Short Duration ECG Into Autoencoder Followed By Clustering

An Explorational Study

S.R. van den Bos

TU Delft

Short Duration ECG Into Autoencoder Followed By Clustering

An Explorational Study

by

S.R. van den Bos

Student Name	Student Number
van den Bos	4239261

Supervisor:	C. Lofi
Daily Supervisor:	A. Ghahremani
Committee Members:	C. Lofi, G.J.P.M Houben, D.M.J Tax
Defense Date:	October 14, 2022, 14:00 hour
Faculty:	Electrical Engineering, Mathematics and Computer Science, Delft

Style: TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Summary

Electrocardiography is the craft of producing electrocardiograms. These graphs give physicians insight into the potential pathology of the heart. In order to come to a diagnosis, physicians use electrocardiograms in combination with follow-up physical examinations. There has been extensive research into automated methods that can differentiate healthy individuals from pathological individuals when given only an electrocardiogram [10, 18, 14, 13]. Some of these methods make use of neural networks that do feature extraction and classification in an end-to-end fashion [10, 14, 13]. [18] Perform feature engineering on ECG data followed by clustering. The evaluation [18] shows that the resulting clusters coincide with heart pathology annotations. This master thesis describes the search for a machine learning pipeline, where a combination of input representation, autoencoder, and clustering algorithm that produces clusters coinciding with heart pathology without being biased by either heart pathology annotations or feature engineering that is already known to be predictive of heart pathology.

Although the preexisting methods yield state-of-the-art accuracies, they do not allow us to learn about the structure and patterns in the data and the supervised methods are very costly to create. There has been no research into feature extraction by autoencoder followed by unsupervised clustering. Doing this could expose patterns in the data that could lead to improved diagnostics of heart pathology, and to automatic methods that are cheaper to create by turning the problem of diagnosis into an unsupervised or semi-unsupervised problem.

In order to find the combination of input representation, autoencoders, and clustering that is best suited for predicting heart pathology from ECG signals, experiments are done that give insight into how much the resulting clusters coincide with heart pathology. The experiments work by first feeding some representation of one-second ECG objects into the autoencoder, after which the resulting low-dimensional representations are fed into a clustering algorithm. The clustering algorithm gives every low-dimensional ECG object a cluster label. The cluster labels are mapped to heart pathology labels by making use of existing heart pathology labels. This heart pathology label can now be interpreted as a class prediction in the context of classification. This setup is created to quantitatively answer to what degree the resulting clusters coincide with heart pathology.

In the end, the concatenated image plot representation objects fed into a convolutional autoencoder followed by SOM clustering is compared to the research by [18]. The classification accuracy achieved by the autoencoder pipeline formed in this research is 0.76 ± 0.01. This means that the clusters formed by [18] coincide much more with heart pathology labels than the clusters from this research. A qualitative visualization from the low dimensional representations after the autoencoder, however, shows that the setup from this research is better at identifying patient IDs than heart pathology. This means that the features extracted by the autoencoder are salient for identifying persons, but not for identifying heart pathology.

Contents

Summary i

1 Introduction 1

2 Important Concepts 3
2.1 Artificial Neural Networks . . . 3
2.1.1 Machine Learning . . . 3
2.1.2 Gradient Descent . . . 3
2.1.3 Artificial Neural Networks . . . 4
2.1.4 Autoencoders . . . 4
2.1.5 The Curse of Dimensionality . . . 6
2.1.6 Clustering . . . 6
2.2 Electrocardiography . . . 6

3 Methodology 11
3.1 Pre-processing . . . 11
3.2 Dataset . . . 12

4 First Experiment 14
4.1 Introduction . . . 14
4.2 Extra Literature . . . 14
4.3 Methodology . . . 15
4.4 Discussion . . . 16

5 ECG Object Representations 22
5.1 Introduction . . . 22
5.2 Extra Literature . . . 22
5.3 Methodology . . . 23
5.4 Discussion . . . 24
5.4.1 Plotted FFT representations . . . 24
5.4.2 Plotted representations . . . 32
5.4.3 One-Dimensional convolutional representations . . . 32
5.4.4 Image fields representation . . . 33

6 Deep Vision Models 34
6.1 Introduction . . . 34
6.2 Extra Literature . . . 34
6.3 Methodology . . . 35
6.3.1 Hyper Parameters . . . 35
6.4 Discussion . . . 35

7 Recurrent Numeric Models 37
7.1 Introduction . . . 37
7.2 Extra Literature . . . 37
7.2.1 LSTM-based arrhythmia classification . . . 37
7.2.2 LSTM unsupervised video representations . . . 38
7.3 Methodology . . . 38
7.4 Discussion . . . 40

8 Variational Numeric models 41
8.1 Introduction . . . 41
8.2 Methodology . . . 41
8.2.1 Data Set . . . 41

8.2.2 Hyperparameters 42
8.3 Discussion 42

9 Clustering Techniques 45
9.1 Introduction 45
9.2 Methodology 45
9.3 Discussion 46

10 MIT Dataset 49
10.1 Introduction 49
10.2 Methodology 49
10.3 Discussion 49

11 Conclusion 52
11.1 Contributions 52
11.2 Recommendation Future Research 52

References 54

1

Introduction

ECG scans are measurements of multiple vectors containing detailed three-dimensional information of how the heart beats. This information is used by medical professionals to find heart pathology. Unsupervised ECG clustering is the process of looking for similarity patterns in a population of ECG objects, requiring no annotations. In this research feature extraction by artificial neural networks (ANNs) without human bias (autoencoders) is combined with unsupervised clustering in order to find patterns in the data that are not distorted by existing notions of heart pathology. The main research questions of this thesis are:

- Can autoencoders improve ECG diagnostics?
- Can autoencoders at least distinguish healthy individuals from individuals suffering from heart sicknesses?

Many automated methods exist that predict heart pathology based on ECG data. Most of these methods are based on supervised machine learning. Supervised means that during the training of machine learning, annotations are used. In the case of heart pathology annotations, they must be created very carefully by medical experts, which for some hospitals is infeasible, and they are derived from existing knowledge of heart pathology, which makes them biased.

[18] Do feature engineering, where they use existing knowledge to extract good features with respect to heart pathology, making their research biased by existing notions of what are good features with respect to the predictability of heart pathology. They follow this feature engineering up with unsupervised clustering.

The results of automated ECG diagnostics based on supervised machine learning show that supervised deep learning models can extract features from ECGs that allow these deep learning models to achieve higher than 99% accuracy on the task of classifying heart pathology. Since autoencoders are neural networks as well, they could be effective at extracting features that are salient with respect to heart pathology. Potential benefits of autoencoder feature extraction followed by unsupervised clustering are to train diagnostics algorithms without the need for (expensive) labeling of data and to find structures in the ECG data that are not biased by existing human medical science. The latter could lead to new insights, in the form of an improved system for the classification of heart diseases, or the discovery of new features that are good predictors of heart pathology.

