsymbol{\omega}}(\mathbf{h}) \right] \tag{3.16}$$

$$- \lambda_{GP}\, \mathbb{E}_{\hat{\mathbf{x}}\sim p_{\hat{x}}} \left[ \left( ||\nabla_{\hat{\mathbf{x}}} D_{\omega}(\mathbf{x})||_2 - 1 \right)^2 \right], \tag{3.17}$$

**Algorithm 4:** training algorithm

Set parameters $\lambda_{GP}, \lambda_{REC}, \lambda_{TV}, \lambda_{ADV}$, the number of critic per generator iteration $n_{critic}$, the batch size $M$, number of inference network samples $J$, Adam hyperparameters $\alpha, \beta_1\beta_2$.

**Initialise**

  $\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{\phi} \leftarrow$ Initialise

**for** $i = 1, ..., N_{iter}$ **do**

  Sample real data batch $\mathbf{X}^M = (\mathbf{x}_1, ..., \mathbf{x}_M)$, $\mathbf{x} \sim p_r$

  **Encoder Decoder optimization**

  Sample $J \times M$ latent codes $\mathbf{h}^{(m,l)} \sim q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x}^{(m)})$

  Forward pass through Generator $\hat{\mathbf{x}}^{(m,j)} = G_{\boldsymbol{\theta}}(\mathbf{h}^{(m,j)}|S_{x^m})$

  $\mathcal{L}_{rec} \leftarrow \frac{1}{NJ} \sum_{m=1,j=1}^{M,J} \left(\mathbf{x}^{(m)} - \hat{\mathbf{x}}^{(m,j)}\right)^2$

  $\mathcal{L}_{adv} \leftarrow \frac{1}{JM} \sum_{m=1,j=1}^{M,J} \left(D_\omega(\mathbf{h}^{(m,j)})\right)$

  $\mathcal{L}_{tv} \leftarrow \frac{1}{L} \sum_{l=1}^{L} \sum_{i=1}^{N-1} |\hat{\mathbf{x}}_{i+1}^{(m,j)} - \hat{\mathbf{x}}_i^{(m,j)}|$

  $g \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \left(\lambda_{REC}\mathcal{L}_{rec} - \lambda_{ADV}\mathcal{L}_{adv} + \lambda_{TV}\mathcal{L}_{tv}\right)$

  $\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow \text{Adam}(g, \alpha, \beta_1\beta_2)$

  **Discriminator Update**

  **for** $j = 1, ..., n_{critic}$ **do**

    Sample real data batch $\mathbf{X}^M = (\mathbf{x}_1, ..., \mathbf{x}_M)$, $\mathbf{x} \sim p_r$

    Sample $M$ latent codes from inference network $\mathbf{h}_d^{(m)} \sim q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x}^{(m)})$

    Sample $M$ latent codes from imposed prior $\mathbf{h}_r^{(m)} \sim p_h(\mathbf{h})$

    Compute gradient penalty $GP$ of discriminator

    $\mathcal{L}_w \leftarrow \frac{1}{M} \sum_{m=1}^{M} \left(D_\omega(\mathbf{h}_r^{(m)}) - D_\omega(\mathbf{h}_d^{(m)})\right) + \lambda_{GP}GP$

    $g \leftarrow \nabla_{\boldsymbol{\omega}}\mathcal{L}_w$

    $\boldsymbol{\omega} \leftarrow \text{Adam}(g, \alpha, \beta_1, \beta_2)$

and the network is trained by a min-max optimization

$$\min_{\boldsymbol{\phi},\boldsymbol{\theta}} \max_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}; \mathbf{x}). \tag{3.18}$$

The algorithm devoted to train the network by optimizing 3.18, is described in 4.

At convergence, the model has learned an inference network $q_{\hat{\phi}}(\mathbf{h}|\mathbf{x})$ which proposes the most likely $\mathbf{h}$ which generates the given $\mathbf{x}$, and a generator network, which can sample from $p_r$ if assumption in 3.1 holds.
Given a new heartbeat $\mathbf{x}_0$ which is not necessarily from the sinus heartbeat distribution $p_r$, the model assesses its abnormality by computing the reconstruction error as anomaly score. The score is measured by computing

$$A(\mathbf{x}_0) = \frac{1}{J}\sum_{j=1}^{J} MSE_j(\mathbf{x}_0, \hat{\mathbf{x}_0}) \tag{3.19}$$

where

$$MSE_j(\mathbf{x}_0, \hat{\mathbf{x}_0}) = \sum_{i=0}^{N\times L}\left(\mathbf{x}_{(i)} - G_{\boldsymbol{\theta}}(\mathbf{h}_j|S_{\mathbf{x}_0})_{(i)}\right)^2, \quad \mathbf{h}_j \sim q_{\hat{\boldsymbol{\phi}}}(\mathbf{h}|\mathbf{x}), \tag{3.20}$$

i.e. it is a MCMC estimation of the expected reconstruction error. Finally, the unseen heartbeat $\mathbf{x}_0$ is regarded to be anomalous if the anomaly score exceeds a given threshold, which is estimated by maximizing the f1-score on a validation set containing both normal and abnormal heartbeats.
The model provides an additional level of abnormality score. It can pinpoint the single time ticks which deviate significantly from the normal behaviour. This can be realised by computing the mean squared error residual for each time tick

$$a(x_0^{(t_i)}) = \sum_{j=1}^{J}\left(x_0^{(t_i)} - \hat{x}_{0,j}^{(t_i)}\right)^2 \tag{3.21}$$

where $\hat{x}_{0,j}^{(t_i)} = G_{\hat{\boldsymbol{\theta}}}(\mathbf{h}_j|S_{x_0})^{(t_i)}, \mathbf{h}_j \sim q_{\hat{\boldsymbol{\phi}}}(\mathbf{h}|\mathbf{x})$. This latter kind of anomaly score can offer greater interpretability to the model, guiding the attention of the doctor towards the regions of the heartbeat which had led it to be regarded anomalous. A similar approach has already been proposed by some above-mentioned model, such has AnoGAN, BeatGAN, AnoBeat. It is relevant to stress out that this level of interpretability can be realised only if the model robustly outputs heartbeat from the normal distribution $p_r$. In this way, an anomalous heartbeat can be compared only with its "nearest" normal counterpart and the time ticks abnormality score is effectively related to deviation from a normal behaviour. For this reason, using a generative model should help. On the other hand, a simpler reconstruction model, such as the autoencoder, when asked to reconstruct an unseen heartbeat will output a nosier and distorted version of a normal heartbeat.

### 3.1.1 Model implementation

The three elements which constitute the architecture of the model depicted in 3.1, are function parametrized by neural networks. The encoder and the decoder use a convolutional neural network[4] (CNN), while the discriminator adopts a fully connected structure. The choice of convolutional structure follows the one made by previous researches which shows how it

is proficient to find useful features on the ECG time series. It is tempting to use a Recurrent Neural Network[76] by observing that the nature of the ECG data is time dependent. Instead, the CNN learns time-independent features, by sliding adjustable filters along the input. However, the RNN showed to suffer from the vanishing gradient problem, making it not useful for capture long-term dependencies. The recurrent networks derived from the RNNs ,such as LSTM, solves that problem but they are more difficult to optimize. The CNN optimally structured showed good performance, and also to capture long-term dependencies. The layer's structure follows the one of the BeatGAN[85] and AnoBeat[60], which in their turn take inspiration from the DCGAN[64], which is a state-of-the-art deep generative model which make use of CNN in a GAN framework. The encoder network is composed by 5 layer CNN for feature extraction alternating with layers of batch normalisation to reduce gradient explosion or vanishing problem[35] and non-linearity. Then, two different neural networks layers maps the extracted features in the mean and the variance of a multivariate Gaussian of $n_h$ dimensions, i.e. the latent space dimension. The latent code $\mathbf{h} \in \mathbb{R}^{n_h}$ is then chained to the one-hot encoded sex information[1] $\mathbf{s}$, generating a 8-dimensional code which serves as input for the decoder to reconstruct the original heartbeat. The decoder has the same structure of the encoder, but symmetrical. The CNN layers reduce the length of the input signal, so they are useful for feature extraction. However, the decoder has to invert the process by reconstructing the signal starting from the latent code $\mathbf{h}$ chained to the sex information. For this purpose, the transposed convolutional[23] layers are used.

## 3.2 Experiment

This section will discuss how the experiment was conducted. In particular we will see how the data was processed and organized and how the hyperparameters were set at test time.

### 3.2.1 MIT-BIH arrhythmia database

The proposed model is evalutated on the MIT-BIH arrhythmia database [56]. The dataset was realised by selecting a sample of 48 two leads ECG

[1]Three possible codes: $\mathbf{s}$ = [1,0,0] for female, $\mathbf{s}$=[0,1,0] for male, $\mathbf{s}$ = [0,0,1] for unknown or missing

records from an original set of over 4000 recordings obtained by the Beth Israel Hospital Arrhythmia Laboratory between 1975 and 1979. The selected records comes from patients with ages between 23 and 89, and both sexes in a balanced way. Each record was originally made using nine Del Mar Avionics model 445 two-channel recorders, then the analog output was digitized at 360 Hz using hardware constructed at the MIT Biomedical Engineering Center and at the BIH Biomedical Engineering Laboratory. They are at least 30 minutes long. The database was constructed in order to cover the most uncommon heart disorders such as complex ventricular, junctional, and supraventricular arrhythmias and conduction abnormalities. The leads configuration is not uniform for all records. In most of them the first lead is the modified lead II (MLII), and the second is the V1 lead. Some of them uses V2 , V4 or V5 as second lead. In one instance the first lead is not the MLII. The record labels were developed and adjusted during years, thanks to several cardiologist who worked independently. The records contain two type of annotations: one regarding the rhythm type, the other for the heartbeat type. The first kind of annotation is placed at the beginning of a heart rhythm. The second is located on each R peak and classifies the heartbeat in one of the sixteen categories. A list of all beat type is presented in table 3.1. The Association for the Advancement of Medical Instrumentation suggested a different notation, incorporating different beats in bigger classes [2].