ECG signals can contain several hundred data points per second. A naive approach to cluster would be to see every ECG object as a feature vector and cluster in a high-dimensional feature space. This approach would suffer severely from the curse of dimensionality, which is explained in the next section. In order to cluster effectively, the ECG objects must first be compressed into a number of dimensions that do not suffer as much from the curse of dimensionality. The exact upper limit of this dimensionality is not known and differs per use case. State-of-the-art feature extraction by autoencoder and unsupervised clustering [21] achieve an accuracy of around 84% on the task of classification on the MNIST data set. Very simple supervised models easily achieve around 98% accuracy on this same task. This indicates that applying autoencoder feature extraction methods followed by unsupervised clustering requires much thought and experimentation in order to even approach supervised classification methods. Another reason why this is difficult is that there is no easy way of recognizing a previously unknown clustering with respect to heart pathology. This is why this research focuses on the second research question; we can test whether this new clustering distinguishes at least healthy controls from pathological individuals.

Related research into unsupervised dimensionality reduction followed by unsupervised clustering exists. [18] Do manual unsupervised feature extraction by approximating the shape of the QRS complex part of the ECG heart wave by a polynomial function where the parameters found are the lower-dimensional representation of the ECG object. The key difference between [18] and this research is twofold; this research takes the entire ECG wave into account instead of only the QRS complex, and secondly, this research does feature extraction by autoencoder instead of manual feature extraction. Since feature extraction by neural network works so well for supervised neural network classifiers, the expectation is that unsupervised neural networks will also extract salient features with respect to heart pathology.

The key component of this research is that the methodology of finding new patterns in ECG data involves no human knowledge; not in the shape of labels and neither in the shape of feature engineering. The biggest assumption of this research is that unsupervised ANNs (autoencoders) extract salient features because supervised neural networks have been shown to be able to extract salient features. The results show that autoencoders are not able to distinguish healthy controls from pathological individuals. This means that the features are not able to produce a superior clustering, which in turn means that this research will not improve current ECG diagnostics.

The structure of this thesis is as follows; in the first section, all important concepts are explained. The second section is on methodology, this section explains the general setup throughout the different experiments. In the following sections, the setup differs in either the autoencoder part, the data representation part, the clustering part, or the dataset part. Every time one part is changed, the others are kept the same. These sections will all consist of a literature part, a methodology part, an experiment part, and a discussion part. First, four naive setups will be examined. Following this is a section of experiments focusing on data representation where a numerical input representation is compared to different image representations of plotted signals, two 1-dimensional convolutional representations, and one alternative way to map signals to the domain of computer vision. The next section will be about deeper models where ResNet-inspired autoencoders will be evaluated. After the ResNet-inspired models, recurrent models will be analyzed, followed by variational autoencoders. In the following section, different clustering techniques will be looked into and various hyperparameters will be explored. In the final section, the best parts found by the preceding experiments will be compared to an existing similar method that does feature engineering. In this final section, the dataset will be different. The thesis will end with a conclusion and suggestions for future research.

2

Important Concepts

This section outlines the concepts of artificial neural networks, the curse of dimensionality, clustering, and electrocardiography. This forms the basis for the initial, and all consecutive research.

2.1. Artificial Neural Networks

2.1.1. Machine Learning

The essence of most machine learning (ML) methods are mathematical expressions called loss functions. These loss functions are written so that their initial value is some positive number $loss > 0$. The goal of ML models is to make predictions based on a certain input. These inputs are called feature vectors. During training, the desired predictions are known for every feature vector. We refer to these desired predictions as ground truths. The loss function is usually a distance function between the ground truths and the predictions produced by the ML models. For example, if the goal of a neural network would be to predict a single number for every feature vector, the loss function could be

$$loss = \sum_{x=1}^{x=n} |p - gt_x|$$

Where p is the prediction produced by the ML model, x is the feature vector and gt_x is the ground truth associated with that prediction. To further clarify this model, the prediction is often written as a function of the ML model on the input.

$$loss = \sum_{x=1}^{x=n} |model(x) - gt_x|$$

Here the prediction p is substituted by $model(x)$. Initially, the value of the loss function is some positive value. During training, this positive value decreases and approaches zero. The training mechanism that achieves this convergence is called *gradient descent*.

The reasoning that this works is in the assumption that when the differences between the ground truths and the predictions are very small, the model has learned to produce predictions approximating the ground truths. This model can then be used to make predictions on unseen data given the assumption that the unseen data is similar to the feature vectors trained on.

2.1.2. Gradient Descent

The learning mechanism through which machine learning models learn is called gradient descent. It takes the loss function $loss = \sum |model(x) - gt_x|$ and iteratively adapts the parameters within the $model(x)$ part in such a way that the value of the entire loss function converges to some minimum value. A more complete way to write the loss function would be as follows.

$$loss = \sum_{x=1}^{x=n} |model(x; \theta) - gt_x|$$

Here θ signifies the collection of parameters contained in the model function. The place where the machine learning models store what they have learned is thus in the values of these parameters. On an intuitive level, one could say that the learned parameters are the memory or the brain of machine learning. Gradient descent can only be applied to mathematical expressions that are differentiable with respect to their parameters. For example, the graph obtained after plotting $model(x;\theta)$ on the y-axis against θ on the x-axis can not contain vertical jumps or vertical asymptotes.

2.1.3. Artificial Neural Networks

Artificial neural networks belong to a special class of machine learning models. Here the $model(x;\theta)$ function consists of a combination of smaller functions that have the property of being differentiable with respect to their input and their parameters. A very simple design is that of a univariate-multilayer-perceptron depicted in Figure 2.1.

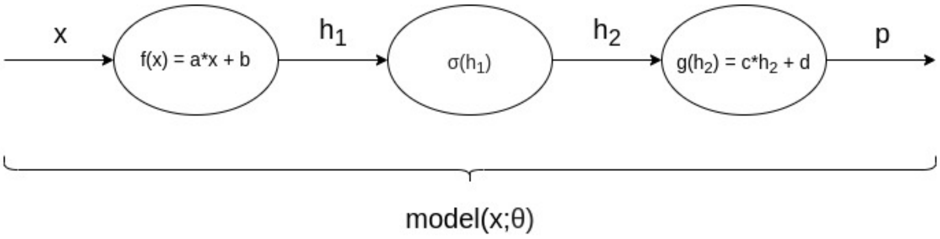


Figure 2.1: univariate-multilayer-perceptron

Here x is the initial input, $f(x)$ and $g(x)$ are linear functions as depicted, σ is the sigmoid function defined as the logistic function $\sigma(x) = 1/(1+e^{-x})$, and p is the output prediction. In this model θ is the collection of parameters $a, b, c,$ and d.