### 3.2.2 Dataset

The dataset used for the experiment is extracted from the MIT-BIH database. The non-homogeneity of the lead configurations brought the choice of using only the MLII lead for the evaluation of the algorithms. Hence, the records 102 and 104 were discarded because they lack the MLII lead. This choice could be regarded as very restrictive, but the leads has relevant morphological differences and the distribution of leads configuration is very unbalanced towards the MLII-V5 one 3.2. Hence, the choice avoids that the least common configurations, which are not sufficiently represented in the train set, compromise the results. The ECG signals exctracted by the MLII lead are then filtered and segmented. As denoising strategy we use a combination of two Wavelet filters. The wavelet transform is ideal to process dynamic time series, such as the ECG, because it can decompose the signal with different resolutions in time and frequencies, unlike a classical Fourier band-pass filter. The mother wavelet is a function $\Psi(x) \in L^2(\mathbb{R})$, hence it must satisfy

| AAMI Type | Annotation | Description |
|---|---|---|
| N | N | Normal beat |
| | L | Left bundle branch block |
| | R | Right bundle branch block |
| | e | Atrial escape beat |
| | j | Nodal (junctional) escape beat |
| A | A | Atrial premature beat |
| | a | Aberrated atrial premature beat |
| | S | supraventricular premature beat |
| | J | Nodal (junctional) premature beat |
| V | V | Premature ventricular contraction |
| | E | Ventricular escape beat |
| | ! | Ventricular flutter wave |
| F | F | Fusion of ventricular and normal beat |
| Q | Q | Unclassifiable beat |
| | / | Paced beat |
| | f | Fusion of paced and normal beat |

Table 3.1

$$\int_{\mathbb{R}} \Psi^2(x)dx < \infty. \tag{3.22}$$

It is called othonormal wavelet if it provides a complete orthonormal system in $L^2(\mathbb{R})$. The Hilbert basis is composed by dyadic translation and dilation of $\Psi$. The wavelet filter transform the original signal by computing the detail coefficients obtained by convolving stretched versions of the mother wavelet with the signal

$$\psi_{ik} = \int x(t) \frac{1}{\sqrt{2^i}\Psi\left(\frac{t-k2^i}{2^i}\right)dt}. \tag{3.23}$$

The coefficient computed are thresholded and then used to reconstruct the filtered signal by the inverse transform. In a first instance, we remove the baseline wander produced by patient movement and respiration with a robust wavelet filter proposed by Sargolzaei et al[73]. Secondly, a wavelet filter with a Symmlet mother wavelet is used to remove the noise originated from other sources, such as baseline inferences. The Symmlet wavelet has a shape which resembles the QRS complex morphology, hence according to [45] and [65] is one of the best mother wavelet for denoising the ECG signal. This filter decomposes the signal in the maximum number of levels, then

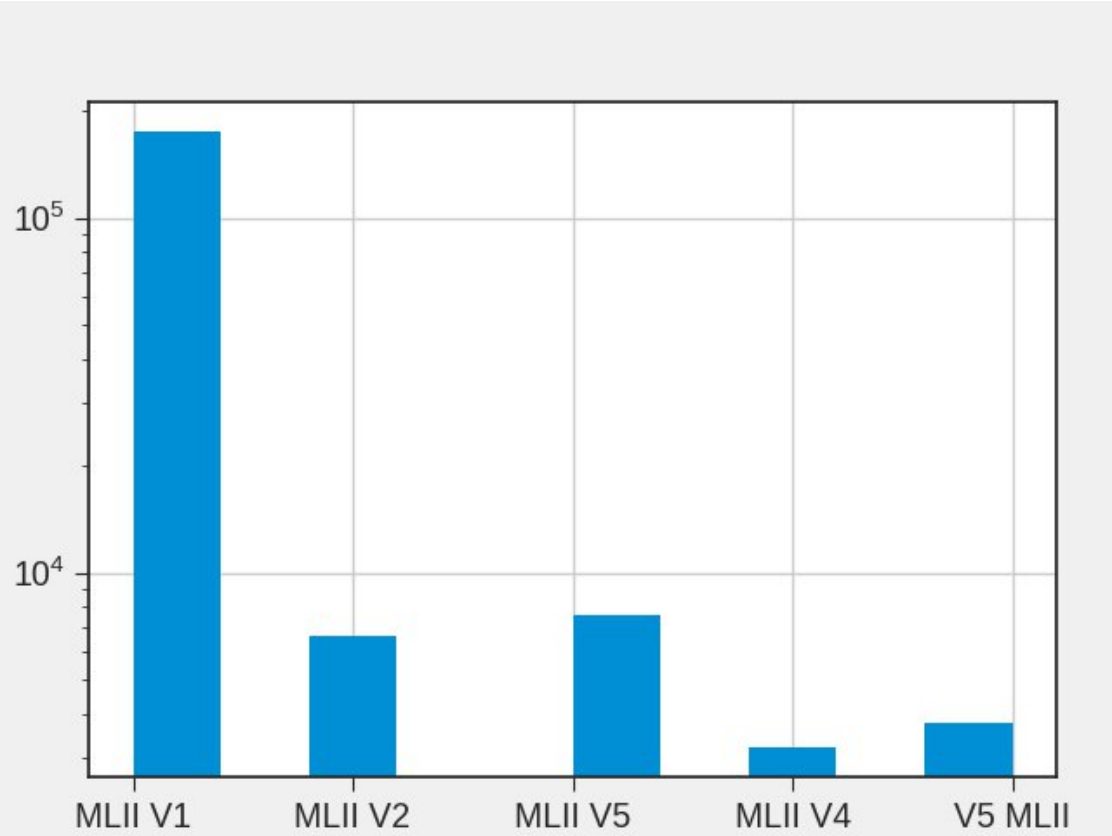



Figure 3.2: The lead configurations are unbalanced

applies a soft threshold $D^S(d|\lambda)$

$$D^S(d|\lambda) = \begin{cases} 0, & \text{for } |d| \leq \lambda \\ d - \lambda, & \text{for } d > \lambda \\ d + \lambda, & \text{for } d < -\lambda \end{cases} \tag{3.24}$$

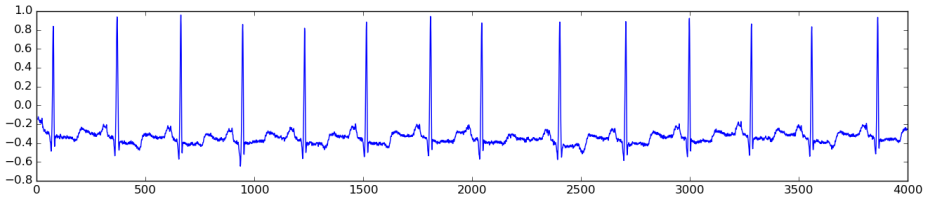
with $\lambda = 0.04$.
The signal filtered is then segmented by taking a fixed window around the R-peak annotation. The window dimension takes into account the typical duration of the cardiac cycle which is about 0.8 seconds and the slight shorter duration of the P wave respect to the T wave. Therefore the window size select 130 time ticks before the R peak and 150 time ticks after which are respectively 0.37s and 0.42s with a sampling frequency of 360 Hz. The choice follows the one adopted by AnoBeat experiment. Finally each heartbeat is rescaled in order to belong in the interval -1 and 1 by

$$\hat{x}_t = \frac{x_t}{M}, \tag{3.25}$$

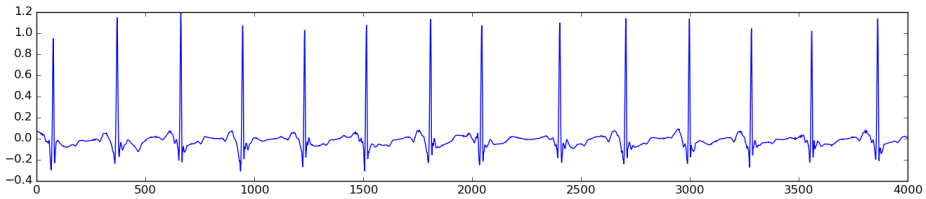
where $M = \max |x_t|$.
At the end of the pre-processing stage, we obtain a dataset of 97,568 labelled heartbeats, 280 time ticks long for a total of 27 million time ticks. In table 3.2 we can see the high unbalance between the AAMI classes.

| Label | **Normal** | **Abnormal** | | | | **Total** |
|---|---|---|---|---|---|---|
| AAMI | **N** | **A** | **V** | **F** | **Q** | |
| # | 87143 | 2781 | 8388 | 757 | 15 | 98221 |
| **Total** | 87143 | 11078 | | | | |

Table 3.2: The majority (88%) of the heartbeats are normal.




(a) Original




(b) Filtered

Figure 3.3: Filtering process

### 3.2.3 Experimental settings

The encoder network is 5 layer CNN of (32,4,2,1) / (64,4,2,1) /(128,4,2,1) / (256,3,2,0) / (512,3,2,0),which means (output size, kernel size, stride, padding), plus two equal final layers, one for the mean and one for the variance of / (5,8,1,0). The decoder is the same as the encoder but symmetrical. The discriminator is a fully connected multi-layer perceptron of three 32 neurons layer. The parameters of the objective function 3.13 are set as $\lambda_{REC} = 1, \lambda_{TV} = 0.001, \lambda_{GP} = 10, \lambda_{ADV} = 1$.The latent space's dimension is fixed at $n_h = 5$ and we impose a Gaussian with 0 mean and identity covariance on it. The network parameters are optimized using the Adam optimizer [24] with $\beta_1 = 0$, $\beta_2 = 0.999$, mini-batch size of $M = 256$,

and the number of critic iteration is $n_{critic} = 5$. The initial learning rate is set to $lr = 0.001$, then it was dynamically adjusted during the training following increments of the PR-AUC over the validation set. When this metric was stagnated for at least 10 epochs, the learning rate decreased by a factor of 10. All the convolutional layers weight are initialized with random normal samples $\psi \sim N(0, 0.02)$ with zero mean and $\sigma^2 = 0.2$. The batch-norm layers weight are initialized by $\psi \sim N(1, 0.02)$ and bias to zero. The codes are implemented on Python 3.8.0 with PyTorch 1.8 and executed on NVIDIA QUADRO K620 GPU. Given the limited computational resources, It was not possible to make an extensive hyper-parameters tuning, hence the results presented in the next chapter could be sub-optimal.

# Chapter 4

# Results

## 4.1 Baseline models

We compare the performance of the proposed model with other simple machine learning algorithm or more deeper architectures already successful in the semi-supervised anomaly detection framework.

### 4.1.1 Principal Component Analysis

We have already discussed the uses of PCA in a semi-supervised anomaly detection problem. The implementation is expressed in the following steps:

- Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{n, L \times N}$ where n is the number of heartbeats, L and N the leads number and the length respectively, be the train set containing only normal heartbeats $\mathbf{x}_i$. We normalize each heartbeat by subtracting the mean and dividing element-wise by the standard deviation
$$\mathbf{x}_{sc} = \frac{(\mathbf{x}_i - \boldsymbol{\mu})}{\mathbf{sd}}, \tag{4.1}$$
where $\boldsymbol{\mu}$ is the mean of each column of $\mathbf{X}$ and $\mathbf{sd}$ its standard deviation. Call $\mathbf{X}_{sc}$ the new scaled dataset.

- We find the linear and orthogonal transformation $\mathbf{W}$ such that
$$\mathbf{W} = \arg\min_{\mathbf{W}} \left(\mathbf{X}_{sc} - \mathbf{W}^T\mathbf{W}\mathbf{X}_{sc}\right)^2, \tag{4.2}$$
with $\mathbf{W} \in \mathbb{R}^{L \times N, n}$ be the orthogonal matrix of the covariance matrix eigenvectors.

- We select the first $\hat{n}$ eigenvectors, such that the accounted variance is at least 95% of the total observations variability.
- The anomaly score for a new observation $\mathbf{x}_0$ is computed as

$$A(\mathbf{x}_0) = (\mathbf{x}_0 - \hat{\mathbf{x}}_0)^2, \tag{4.3}$$

  with $\hat{\mathbf{x}}_0 = \left(\mathbf{W}^T\mathbf{W}\mathbf{x}_{sc} + \boldsymbol{\mu}\right) \odot \mathbf{sd}$, where $\odot$ denotes the element-wise product between two vector and $\mathbf{x}_{sc}$ is the scaled $\mathbf{x}_0$ obtained by deleting $\boldsymbol{\mu}$ and divide by the standard deviation $\mathbf{sd}$.

The PCA performances offer a interesting point of view to the problem. By comparing them to the other models we can understand if the linear transformation offered by the PCA is sufficient for grasp the ECG signal complexity. The other models uses non-linear transformations which has to be justified by better performances.

### 4.1.2 Beat-AutoEncoder

The Auto-encoder is a step above the PCA in capacity. As we have seen, its objective is similar but uses non-linear transformations

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi}} (\mathbf{X} - D_{\boldsymbol{\theta}} \circ E_{\boldsymbol{\phi}}(\mathbf{X}))^2, \tag{4.4}$$

where $E_{\boldsymbol{\phi}} : \mathbb{R}^{N\times L} \mapsto \mathbb{R}^{n_h}$ is the encoder and $D_{\boldsymbol{\theta}} : \mathbb{R}^{n_h} \mapsto \mathbb{R}^{N\times L}$ the decoder. The encoder and decoder networks are implemented as in our model, with the difference that the encoder is deterministic and the latent space dimension in $n_h = 50$, following the choice made by Bin Zhou [85]. The autoencoder performances can show the impact on the predictive power of the latent space regularization used by the proposed model.
The model is trained until convergence which occurs at about 8 epochs. The model is optimized with Adam with $\beta_1 = 0.5, \beta_2 = 0.999$ and learning rate $lr = 0.0001$.

### 4.1.3 Beat-Fast anoGAN

The fast AnoGAN framework was briefly discussed in the last section of the second chapter. It is a state-of-the-art model in semi-supervised anomaly detection. The original implementation operates with image data, hence we provide a slightly different implementation were the encoder, decoder and discriminator networks use convolutional networks for 1-dimensional

signals.The structure of the encoder and the decoder are the same of our model, apart for the deterministic encoder. The discriminator follows the encoder structure, but the last layer is used to output the real/fake score instead of the hidden code. The training consists in two steps: the WGAN training and the Encoder training. In the first step, the WGAN is trained to replicate the normal heartbeats distribution by generating heartbeats from the latent space assumed to be a five-dimensional uniform distribution in the box $[-1,1]^5$. The model was trained until convergence which occurs at about 40 epochs. The learning rate was fized at $lr = 0.00002$ and an Adam optimizer with $\beta_1 = 0, \beta_2 = 0.999$ was used. In the second step, with the generator and discriminator fixed, the encoder learns to map the training data in the latent space by minimizing

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \lambda_{REC}|\mathbf{x} - \hat{\mathbf{x}}| + \lambda_{ADV}|f_\omega(\mathbf{x}) - f_\omega(\hat{\mathbf{x}})|, \tag{4.5}$$

where $f_\omega(\mathbf{x})$ is the the second-to-last layer function of the discriminator. The same loss is used as anomaly score.
At training time $lambda_{REC} = 1$ and $lambda_{ADV} = 0.1$. The encoder is optimized using Adam with $\beta_1 = 0.5, \beta_2 = 0.999$ and $lr = 0.0001$.

### 4.1.4 AnoBeat

The model is implemented as described in the paper. The AnoBeat is the state-of-the-art model specified for semi-supervised anomalous heartbeats detector. We already described it previously, therefore we proceed with the implementation details. The network was implemented following step by step the description provided in the original paper. It is composed by four networks. The Encoder is a six-layer 1-D convolutional neural network with structure (32,4,2,1) / (64,4,2,1) /(128,4,2,1) / (256,3,2,0) / (512,3,2,0) / (50,8,1,0),which means (output size, kernel size, stride, padding). Hence, the structure is similar to our proposed neural network, but it is deterministic and the hidden code size is $n_h = 50$. The decoder has the same structure but symmetrical, using transposed convolutional layers. Each convolutional layer are alternated with batch normalization and leaky ReLU with 0.2 leak activation function for the encoder, and ReLU for the decoder. The discriminator between heartbeats has the same structure as the encoder apart last layer whose output is of 1 dimension with a Sigmoid activation function instead of the latent code $n_h$. The latent discriminator is also convolutional with (4,4,2,0) / (8,4,2,1)/ (16,4,2,1), and a fully connected last layer with scalar output. The networks are optimized with Adam with

$\beta_1 = 0.5, \beta_2 = 0.999, lr = 0.0001$. All the convolutional layers weight are initialized with random normal samples $\psi \sim N(0, 0.02)$ with zero mean and $\sigma^2 = 0.2$. The batchnorm layer weights are initialized by $\psi \sim N(1, 0.02)$ and bias to zero. Despite the attempt to reproduce the exact experiment made by Yingzi Ou et al., the results given in next section does not match the ones obtained by their paper.

## 4.2 Evaluation metrics

The metrics used to evaluate the models take into account the priorities imposed in a real case scenario. In this case, we are far more interested in spot the anomalous heartbeats than the normal. The former could imply life-threatening conditions for the patient, hence we want to maximize the true positive rate (TPR) of the abnormal heartbeat class. Therefore, a metric such as the accuracy is not of our interest because is defined as

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}, \tag{4.6}$$

where True Positives (TP) is the total number of correctly classified abnormal heartbeats, True Negatives (TN) is the number of correctly classified normal heartbeats, and False Positives (FP) and False Negatives (FN) are the ones incorrectly classified. We instead focus our attention on the true positive rate or often called recall

$$R = \frac{TP}{FN + TP} = \mathbb{P}(\hat{X} = 1 | X = 1), \tag{4.7}$$

which is the probability that an observation regarded positive is positive. Obviously, a model that says every heartbeat is abnormal would achieve perfect recall, hence we need to take into account also the precision

$$P = \frac{TP}{TP + FP} = \mathbb{P}(X = 1 | \hat{X} = 1), \tag{4.8}$$

defined as the probability that a positive observation is predicted as positive. Sometimes it is useful to summarize both metric in a single number. The $F_\beta-$score was derived by by van Rijsbergen (1979)[48] and it is defined as the harmonic mean between the two measures

$$F_\beta = (1 + \beta^2) \frac{P \cdot R}{\beta^2 P + R}, \tag{4.9}$$

where the $\beta$ parameter adjust the importance of the recall with respect to the precision. For instance, $\beta = 2$ means that we regards the recall two times more important than the precision. In our case study, it is reasonable to sacrifice some precision in favour of recall. It is preferable to have a false alarm than no alarm at all. Hence, we will compare our model using the $F_2-$score.
All the measures discussed so far are based on a class prediction. In other words, they expect the model to output a label, e.g. 1 for the positive class, anomalous heartbeats in our case, 0 for the negative class. However, the models under examination output an anomaly score which is a positive continuous value whose magnitude express the abnormality of the observation. In order to obtain a label prediction we have to threshold the anomaly score with a value that has to be computed apart or estimated on a validation set composed by both positive and negative instances which has to be different to the test set. However, we would like to compare the models' performance regardless of the chosen threshold. For this purpose, the area under the precision recall curve (PR-AUC) is helpful. Each point of this curve represents the precision vs recall obtained by the model with a different threshold. A PR-AUC of 1 denotes the best possible model which assigns an anomaly score of 0 at negative instances and an infinite score to positive instances. On the other hand, a PR-AUC of 0.5 denotes random guess model which assign positve or negative labels with equal probability. The PR-AUC is indicated when the positive class is more relevant than the negative since it relates two measures related to the positive class. However, the precision depends on the positive label frequency of the observed population. From Bayes Theorem we have

$$\mathbb{P}(X = 1|\hat{X} = 1) = \frac{\mathbb{P}(\hat{X} = 1|X = 1)\mathbb{P}(X = 1)}{\mathbb{P}(\hat{X} = 1)}, \tag{4.10}$$

which shows that the precision vary if a different population is used. The area under the receiver operating characteristic curve (ROC-AUC) overcome this problem by relating the recall versus the specificity

$$S = \frac{TN}{TN + FP} = \mathbb{P}(\hat{X} = 0|X = 0), \tag{4.11}$$

which are both conditioned to the relative frequencies of the positive class $\mathbb{P}(X = 1)$ and the negative $\mathbb{P}(X = 0)$. Therefore, also the ROC-AUC will be observed.