When writing the above model in a single mathematical expression, one should first substitute the functions with their outcomes.

$$h_1 = a * x + b$$

$$h_2 = \frac{1}{1+e^{-h_1}}$$

$$p = c * h_2 + d$$

Substituting h_2 in the expression of p and substituting h_1 in the expression of h_2 we can rewrite the three expressions above into the one below.

$$p = model(x;a,b,c,d) = c * \frac{1}{1+e^{-a*x-b}} + d$$

For stochastic gradient descent to work on this model, one must produce the derivatives of the above right-hand expression with respect to the parameters $a, b, c,$ and d. It should be kept in mind that this is a toy problem used for explanation. In practice, neural networks can be hundreds of layers deep. The solution to the complexity of such a loss function lies in a powerful property of neural networks.

Derivatives of loss functions with respect to their parameters $\frac{\delta loss}{\delta \theta}$ in neural networks can be easily produced by making use of the chain rule. For example $\frac{\delta loss}{\delta a} = \frac{\delta loss}{\delta p} * \frac{\delta p}{\delta h_2} * \frac{\delta h_2}{\delta h_1} * \frac{\delta h_1}{\delta a}$. Notice that the last three terms of the right-hand equation are the layers of our network of Figure 2.1 in reverse order. One can split up the layers of the neural networks and produce simple partial differentials. Combining these differentials by means of multiplication then solves the differentials of potentially extremely complex loss functions. The advantage of extremely complex neural networks and their loss functions is that they can model and predict very complex behavior, which other ML models can not.

2.1.4. Autoencoders

Autoencoders are neural networks with a specific loss function. This loss function is called the reconstruction loss. Here the prediction is still the output of the network. The difference lies in the ground truth part. For autoencoders, the input will be used as ground truth. This is likely confusing for the reader. To clear up the confusion, the model will be extended to use multiple variables as input instead of just a single value called x.

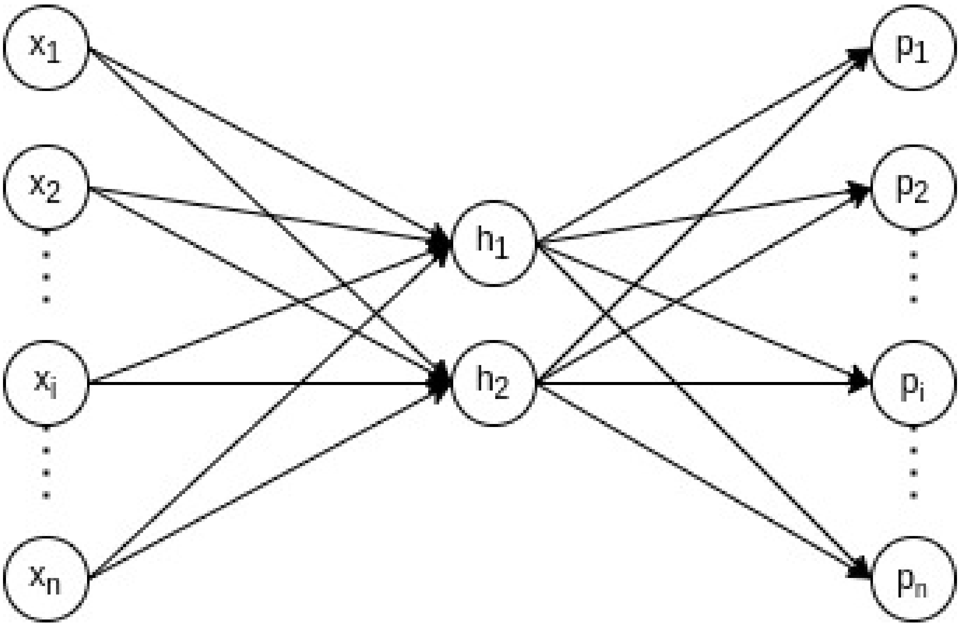


Figure 2.2: Simple autoencoder

Figure 2.2 shows a simple autoencoder. Every arrow indicates that the value from the base of the arrow will end up multiplied by a unique parameter as a term in a sum identified with the circle the arrow is pointing to. To illustrate this, two mathematical definitions of values from Figure 2.2 will follow.

$$h_1 = x_1 a_1 + x_2 a_2 + ... + x_i a_i + ... + x_n a_n$$

$$p_1 = h_1 b_1 + h_2 b_2$$

Where a_i and b_i are unique parameters with indices i. The application of autoencoders in this paper is called feature extraction, but many more applications exist. Feature extraction is the process of deriving quantitative information from the objects of interest. For example in this research, to gain insight, images are plotted in a two-dimensional plane. That means that for every image there should be two values that represent the image in some meaningful way.

For the autoencoder above, one could imagine all the input variables x_i are pixels of an image. The information contained in the pixels is then transformed into the two values h_1 and h_2 according to the equations explained. Initially, the parameters of the model are chosen randomly. This means that the two values h_1 and h_2 are arbitrary values corresponding to an image. Then the two values are transformed back into a number of values that match the number of input pixels. These values should be interpreted as the reconstructed input.

In order to teach the model to match the model prediction to the input, one must formulate a loss function that is zero when the reconstruction is perfect and a positive value when the reconstruction is not perfect. For optimal convergence, the loss function should be lower as the reconstruction gets better. To this end, the loss function from the toy example can be used $loss = \sum_{x=1}^{x=n} |model(x;\theta) - gt_x|$. Since the desired prediction is now the same as the input, the loss function should be rewritten accordingly.

$$loss = \sum_{x=1}^{x=n} |model(x;\theta) - x|$$

In contrast to the toy example, x is now a collection of pixel intensity values instead of a single value.

The model is considered trained when the loss function is deemed low enough. Equivalently that means that the reconstruction is deemed good enough. When that is the case the autoencoder will have learned two things: the model has learned to generate a reconstructed image from two specific values, and the model has learned to compress the input image in a representation that is only two values. The intuition behind this is that the two values now must contain most of the useful information in the image because from these two values the model can reconstruct the associated unique image.

2.1.5. The Curse of Dimensionality

The curse of dimensionality is an important phenomenon in machine learning. Intuitively one would think that having more data points per ECG object increases the knowledge we have of that ECG, which would mean that we could perform ML tasks with a higher degree of accuracy. When clustering in higher dimensions though, distances become less meaningful.

$$\lim_{d \to \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} = 0$$

Distance is a critical metric on which clustering is built. Distances between clusters are typically greater than distances within clusters. As the dimensionality grows, this difference becomes less pronounced. The result is that clustering in high-dimensional feature space performs poorly. The best clustering is achieved when there is a limited number of dimensions in the feature space. What this number is, differs per case and therefore requires experimentation.

2.1.6. Clustering

Most clustering variants fall under the category of unsupervised machine learning. This means that for the objects of study, there are no target values available. In clustering, one can imagine a feature space of two dimensions where the feature vectors consist of two values: an x value and a y value. Every feature vector is plotted in the feature space resulting in a scatter plot.

Some examples are in Figure 2.3. This is a collection of scatter plots where every combination of two out of four possible features is visualized. The features are quantitative descriptions of real-world objects. In this case, researchers have looked at three types of Iris flowers. The red dots signify the Setosa species, the green Versicolor, and the blue Virginica. When researching the flowers, the researchers have come up with four features: sepal length, sepal width, petal length, and petal width. It is important to note that the colors (species) are target values here, but to the machine learning algorithm the colors are not visible and in all cases of unsupervised clustering, the machine learning algorithm does not get to see the color categorization. For the purpose of explanation, however, it helps one understand the goals of clustering.

One of the goals of clustering is to find structure (patterns) in the data. This is something that supervised learning is incapable of. One possible goal is thus to learn something new for oneself instead of teaching a machine to learn something.

A different goal of clustering is prediction. From Figure 2.3 it can be seen that the colored dots are condensed into groups. This is a good sign for prediction. It means that the features of sepal length, sepal width, petal length, and petal width are good for differentiating between species. An important difference between some of the plots is that in some cases the blue and green dots overlap, this is especially the case for sepal length vs. sepal width, and it means that if the targets (flower species) would not be known in this overlapping area, one could not 100% accurately predict the flower species.

According to this clustering, the blue and green species seem to be very close, what does this mean? In a more abstract way, this means that these flowers are similar when expressed in these four features. The distance between two points of the scatter plot signifies a difference in feature quantity, for example, the petal width of all the red species flowers is significantly smaller than those of the blue species, which is why there is a large horizontal distance between the red dots and blue dots in all the right-most scatter plots.

There is an additional important aspect to the distance between any two points. What does diagonal distance mean? Visually the way humans perceive distance is called Euclidean distance $d = \sqrt{\Delta x^2 + \Delta y^2}$, where Δ signifies a difference in the following feature. In general, one can cluster with different ways of combining multiple dimensions. For example a simple alternative definition for distance $d_a = |\Delta x| + |\Delta y|$ is called the Manhattan distance.

2.2. Electrocardiography

Electrocardiography is the creation of an electrocardiogram from data obtained by measuring voltages of the human heart. A set of ten electrodes are placed on the human body. These ten electrodes are then

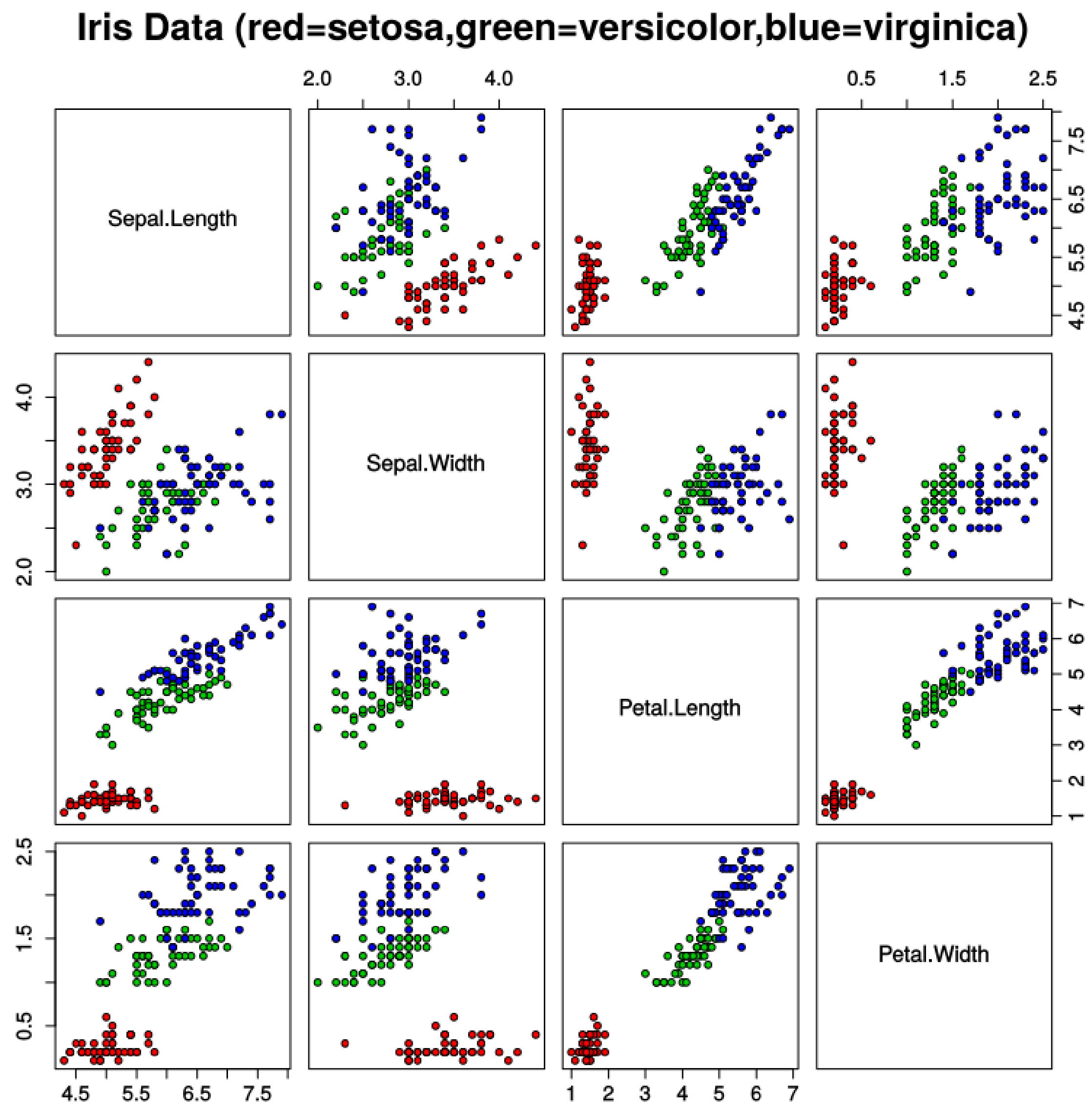


Figure 2.3: Iris dataset scatter plot from [22].

combined into usually twelve signals called leads. These leads are then combined into a cardiograph, which can be read by medical experts.