## 4.3 Result discussion

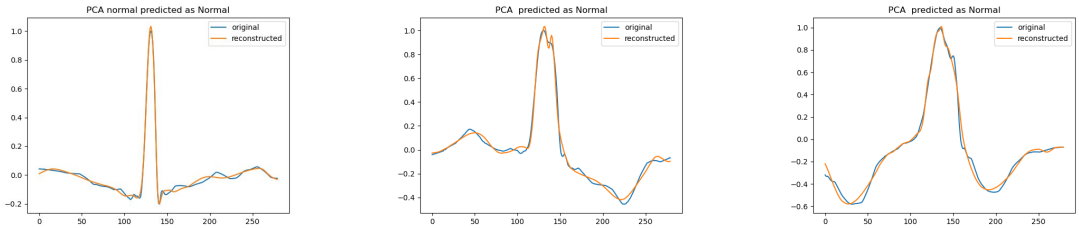
The models are evaluated on the dataset excrated from the MIT-BIH database previously described, with a 5-fold cross validation fashion.Therefore, we divide the normal heartbeats in 5 sets. At the i-th evaluation the i-th set is used for testing together with the abnormal heartbeats, while the other normal heartbeats are used for training. Moreover, for each iteration we generate an additional evaluation set to tracking the models performances and computing the threshold that will be used for the $f_2$-score in testing. In each fold, the validation set contains 5% of normal instances from the train and 10% of abnormal instances from test set. The results are tabulated in 4.1, reporting the mean values and the standard deviations.

Table 4.1: Results. In bold the best ones for each metric. Beat-AE and MyModel have comparable performances for the F2-score due to the overlapping standard deviations.

| **Model** | **PR-AUC** | **ROC-AUC** | **F2-SCORE** |
|---|---|---|---|
| PCA | 0.7026 ± 0.0048 | 0.8352 ± 0.0020 | 0.5324 ± 0.0041 |
| Beat-AE | 0.8731 ± 0.0043 | 0.9027 ± 0.0038 | **0.6685 ± 0.0275** |
| Beat-Fast AnoGAN | 0.8304 ± 0.0113 | 0.8865 ± 0.0057 | 0.5517 ± 0.0427 |
| AnoBeat[0] | 0.8799 ± 0.0038 | 0.9151 ± 0.0038 | 0.6084 ± 0.0188 |
| AAECG | **0.9204 ± 0.0077** | **0.9504 ± 0.0045** | **0.6726 ± 0.0149** |

Overall, the models which make use of deep non-linear transformations perform best with respect to the PCA. One could intuitively think that the linearity of the transformation is too simple in order to represent the complex ECG signal. This insight is also proposed by Bin Zhou[85]. However, the signals reconstructed by the PCA are quite good and clean. The bad performances are due to the opposite: the set of latent variables learned by the PCA are able to reconstruct well also abnormal heartbeats 4.1,hence plenty of them are erroneously regarded as normal. The linear transformation limits the model by another perspective which is the shape complexity of the data manifold which it can capture. The PCA can been interpreted as finding an ellipsoid which captures most variability of the data. Given the experiment, this ellipsoidal manifold shape is too restrictive for the ECG data.
Among the model with non-linear transformations, the model based on the fast-AnoGAN framework is the worst. The outcome is not expected, since

[0]The results on the original paper are quite different and comparable with our model: 0.9276±0.0060 for PR-AUC and 0.9596 ± 0.0035 for ROC-AUC




(a) (b) (c)

Figure 4.1: 4.1a is normal heartbeat well reconstructed by the PCA. 4.1b and 4.1c are abnormal heartbeats over-reconstructed by the PCA and predicted as normal instances.

the anoGAN model is quite similar to our proposed framework. They are both based on a deep generative model, but the anoGAN minimizes the wasserstain distance between the data distribution $p_r$ and the generated $p_g$ while our model minimizes the mean squared error which, as we have seen previously, is equivalent to maximize the expected log-likelihood or minimize the Kullback-Leibler diveregence. Furthermore, our model assumes that the data distribution conditioned to the latent variables is Gaussian with identity covariance matrix. The problem of the anoGAN is that it is not robust to noise in the ECG signal. The training was unstable and the model ends up to learn also unwanted noise 4.2. The sample generated by the GAN are noisy, and this results in bad reconstructions even for normal heartbeats. The samples generated by our model are instead much more clean 4.3, confirming the importance of the total variation penalty.
The advantage of using a generative model, is that the reconstructed heartbeats are always sampled from the normal heartbeats manifold. Hence, the reconstruction error is obtained with respect to the "nearest" normal heartbeat. In other word, we avoid the over-reconstruction problem observed in the case of PCA. This is particularly true for the fast-AnoGAN framework if we force the encoder to output only in the n-dimensional box $[-1,1]^{n_h}$, used as source of input noise during GAN training. The generator can handle input coming from this subset because it has previously seen it during training. Our model is based on an adversarial autoencoder, hence it is less robust from this point of view. The encoder network could have strange behaviour if asked to encode new out-of-distribution data, providing codes which the generator can't handle correctly. However, as we can see in 4.4 our model seems to be robust to such eventualities.

This characteristic let us to build a robust anomaly score for the single

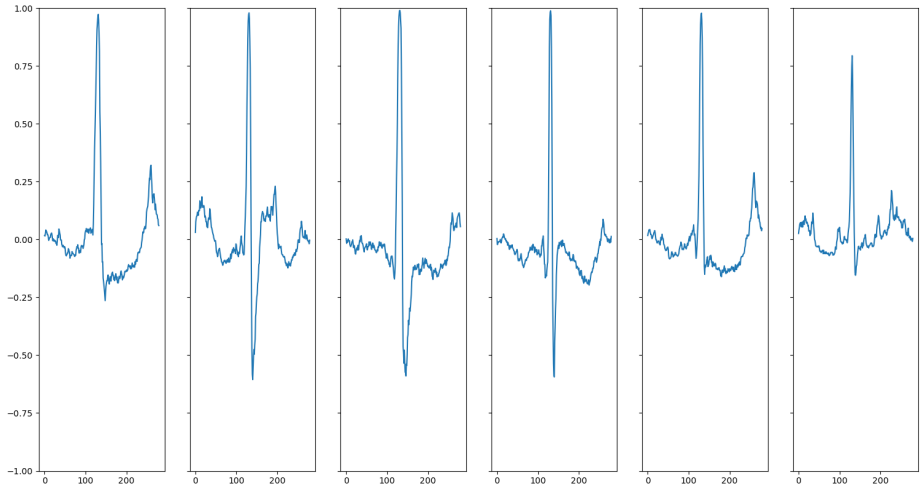



Figure 4.2: The samples from the anoGAN are noisy

time ticks inside the heartbeat. We can compute the squared distance between each time ticks of the original and reconstructed signals and pinpoint the areas where the beat is more anomalous with respect to the normal behaviour. This adds a layer of interpretability to the model, which not

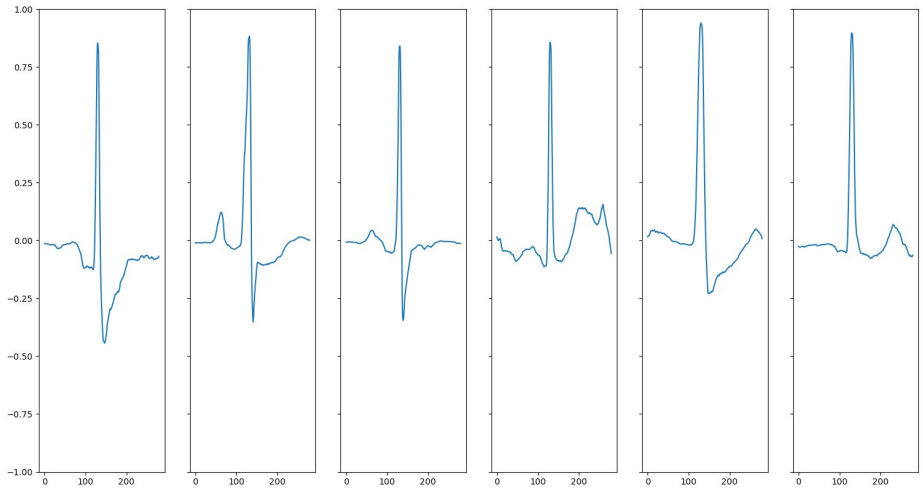



Figure 4.3: Samples from AAECG. The penalty on the total variation has a de-noising effect.