For a standard twelve-lead electrocardiogram, first, a set of ten electrodes are placed at specific locations on the patient's body (Figure 2.4). These electrodes can detect muscle contraction by the small amount of voltage produced when muscles contract. The abbreviations mean: right arm (RA), left arm (LA), right leg (RL), left leg (LL), and v1 up to and including v6 are called the precordial electrodes. The electrodes record a patient's heart's behavior by taking a measurement of the voltage at a certain frequency. This frequency differs per instrument. This creates a signal at every electrode. Next up the electrode measurements are combined into something called leads. A lead is a linear combination of the signals of two or three electrodes, that yields information about how the heart is beating and in what direction.

The standard electrocardiograph consists of twelve leads. The first three leads (I, II, and III) are called limb leads. They are composed of signals measured by the electrodes called after the limbs.

- I = LA - RA
- II = LL - RA

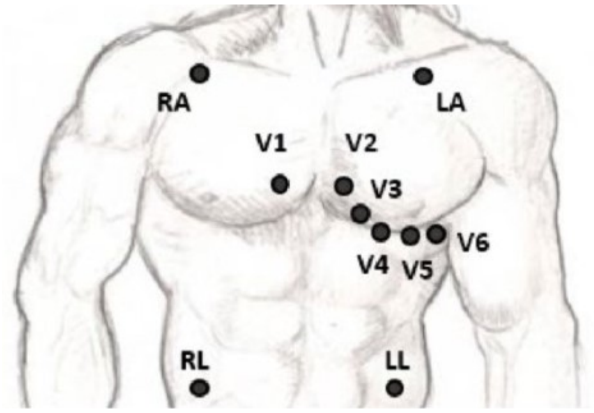


Figure 2.4: Electrode placements as described by [15]

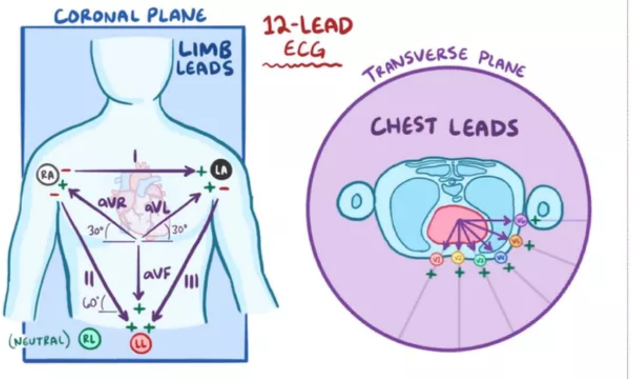


Figure 2.5: Planes in which the ECG leads point their measurements. Taken from [1].

- III = LL - LA

The next three leads are called augmented limb leads. They are formed by taking the same three limb electrodes, but they are compared to an additional (virtual) electrode called the *Goldberger's central terminal*. They are known as the augmented vector right (aVR), the augmented vector left (aVL), and the augmented vector foot (aVf), and they are derived as follows:

- aVR = RA - $\frac{1}{2}$ ($LA + LL$)
- aVL = LA - $\frac{1}{2}$ ($RA + LL$)
- aVF = LL - $\frac{1}{2}$ ($RA + LA$)

The remaining six leads are formed by the precordial electrodes by using the precordial electrodes as positive poles and a virtual electrode called Wilson's central terminal as a negative pole.

As depicted in Figure 2.5, the first six leads called the limb leads all record voltages in directions in the coronal plane, which is colored light blue. The last six leads called the chest leads, record their voltages in directions in the transverse plane, which is colored purple.

An ECG lead signal is a periodic signal that follows a repeating sequence of depolarization and repolarization. For trained medical experts, an ECG yields a lot of information. Among other things, an ECG can help a medical expert determine the heart rate, size of the heart chambers, and damage to the heart. In medical literature on heart pathology, a single ECG period contains three important regions; the P-wave, the QRS-complex, and the T-wave. These and more can be seen in Figure 2.6. Heart pathology often shows by means of anomalies in the described regions, however, sometimes additional physical examinations are needed to ascertain or exclude a diagnosis. For example, a cardiac enzyme test can help diagnose or exclude heart attacks.

Figure 2.7 shows how the 12 traditional lead signals are combined into one single format that is used by medical experts. Leads that are given the same color in the first figure describe heart functioning in

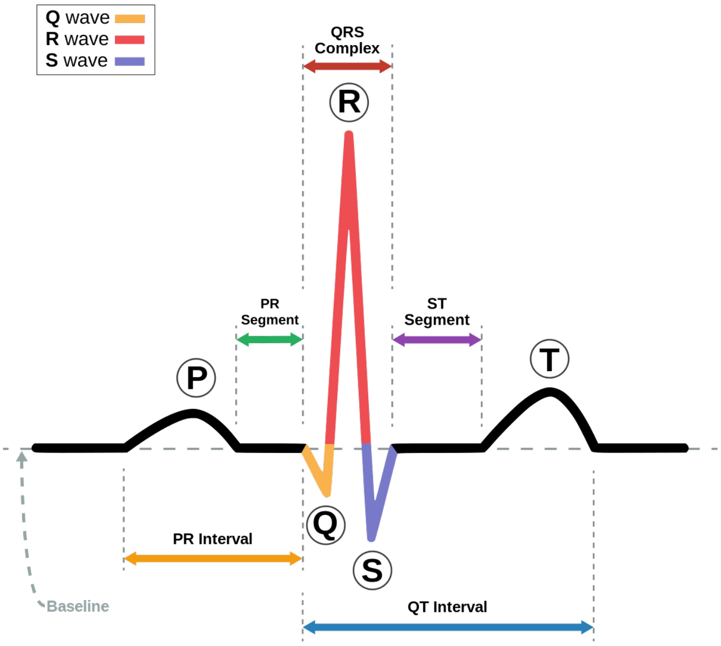
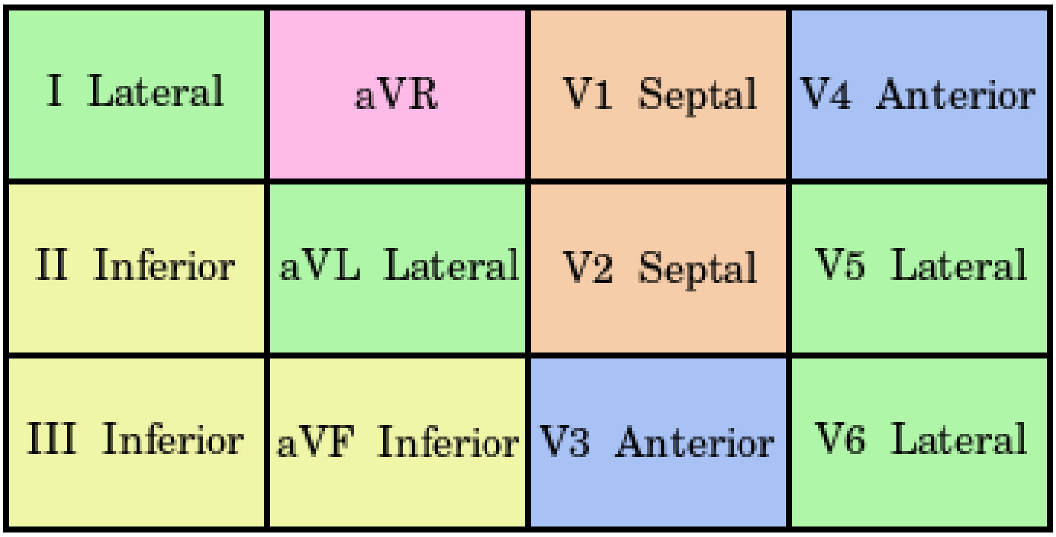


Figure 2.6: Marked parts of a single ECG lead period by [3].