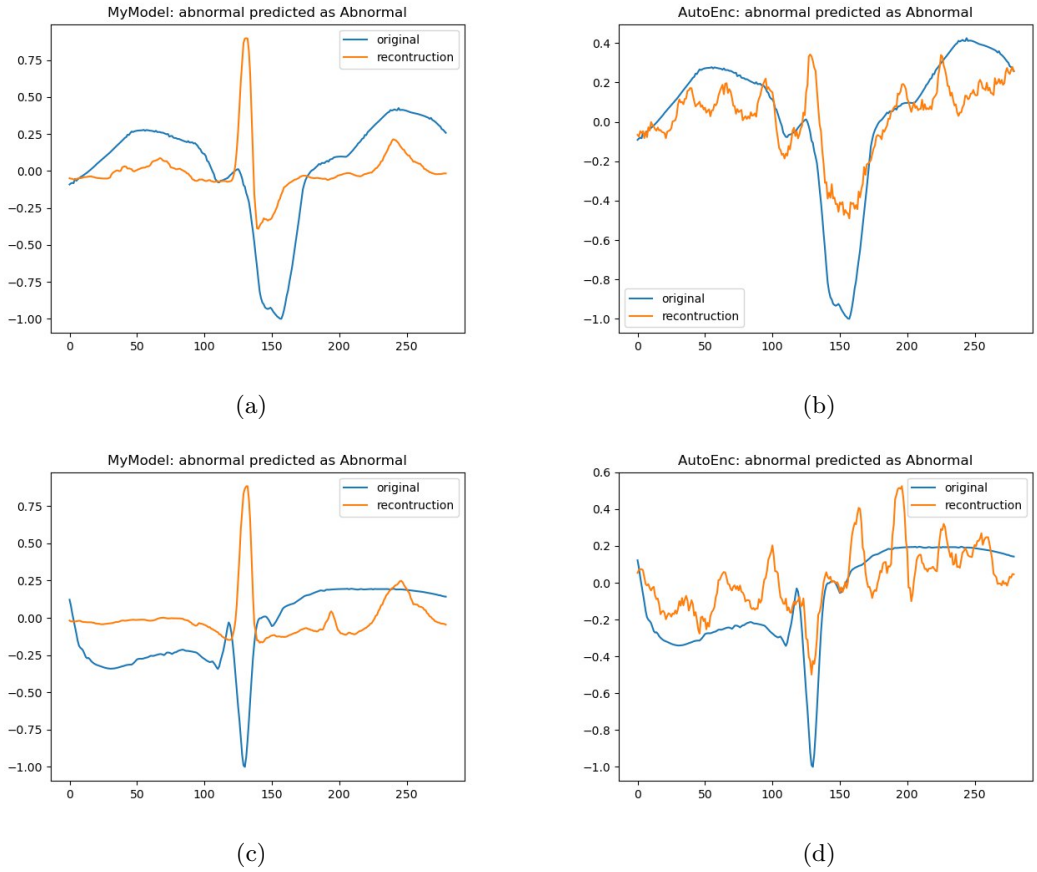



Figure 4.4: 4.4c and 4.4b shows a premature ventricular contraction beat reconstructed by our model (on the left) and by the autoencoder (on the right). 4.4c and 4.4d are the respective reconstructions of an unclassifiable beat. All of them are correctly predicted, but our model is more robust since it does not reconstruct abnormalities.

only tells us whether a beat is abnormal or not, but also gives an indication of the most abnormal regions. As done by BeatGAN and ANoBEAT we contruct a heatmap 4.5 which could help a doctor to focus the attention on a specific heartbeat region.

### 4.3.1 Further analysis on the model

Apparently, the proposed model successfully captured the normal heartbeats variability available in the dataset. The five dimensional hidden code, plus the three dimensional code representing the one-hot encoded patient sex, is able to control the waves morphologies. To observe how the single hidden codes can influence the heartbeat shape, we vary only one latent

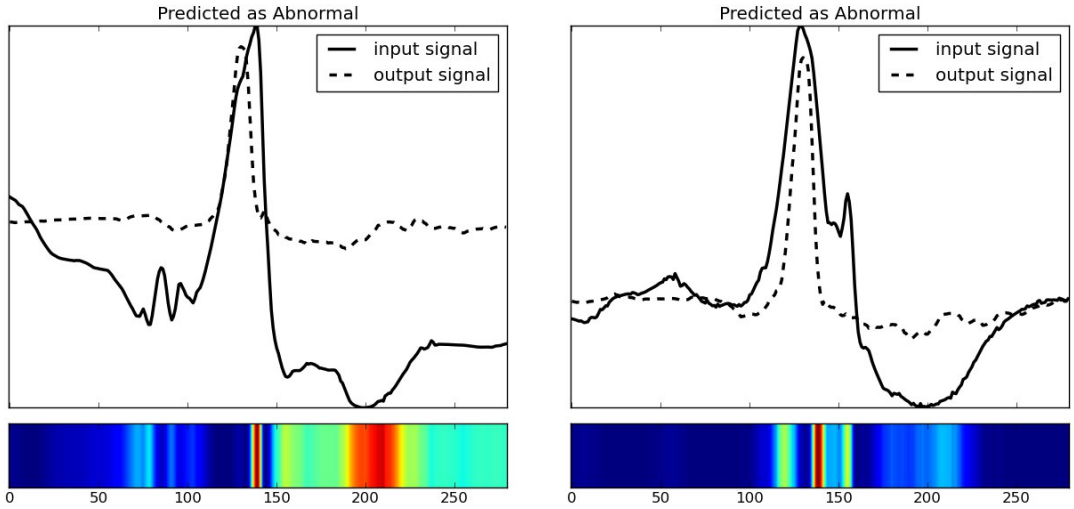
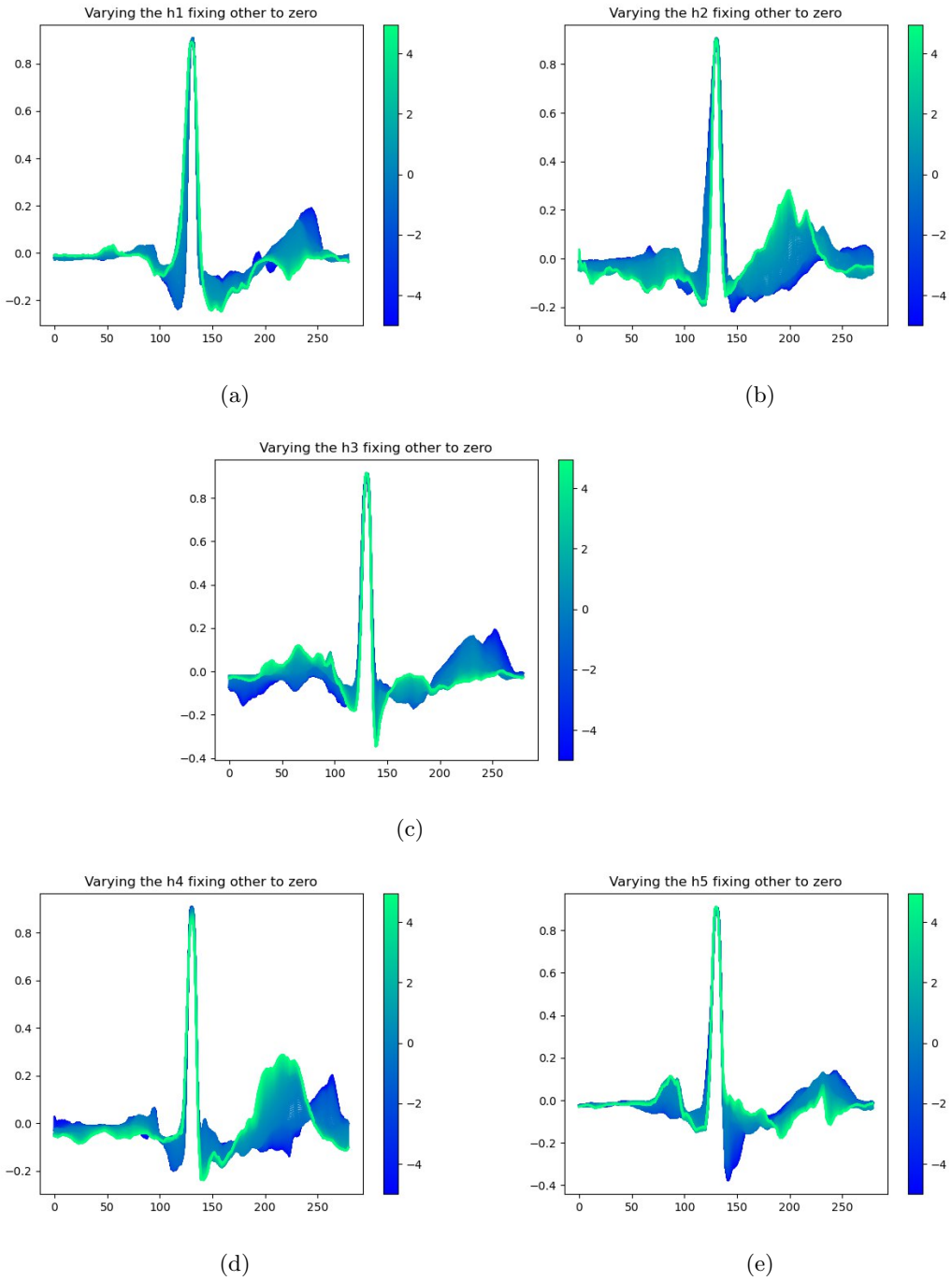



Figure 4.5: Two Abnormal heartbeats correctly identified. We can pinpoint the anomalous regions by comparing each time ticks.

variable at time while keeping the other four and the sex code fixed. In this way we can see if there are some similar semantics with the human understanding of Sinus heartbeat. Figure 4.6 shows how the beat shape is affected by varying only one hidden variable in the interval $[-5,5]$, while the other are keep fixed to zero. None of them represent directly a human-readable semantic. The fifth latent variable seems to control the amplitude of the P-wave and of the S wave. Decreasing it towards -5 makes the P-wave smaller and the S wave more prominent. The third latent variable increases the T wave amplitude while making the S wave flatter. However, their interpretation is very difficult for a human, also taking into account that the model uses all possible interactions between them. Picture 4.9 shows one of them.

The model seems also to exploit the additional sex information added as input to the generator. To analyze its influence, we generate several couple of heartbeats obtained by the same latent code but different sex information. Six of them are reported in 4.7. We can observe some differences in the P and T wave-form. Obviously, the patient variability observed by the model is too small in order to generalize possible sex differences found.

However, if in future the model will be trained on a dataset that contains larger patient variability, it could discover important ECG differences.This open the possibility of more patient specific diagnosis, which takes into account additional information other than the simple ECG wave-forms.




(a) (b) (c) (d) (e)

Figure 4.6: We investigate the latent variable effects on the heartbeat morphology, by varying one of them in the interval [-5, 5] and keeping fixed the sex and other latent variables. In green and dark blue we can observe the beat generated with the latent variable value of respectively 5 and -5. The representation learned by the model is quite entangled and difficult to understand.

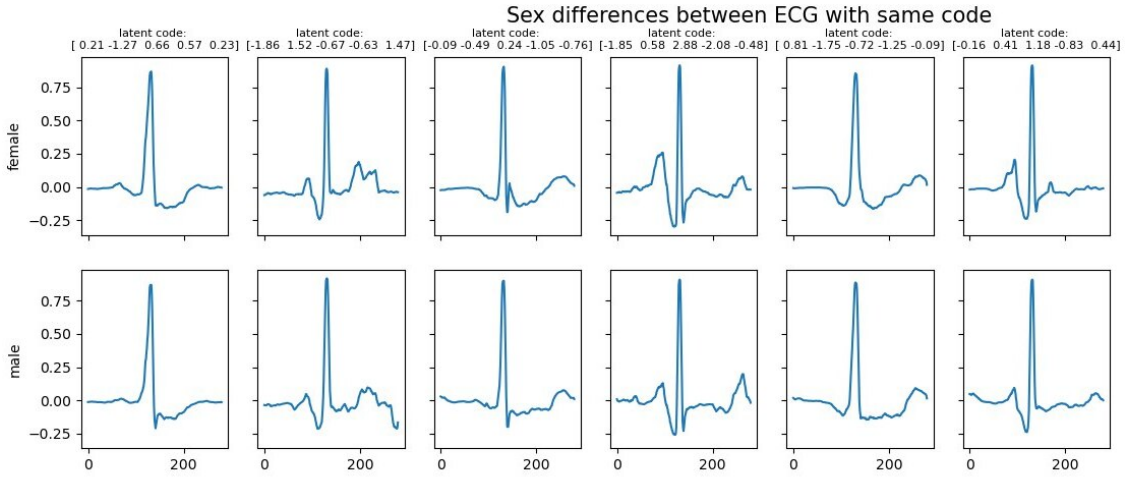



Figure 4.7: Each column shows two heartbeats generated from the same latent variable values, with different sex information. It seems to influence the shape and amplitude of the P and T waves. Obviously, the patients' statistical sample observed (48) is too small to generalize the discrepancies observed.

Concluding the analysis, we point out an interesting phenomena which involves how the encoder reacts to out-of-distribution input, i.e. the abnormal heartbeats. Remember that the inference network, or encoder, is stochastic, indeed its output determines the mean and the variance of a multivariate Gaussian distribution. While the mean indicates the most likely hidden code which has generated the given heartbeat, we expect that the variance expresses the uncertainty on the prediction. A straightforward insight is that when the input has different property than expected the uncertainty on the estimate is high. To observe if this property is realistically respected by the model, we plot the variance $\sigma^2$ against the anomaly score of 2000 normal heartbeats and 2000 abnormal heartbeats 4.8.

Since the variance takes values of different scales of magnitude, we plot the $log(1+\sigma^2)$. The regression line has a positive coefficient of $21.40 \pm 0.306$ statistically significant different from 0 (p-value $< 0.0001$). However, the $R^2 = 0.435$ statistic is low and the correlation coefficient between the two variables is about 0.4. Therefore, heartbeats with high anomaly score seems to produce more uncertainty in the inference of the encoder network, but this correlation is slight.

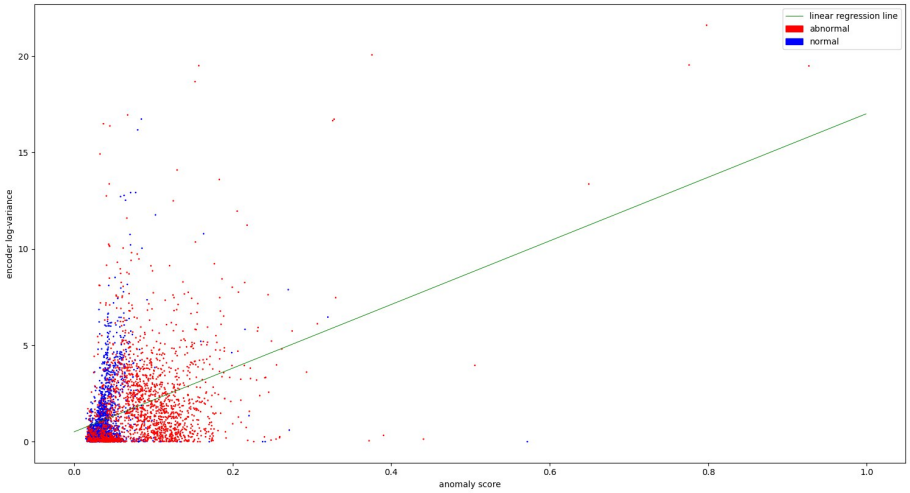



Figure 4.8: 2000 normal and abnormal heartbeats are used to analyze the correlation between the anomaly score, as a proxy of unexpected heartbeat shape, and $log(1 + \sigma^2)$, where $\sigma^2$ is the variance of the inference network. The correlation is low, 0.4, the regression line has a slope of $21.40 \pm 0.306$. There is a slight correlation, but it does not confirm the straightforward insight that the encoder inference variance grows as the input heartbeat shape is more unexpected.

## 4.4 Thesis conclusions

In this thesis we have outlined the importance of computer-aided electrocardiography, especially for analyze long ECG records or continuously monitoring patient in intensive care. We have showed that a crucial step for most of ECG anomaly detection algorithm is identifying irregular heartbeats in the recordings. The literature offers mainly classification algorithms, which can operate only on a restricted number of anomalies. However, the type of heartbeat abnormalities are plenty and some of them are rare. In practice is not feasible to generate a balanced dataset containing all possible anomalies. Therefore, we tackled the problem in a semi-supervised anomaly detection fashion, where the data used during the training process comes only from normal instances which are abundant and easy to obtain. The model learns the normal data variability, and use this knowledge to asses if a new heartbeat is normal or not. From this point of view, the problem can be rephrased with a statistical flavour, assuming the existence of a $p_{normal}(\mathbf{x})$ distribution of normal data. Then, we have to estimate it given

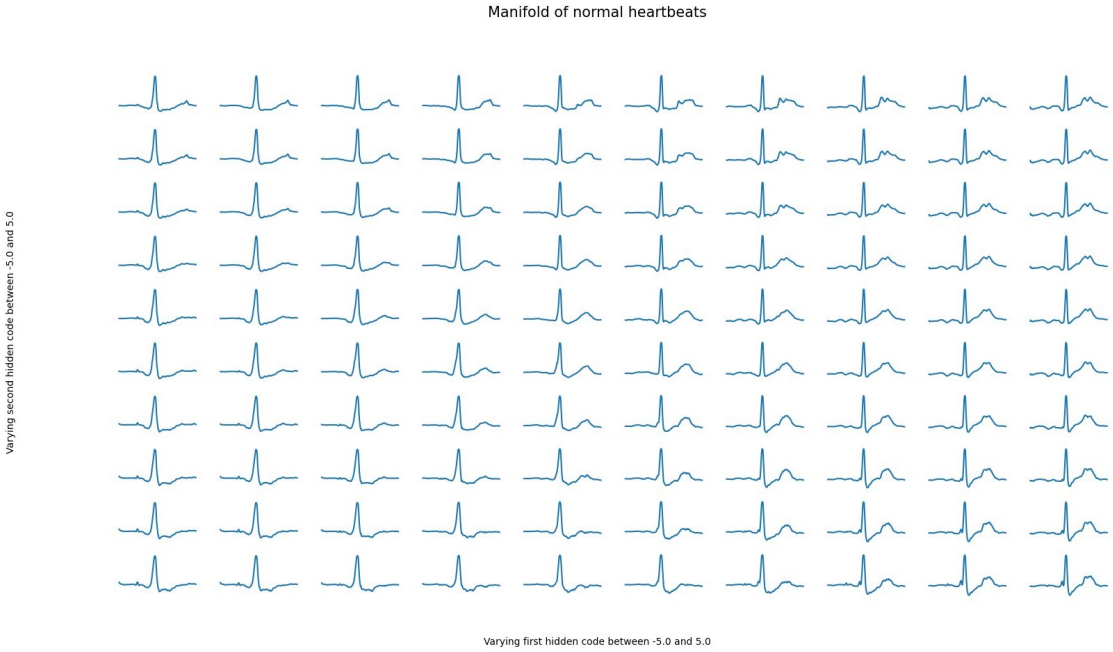



Figure 4.9: The grid is obtained by fixing the last three latent variables to 0, and varying the first two between [-5,5].

the set of observations $(\mathbf{x}_1, \ldots, \mathbf{x}_N) \sim p_{normal}$, and asses if a new heartbeat $\mathbf{x}_0$ is an out-of-distribution sample or not. Given the high dimensional data, we choose a deep generative model to address the problem. In particular, a model derived by the adversarial autoencoder is chosen. This model has been shown to be able to replicate the distribution of normal heartbeats, by using also the additional patient sex information from which the ECG was recorded. The proposed framework compared to other baselines reached state-of-the-art performances on the MIT-BIH database. Moreover, It showed to be capable of sampling good-looking heartbeats, thanks also to the total variation penalty added to the objective function.