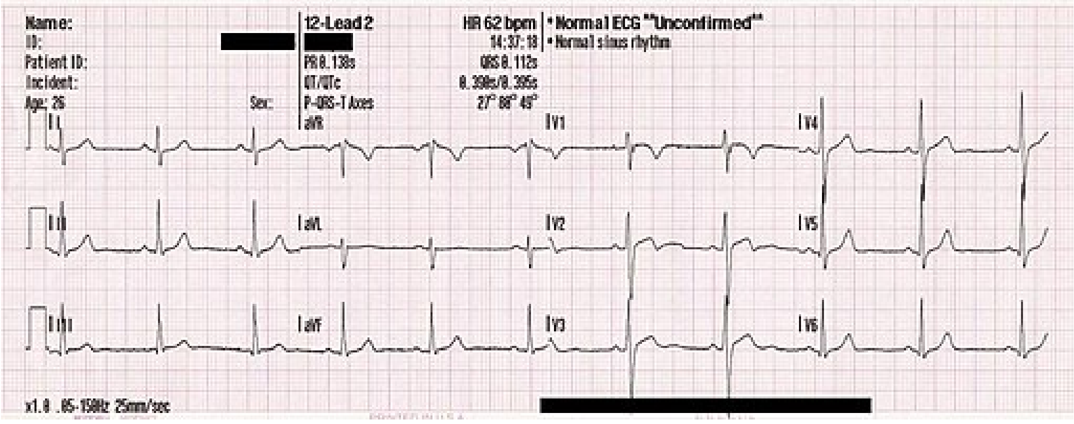
contiguous heart regions. Explanations of how and why the leads are in this specific configuration is complex and too profound for this thesis.

Figure 2.7: Electrocardiograph, the human-readable combination of 12 lead signals.



I Lateral	aVR	V1 Septal	V4 Anterior
II Inferior	aVL Lateral	V2 Septal	V5 Lateral
III Inferior	aVF Inferior	V3 Anterior	V6 Lateral

(a) Diagram that combines the 12 lead signals into 3 signals parallel in time that each consist of 4 leads contiguous in time. Image from [6].



Name:
ID: [redacted]
Patient ID:
Incident:
Age: 26
Sex:

12-Lead 2
[redacted]
PR 0.138s
QT/QTc
P-QRS-T Axes

HR 62 bpm
14:37:18
QRS 0.112s
0.390s/0.395s
27° 68° 49°

• Normal ECG "Unconfirmed"
• Normal sinus rhythm

x1.0 .05-150Hz 25mm/sec
[redacted]

(b) A practical example of an electrocardiograph, lead names are included. Image from [20].

3

Methodology

In this section, the machine learning pipeline, data representations, and the dataset will be introduced. The aim of the methodology is to answer quantitatively; to what degree do the lower-dimensional representations of the ECG objects contain information about heart pathology? The answer is found in the resulting structure and patterns in the lower-dimensional data. In a perfect scenario, the lower-dimensional data forms clusters where the data points within one cluster are all derived from patients with the same heart pathology. In the machine learning pipeline as described below and visible in Figure 3.1, this perfect scenario yields an accuracy of 1.0.

In order to find structures and patterns in the data, one could create a pipeline that directly clusters the ECG objects in a feature space equal to the dimensionality of the ECG vectors. The problem is that doing this will yield poor results since the curse of dimensionality will have severe effects in a 1000-dimensional feature space. To overcome this hurdle one should compress the ECG vectors in such a fashion that most of the information of interest is retained. In this case, the lower-dimensional counterparts of the ECG objects should retain as much information on heart pathology as possible. In order to facilitate this compression, different kinds of neural networks called autoencoders are experimented with. The general setup is as follows: the ECG objects in some representation will be fed into some autoencoder for compression, after which the compressed ECG objects will be used in some clustering algorithm. The cluster labels found will then be mapped to the heart pathology annotation that is most abundant within that cluster. The set of resulting labels will be interpreted as the predicted label in the machine learning context of classification. The performance metrics of this task are then compared to [18] and [10] to answer the research questions. Many visualizations will be made and discussed in order to gain insight into why the particular experiment setup is (not) working.

3.1. Pre-processing

Unless indicated otherwise the datasets are pre-processed into two different data representations. The first representation is a single-channel image of a plot of the leads. The leads are combined into a single

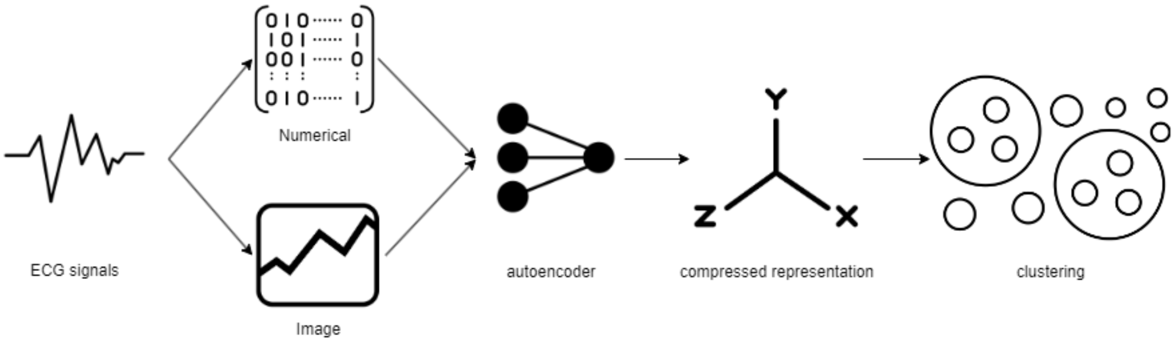


Figure 3.1: ML Pipeline schematic

Table 3.1: Diagnostic Classes.