### 4.4.1 Limitations and future works

The model is intended to monitor continuously patients in intensive care, alarming doctor only if a heartbeat is detected as abnormal. Given the low performances for the f2-score and hence the rate positive predicted abnormalities, this model is far from a real applicability. However, we have to point out that only one lead was used for the evaluation, some abnormalities are more visible on the other record. Moreover, the results obtained could be too optimistic: the train and the test set contains heartbeats which could potentially come from the same patient. In this way we are not taking into account the high patients variability in the ECG records[43]. This paradigm could be useful if we want to monitor patients from the same group, but it is not generalisable. Some studies evaluate the models by taking also in account the difference across the patients. However, the problem is in the MIT-BIH database which contains records from only 48 patients. Therefore, the statistical sample is too restricted to evaluate generalization across patients even if we use an inter-patient paradigm to construct train and test sets.
Nonetheless, the model shows to be capable of understanding the sex differences between heartbeats. If in future will be assembled a richer dataset, covering also other general information, this approach could let to study the effect of some conditions, drugs or other personal details to the Sinus heartbeat wave-form, opening a path towards more patient-specific diagnosis.

# Bibliography


[1] Harold P. Adams et al. "Guidelines for the Early Management of Adults With Ischemic Stroke". In: *Stroke* 38.5 (2007), pp. 1655–1711. DOI: 10.1161/STROKEAHA.107.181486. eprint: https://www.ahajournals.org/doi/pdf/10.1161/STROKEAHA.107.181486. URL: https://www.ahajournals.org/doi/abs/10.1161/STROKEAHA.107.181486.

[2] A. for the Advancement of Medical Instrumentation et al. *"Testing and reporting performance results of cardiac rhythm and st segment measurement algorithms"*. 1998.

[3] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. "Ganomaly: Semi-supervised anomaly detection via adversarial training". In: *Asian conference on computer vision*. Springer. 2018, pp. 622–637.

[4] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network". In: *2017 International Conference on Engineering and Technology (ICET)*. Ieee. 2017, pp. 1–6.

[5] Joseph Alpert. "Can You Trust a Computer to Read Your Electrocardiogram?" In: *The American journal of medicine* 125 (June 2012), pp. 525–6. DOI: 10.1016/j.amjmed.2012.02.001.

[6] Jinwon An and Sungzoon Cho. "Variational autoencoder based anomaly detection using reconstruction probability". In: *Special Lecture on IE* 2.1 (2015), pp. 1–18.

[7] Martin Arjovsky, Soumith Chintala, and Léon Bottou. *Wasserstein GAN*. 2017. arXiv: 1701.07875 [stat.ML].

[8] Hussein Atoui, Jocelyne Fayn, and Paul Rubel. "A Novel Neural-Network Model for Deriving Standard 12-Lead ECGs From Serial Three-Lead ECGs: Application to Self-Care". In: *Information Technology in Biomedicine, IEEE Transactions on* 14 (2010), pp. 883–890. DOI: 10.1109/TITB.2010.2047754.

[9] Zachi I Attia et al. "Age and sex estimation using artificial intelligence from standard 12-lead ECGs". In: *Circulation: Arrhythmia and Electrophysiology* 12.9 (2019), e007284.

[10] Piotr Augustyniak. "On The Equivalence Of The 12-Lead Ecg And The Vcg Representations Of The Cardiac Electrical Activity". In: (2002).

[11] Giuliano Basso. *A Hitchhiker's guide to Wasserstein distances*. 2015.

[12] SM Bendre. *Outliers in Statistical Data*. 1994.

[13] Martin Bickerton and Alison Pooler. "Misplaced ECG electrodes and the need for continuing training". In: *British Journal of Cardiac Nursing* 14.3 (2019), pp. 123–132. DOI: 10.12968/bjca.2019.14.3.123.

[14] Christopher M Bishop. "Novelty detection and neural network validation". In: *IEE Proceedings-Vision, Image and Signal processing* 141.4 (1994), pp. 217–222.

[15] Antzelevitch C et al. "Brugada syndrome: report of the second consensus conference. Heart Rhythm." In: (2005). DOI: 10.1016/j.hrthm.2005.01.005.

[16] Chandola et al. "Anomaly Detection: A Survey". In: *ACM Comput. Surv.* 41 (July 2009). DOI: 10.1145/1541880.1541882.

[17] Chaudhari et al. "Learning from Positive and Unlabelled Examples Using Maximum Margin Clustering". In: *Neural Information Processing*. Ed. by Huang et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 465–473. ISBN: 978-3-642-34487-9.

[18] Hyunsun Choi, Eric Jang, and Alexander A Alemi. "Waic, but why? generative ensembles for robust anomaly detection". In: *arXiv preprint arXiv:1810.01392* (2018).

[19] Young-Sik Choi. "Least squares one-class support vector machine". In: *Pattern Recognition Letters* 30.13 (2009), pp. 1236–1240.

[20] Luc Devroye. *Non- Uni form Random Variate Generation*. y Springer-Verlag New York Inc, 1986. ISBN: 3-540-96305-7.

[21] Federico Di Mattia et al. "A survey on gans for anomaly detection". In: *arXiv preprint arXiv:1906.11632* (2019).

[22] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. "Adversarial feature learning". In: *arXiv preprint arXiv:1605.09782* (2016).

[23] Vincent Dumoulin and Francesco Visin. "A guide to convolution arithmetic for deep learning". In: *arXiv preprint arXiv:1603.07285* (2016).

[24] Francis Ysidro Edgeworth. "Xli. on discordant observations". In: *The london, edinburgh, and dublin philosophical magazine and journal of science* 23.143 (1887), pp. 364–375.

[25] IPASVI Enna. *L'Elettrocardiogramma*. URL: https://web.archive.org/web/20180428144853/http://www.ipasvienna.it/L'ELETTROCARDIOGRAMMA.pdf. accessed on 30 march 2021.

[26] Scott E Fahlman, Geoffrey E Hinton, and Terrence J Sejnowski. "Massively parallel architectures for Al: NETL, Thistle, and Boltzmann machines". In: *National Conference on Artificial Intelligence, AAAI*. 1983.

[27] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Ed. by The MIT Press Cambridge MA USA. 2016. ISBN: 978-0262035613.

[28] Nico Görnitz et al. "Support Vector Data Descriptions and $k$-Means Clustering: One Class?" In: *IEEE transactions on neural networks and learning systems* 29.9 (2017), pp. 3994–4006.

[29] Ishaan Gulrajani et al. "Improved training of wasserstein gans". In: *arXiv preprint arXiv:1704.00028* (2017).

[30] John Hampton Joanna Hampton. *The ECG Made Easy, 9th Edition*. Elsevier, 2019. ISBN: 9780702074578.

[31] Geoffrey E Hinton. "Boltzmann machine". In: *Scholarpedia* 2.5 (2007), p. 1668.

[32] Geoffrey E Hinton. "Deep belief networks". In: *Scholarpedia* 4.5 (2009), p. 5947.

[33] Geoffrey E Hinton et al. "Learning distributed representations of concepts". In: *Proceedings of the eighth annual conference of the cognitive science society*. Vol. 1. Amherst, MA. 1986, p. 12.

[34] Tadeusz Inglot and Piotr Majerski. "Simple upper and lower bounds for the multivariate Laplace approximation". In: *Journal of Approximation Theory* 186 (2014), pp. 1–11.

[35] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *ArXiv* abs/1502.03167 (2015).

[36] "Gitter M. J. et al. "Variability of different methods for measurement of ECG intervals and ECG interval temporal variation." In: *Journal of electrocardiology, 22 Suppl,* (1989), pp. 125–126. DOI: https://doi.org/10.1016/s0022-0736(07)80110-5.

[37] Dana Mackenzie Judea Pearl. *The Book of Why: The New Science of Cause and Effect*. Ed. by Basic Books; 1° edizione. 2017. ISBN: 046509760X.

[38] Tim C Kietzmann, Patrick McClure, and Nikolaus Kriegeskorte. "Deep Neural Networks in Computational Neuroscience". In: *bioRxiv* (2018). DOI: 10.1101/133504. eprint: https://www.biorxiv.org/content/early/2018/06/05/133504.full.pdf. URL: https://www.biorxiv.org/content/early/2018/06/05/133504.

[39] Diederik P Kingma, Tim Salimans, and Max Welling. "Variational dropout and the local reparameterization trick". In: *arXiv preprint arXiv:1506.02557* (2015).

[40] Kingma et al. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[41] Paul Kligfield et al. "Recommendations for the Standardization and Interpretation of the Electrocardiogram". In: *Journal of the American College of Cardiology* 49.10 (2007), pp. 1109–1127. DOI: 10.1016/j.jacc.2007.01.024. eprint: https://www.jacc.org/doi/pdf/10.1016/j.jacc.2007.01.024. URL: https://www.jacc.org/doi/abs/10.1016/j.jacc.2007.01.024.

[42] Mario Köppen. "The curse of dimensionality". In: *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*. Vol. 1. 2000, pp. 4–8.

[43] Ruggero Donida Labati et al. "Deep-ECG: Convolutional neural networks for ECG biometric recognition". In: *Pattern Recognition Letters* 126 (2019), pp. 78–85.

[44] Hongzu Li and Pierre Boulanger. "A Survey of Heart Anomaly Detection Using Ambulatory Electrocardiogram (ECG)". In: *Sensors* 20.5 (2020). ISSN: 1424-8220. DOI: 10.3390/s20051461. URL: https://www.mdpi.com/1424-8220/20/5/1461.

[45] H-Y Lin et al. "Discrete-wavelet-transform-based noise removal and feature extraction for ECG signals". In: *Irbm* 35.6 (2014), pp. 351–361.

[46] Shuang Liu, Olivier Bousquet, and Kamalika Chaudhuri. "Approximation and convergence properties of generative adversarial learning". In: *arXiv preprint arXiv:1705.08991* (2017).

[47] Willoughby C. Liwanag M. *Atrial Tachycardia.* URL: https://www.ncbi.nlm.nih.gov/books/NBK542235/. accesed on 30 march 2021.