Diagnosticclass	*Numberofsubjects*
Myocardial infarction	148
Cardiomyopathy/Heart failure	18
Bundle branch block	15
Dysrhythmia	14
Myocardial hypertrophy	7
Valvular heart disease	6
Myocarditis	4
Miscellaneous	4
Healthy controls	52

image in different ways depending on the experiment. These different ways will be discussed in the sections of the experiments where they are used. The second representation is a matrix of voltages, where the different rows of the matrix signify the different ECG leads and the columns signify the moment in time where the voltage was measured. This means that the matrices have twelve to fifteen rows depending on the number of ECG leads, and have 1000 columns, which is the result of the sample rate of 1000 hertz and the length of the ECG recording of one second. All representations that directly make use of these voltages, we call numerical representations in contrast to the image representations.

For both cases the data is stored in a Numpy Array that has four indices; object (N), channel (C), height (H), and width (W). The NCHW format is an often used format in the world of computer vision deep learning. Some neural network layers, for example, the convolutional layer, require the data to be in this format.

The recordings are multiple minutes in length. In order to make every feature vector equal in length and to have enough data, the recordings are cut into pieces of one second. From Figure 3.1, it can be seen that the diagnostic class distribution over patients is not uniform. This could introduce unwanted biases, which should be avoided. This is done by cutting up the ECG pieces with a sliding window. The ECGs of patients with diagnostic classes that are abundant have a slight or non-existent overlap. For rare diagnostic classes on the other hand, a sliding window with a large overlap is used. In this manner, data augmentation and avoidance of a diagnostic class bias are achieved simultaneously.

During pre-processing, the feature vectors with miscellaneous and myocardial hypertrophy labels are dropped.

3.2. Dataset

The dataset used in this thesis is called the Physikalisch-Technische Bundesanstalt dataset [4], [16]. This is German for the physical-technical federal institute, it refers to the national metrology institute of Germany in Berlin. The dataset was created in 1995 by multiple medical experts who in consultation with each other and by taking into account multiple physical examinations per patient provided diagnostic annotations. The resulting quality of the dataset is very high and [4] consequently has been cited 600 times according to google scholar.

The dataset contains 549 records from 290 subjects aged 17 to 87. 209 Of these subjects are male, while the other 81 are female. Each subject appears in one to five records. Each record contains 15 leads; the 12 conventional leads and 3 Frank leads. Each signal has a sampling rate of 1000 hertz with a 16-bit resolution divided over ± 16.384 mV. Included in extra files for most of these ECG records are detailed clinical summaries, including age, gender, diagnosis, and where applicable more information. The distribution of diagnostic classes is depicted in Figure 3.1.

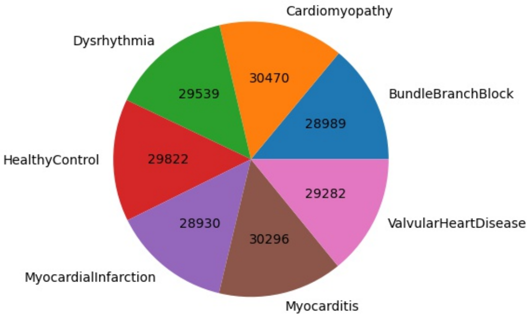


Figure 3.2: Label distribution of the training set.

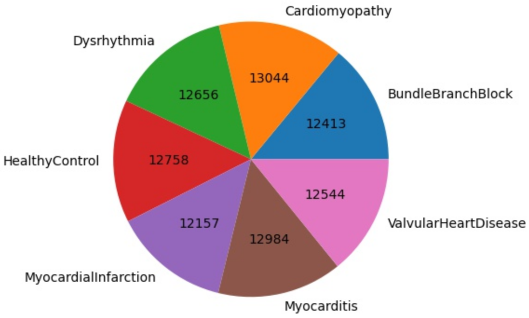


Figure 3.3: Label distribution of the test set.

4

First Experiment

4.1. Introduction

In this experiment, the aim is to create a simple baseline model that already shows either a new clustering that could lead to new insights or a clustering of the ECG data similar to existing medical classification models. In later experiments, the aim is to improve the found clustering in a way that makes the boundaries between clusters clearer.

4.2. Extra Literature

Existing research ([10], [14]) on ECG classification shows that supervised deep learning techniques allow for the extraction of salient features with regard to the classification of heart pathology.

[18] found that using a feature engineering technique called Hermite basis functions, and a clustering technique called self-organizing maps (SOM), a clustering can be obtained that wrongly classifies only 1.5% of the ECGs of the MIT ECG dataset. In their research [18] focus on fitting the QRS complex to a sequence of polynomials. Feature engineering has the weakness of not extracting the features that the high-performing deep-learning supervised models extract. Additionally, the feature engineering technique in question ignores the parts of the ECG outside the QRS complex, ignoring additional information. This additional information could not only improve classification performance but also lead to new insights when using the model for data exploration.

A simple first approach to the problem of feature extraction followed by clustering is to use a state-of-the-art feature extractor called an autoencoder followed by k-means clustering. [21] Created a model called deep-k-means. This model does feature extraction and clustering at the same time. In the first step of their model, they pre-train an arbitrary autoencoder, which in their research was a simple stacked fully-connected autoencoder. After pre-training, they initialize k cluster centers using the traditional k-means clustering. In the second phase of the training, the deep-k-means model performs gradient descent on two loss objectives; the first objective is the traditional reconstruction objective of the autoencoder. The second objective is a cluster loss that minimizes intra-cluster distance and maximizes inter-cluster distances. The research shows superior performance on various tasks, each

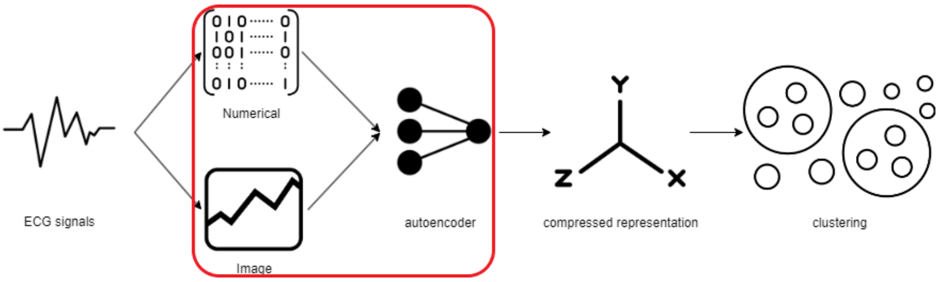


Figure 4.1: Subject of experiments for this section.

with a different dataset. One of which is (unsupervised) classification on the MNIST written digits dataset. The deep-k-means model produces ten clusters, one cluster for every digit. The authors then use a bipartite matching algorithm that matches every cluster label to their most likely ground truth digit class. In this way, they do not train the deep learning model on the true labels, but the true labels are used in the evaluation method.