[48] Rachel Tsz-Wai Lo, Ben He, and Iadh Ounis. "Automatically building a stopword list for an information retrieval system". In: *Journal on Digital Information Management: Special Issue on the 5th Dutch-Belgian Information Retrieval Workshop (DIR)*. Vol. 5. 2005, pp. 17–24.

[49] Lars Maaløe et al. "Biva: A very deep hierarchy of latent variables for generative modeling". In: *arXiv preprint arXiv:1902.02102* (2019).

[50] PW Macfarlane et al. "Effects of age, sex, and race on ECG interval measurements". In: *Journal of electrocardiology* 27 (1994), pp. 14–19.

[51] Prasanta Chandra Mahalanobis. "On the generalized distance in statistics". In: National Institute of Science of India. 1936.

[52] Alireza Makhzani et al. *Adversarial Autoencoders*. 2015. arXiv: 1511.05644 [cs.LG].

[53] Xudong Mao et al. "Least squares generative adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2794–2802.

[54] Luke Metz et al. "Unrolled generative adversarial networks". In: *arXiv preprint arXiv:1611.02163* (2016).

[55] Guido Montúfar. "Deep narrow Boltzmann machines are universal approximators". In: *arXiv preprint arXiv:1411.3784* (2014).

[56] George B Moody and Roger G Mark. "The impact of the MIT-BIH arrhythmia database". In: *IEEE Engineering in Medicine and Biology Magazine* 20.3 (2001), pp. 45–50.

[57] Eric Nalisnick et al. "Do deep generative models know what they don't know?" In: *arXiv preprint arXiv:1810.09136* (2018).

[58] Staff Writer Nicole Lou. *Study: STEMI Often Not What It Appears in COVID-19.* URL: https://www.medpagetoday.com/infectiousdisease/covid19/86031. accessed on 30 march 2021.

[59] Keith Noto, Carla Brodley, and Donna Slonim. "FRaC: a feature-modeling approach for semi-supervised and unsupervised anomaly detection". In: *Data mining and knowledge discovery* 25.1 (2012), pp. 109–133.

[60] Yingzi Ou et al. "Anobeat: Anomaly Detection for Electrocardiography Beat Signals". In: July 2020, pp. 142–149. DOI: 10.1109/DSC50466.2020.00029.

[61] J. Pan and W. J. Tompkins. "A Real-Time QRS Detection Algorithm". In: *IEEE Transactions on Biomedical Engineering* BME-32.3 (1985), pp. 230–236. DOI: 10.1109/TBME.1985.325532.

[62] Sejun Park et al. "Minimum width for universal approximation". In: *arXiv preprint arXiv:2006.08859* (2020).

[63] Emanuel Parzen. "On estimation of a probability density function and mode". In: *The annals of mathematical statistics* 33.3 (1962), pp. 1065–1076.

[64] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint arXiv:1511.06434* (2015).

[65] Hari Mohan Rai and Anurag Trivedi. "De-noising of ECG Waveforms based on Multiresolution Wavelet Transform". In: *International Journal of Computer Applications* 45.18 (2012), pp. 25–30.

[66] Stephen Roberts and Lionel Tarassenko. "A probabilistic resource allocating network for novelty detection". In: *Neural Computation* 6.2 (1994), pp. 270–284.

[67] M Rosenblatt. "Remarks on some nonparametric estimates of a density function. Annals of Mathematical Statistics". In: (1956).

[68] Volker Roth. "Outlier detection with one-class kernel fisher discriminants". In: *Advances in Neural Information Processing Systems* 17 (2004), pp. 1169–1176.

[69] Leonid I Rudin, Stanley Osher, and Emad Fatemi. "Nonlinear total variation based noise removal algorithms". In: *Physica D: nonlinear phenomena* 60.1-4 (1992), pp. 259–268.

[70] Lukas Ruff et al. "A unifying review of deep and shallow anomaly detection". In: *Proceedings of the IEEE* (2021).

[71] Tim Salimans et al. "Improved techniques for training gans". In: *arXiv preprint arXiv:1606.03498* (2016).

[72] Mathew Salvaris, Danielle Dean, and Wee Hyong Tok. "Generative Adversarial Networks". In: *Deep Learning with Azure* (2018), pp. 187–208. DOI: 10.1007/978-1-4842-3679-6_8. URL: http://dx.doi.org/10.1007/978-1-4842-3679-6_8.

[73] Arman Sargolzaei, Karim Faez, and Saman Sargolzaei. "A new robust wavelet based algorithm for baseline wandering cancellation in ECG signals". In: *2009 IEEE International Conference on Signal and Image Processing Applications*. IEEE. 2009, pp. 33–38.

[74] Thomas Schlegl et al. "f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks". In: *Medical Image Analysis* 54 (2019), pp. 30–44. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2019.01.010. URL: https://www.sciencedirect.com/science/article/pii/S1361841518302640.

[75] Thomas Schlegl et al. "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery". In: *International conference on information processing in medical imaging*. Springer. 2017, pp. 146–157.

[76] Alex Sherstinsky. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network". In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306.

[77] Zhenming Shun and Peter McCullagh. "Laplace approximation of high dimensional integrals". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 57.4 (1995), pp. 749–760.

[78] Simonson et al. “Sex differences in the electrocardiogram”. In: *Circulation* 22.4 (1960), pp. 598–601.

[79] *Sinus Rhythm | EKG Training*. URL: `https://www.practicalclinicalskills.com/ekg%02reference`. accessed on 4 March 2020.

[80] Michael E Tipping and Christopher M Bishop. “Probabilistic principal component analysis”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999), pp. 611–622.

[81] Vladimir N Vapnik. “An overview of statistical learning theory”. In: *IEEE transactions on neural networks* 10.5 (1999), pp. 988–999.

[82] Cédric Villani. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media, 2008.

[83] Haowen Xu et al. “Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications”. In: *Proceedings of the 2018 World Wide Web Conference*. 2018, pp. 187–196.

[84] Zhancheng Zhang et al. “Heartbeat classification using disease-specific feature selection”. In: *Computers in biology and medicine* 46 (2014), pp. 79–89.

[85] B. Zhou et al. “BeatGAN: Anomalous Rhythm Detection using Adversarially Generated Time Series”. In: *IJCAI*. 2019.

# Appendix A

# proofs

*Proof.* (proof of theorem 2.2.13 reported from [7]). Let $\theta$ and $\theta_0$ be two parameter vectors in $\mathbb{R}^d$. Then, we will first attempt to bound $W(\mathbb{P}_\theta, \mathbb{P}_{\theta'})$, from where the theorem will come easily. The main element of the proof is the use of the coupling $\gamma$, the distribution of the joint $(g_\theta(Z), g_{\theta'}(Z))$, which clearly has $\gamma \in \prod(\mathbb{P}_\theta, \mathbb{P}_{\theta'})$.
By definition n of the Wasserstein distance, we have

$$\begin{aligned} W(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) &\leq \int_{\mathcal{X}\times\mathcal{X}} ||x-y|| d\gamma \\ &= \mathbb{E}_{(x,y)\sim\gamma}[||x-y||] \\ &= \mathbb{E}_z[||g_\theta(z) - g_{\theta'}(z)||] \end{aligned}$$

If g is continuous in $\theta$, then $g_\theta(z) \to g_{\theta'}(z), \theta \to \theta'$, so $||g_\theta - g_{\theta'}|| \to 0$ pointwise as functions of z. Since $\mathcal{X}$ is compact, the distance of any two elements in it has to be uniformly bounded by some constant M, and therefore $||g_\theta - g_{\theta'}|| \leq M$ for all $\theta$ and z uniformly. By the bounded convergence theorem, we therefore have

$$W(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) \leq \mathbb{E}_z[||g_\theta(z) - g_{\theta'}(z)||] \to 0, \quad \theta \to \theta'$$

Finally, we have that

$$|W(\mathbb{P}_r, \mathbb{P}_\theta) - W(\mathbb{P}_r, \mathbb{P}_{\theta'})| \leq W(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) \to 0, \quad \theta \to \theta'$$

proving the continuity of $W(\mathbb{P}_r, \mathbb{P}_\theta)$.
Now let g be locally Lipschitz. Then, for a given pair $(\theta, z)$ there is a constant $L(\theta, z)$ and an open set $\mathcal{U}$ such that $(\theta, z) \in \mathcal{U}$, such that for every $(\theta', z') \in \mathcal{U}$ we have

$$||g_\theta(z) - g_{\theta'}(z)|| \leq L(\theta, z)(||\theta - \theta'|| + ||z - z'||)$$

By taking expectations and $z' = z$ we

$$\mathbb{E}_z[||g_\theta(z) - g_{\theta'}(z)||] \leq ||\theta - \theta'|| \, \mathbb{E}_z[L(\theta, z)]$$

whenever $(\theta', z) \in \mathcal{U}$. Therefore, we can define $\mathcal{U}_\theta = \theta'|(\theta', z) \in \mathcal{U}$. It's easy to see that since $\mathcal{U}$ was open, $\mathcal{U}_\theta$ is as well. Furthermore, assuming that there are local Lipschitz constants $L(\theta, z)$ such that

$$\mathbb{E}_{z\sim p}[L(\theta, z)] < \infty,$$

we can define $L(\theta) = \mathbb{E}_z[L(\theta, z)]$ and achieve

$$|W(\mathbb{P}_r, \mathbb{P}_\theta) - W(\mathbb{P}_r, \mathbb{P}_{\theta'})| \leq W(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) \leq L(\theta)||\theta - \theta'||$$

for all $' \in \mathcal{U}_\theta$, meaning that $W(\mathbb{P}_r, \mathbb{P}_\theta)$ is locally Lipschitz. This obviously impies that $W(\mathbb{P}_r, \mathbb{P}_\theta)$ is everywhere continuous, and by Radamacher's theorem we know it has to be differentiable almost everywhere. The counterexample for item 3 of the Theorem is indeed Example 2.2.15. □