[9] researched a similar model. This model is a combination of an autoencoder and k-means clustering as well. The main differences are twofold; the autoencoder consists of convolutional layers instead of fully connected layers, and the definition of cluster loss differs in the way the nearest cluster is defined. [9] Defines the nearest cluster with an argmin operation, while [21] formulate an argmin by means of an e-power. [21] Argue that an argmin constructed in this fashion optimizes the gradient descent based learning because the gradient of this argmin definition is smoother than the traditional discrete argmin.

In contrast to the previous two models discussed, [14] create a supervised model. In their research [14] create a convolutional classifier that classifies ECGs into different classes of heart pathology. In contrast to all other ECG machine learning, this research takes as input not vectors or matrices of signal values, but images of plots of the ECG signals. In this way, the authors map the ECG classification problem to a computer vision problem. The model scores above 99% average accuracy in a ten-fold cross-validation experiment.

4.3. Methodology

In the experiments, two models are compared on two different input representations. Quantitative comparisons will be made with respect to the performance metrics of accuracy, adjusted random score, and normalized mean information. The first two metrics will be derived from a comparison between the predicted labels and the actual labels, where the predicted labels will be obtained by using a simple algorithm that maps the cluster labels to their most abundant ground truth label. Normalized mean information will be derived from comparisons between the cluster labels and the patient id annotation. The last metric will yield a measure of the degree to which ECG data from the same patient will be grouped in the same cluster. Qualitatively reconstructions and clustering spaces will be visualized.

If the clusters yield high quantitative performance metrics, it means that the features extracted by the algorithm are likely the features that existing medical science also considers to be good predictors for heart pathology. If the quantitative metrics are bad, the clusters found will be different from the existing medical perspective. Now the visualizations should give insight into whether this new clustering at least discerns healthy individuals from sick ones.

The first input representation consists of the twelve ECG leads of length 1000 stacked on top of each other to form a matrix of twelve rows and 1000 columns. Every matrix will be normalized by dividing by the maximum value. This way normalization is achieved while differences between leads are kept intact. This is important because differences in leads indicate abnormalities in the direction the heart is beating. The second input representation will be a mapping of the ECG problem to the field of computer vision after the model of [14]. This means that inputs are single-channel images of plots of the ECG signals. The twelve signals are combined by means of concatenation. The resulting plot is thus a plot of a single vector of length 12000.

The two autoencoders that are compared will be a fully-connected autoencoder against a convolutional autoencoder. The fully-connected autoencoder will have fully connected layers followed by batch normalization, ReLU non-linear activation layers, and dropout layers. The convolutional autoencoder will use a pattern of convolutional layers alternated with ReLU non-linear activation layers. The layers sizes of the fully-connected autoencoder will be d-500-500-2000-embedding-2000-500-500-d with d signifying the input dimensionality. The encoder of the convolutional autoencoder will consist of three convolutional layers; the first one has a kernel of size five by five, and 32 output channels. The second layer has a kernel of size five by five as well, and 64 output channels. The last layer has a kernel of size three by three and 128 output channels. In order to condense the number of features to the desired embedding size, the last convolutional layer is followed by a flattening layer and a fully connected layer. The decoder is the same as the encoder only in reversed order.

All networks are pre-trained for 50 epochs and fine-tuned for 20 epochs with a replication factor of ten in order to get an uncertainty quantification. The reconstruction error used is the mean squared error and the optimizer used is Adam with an initial learning rate $lr = 1e^{-3}$. For the embedding size,

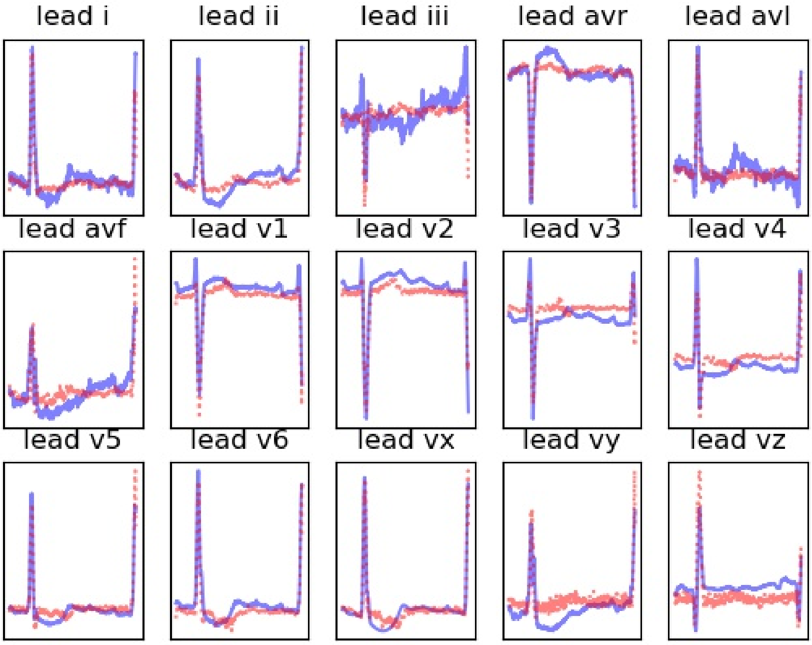


Figure 4.2: Input and reconstruction of the convolutional autoencoder on the numeric data representation. Blue indicates the input, red indicates the reconstruction.

Table 4.1

autoencoder	input representation	train accuracy	train nmi	test accuracy	test nmi
fully connected	numeric	0.23 ± 0.01	0.15 ± 0.01	0.32 ± 0.01	0.29 ± 0.01
fully connected	plot	0.23 ± 0.03	0.12 ± 0.05	0.23 ± 0.02	0.14 ± 0.05
convolutional	numeric	0.23 ± 0.01	0.09 ± 0.02	0.24 ± 0.02	0.12 ± 0.02
convolutional	plot	0.30 ± 0.01	0.24 ± 0.01	0.30 ± 0.03	0.27 ± 0.02

the same embedding will be used as in [21]. They fixed their embedding size to 10. The experiments are run on a desktop computer with 48 GB of main memory, an intel 9700k i7, and an Nvidia 2800 RTX graphics card. The entire experiment takes four days, fourteen hours, and seven minutes.

4.4. Discussion

The results show that all outcomes perform far worse than the feature engineering model from [18]. The accuracy scores in Table 4.1 show that all configurations of the experiment score poorly. The scores in Table 4.1 are accuracies after pre-training, but before fine-tuning. The fine-tuning process derived from [21] does not seem to work on this data. This is why from this point on, future experiments will not be using the fine-tuning deep-k-means algorithm.

The fully-connected autoencoder seems to score relatively good on the numeric representation while the convolutional autoencoder scores better on the plot representation. This is expected since convolutional models are known to perform better at computer vision problems.

The visualizations of the reconstructions of the numeric data representation Figures 4.2 and 4.3 show something unexpected; the convolutional model, which has significantly worse accuracy, is able to reconstruct the data better than the fully-connected autoencoder. Results on the visualizations of the plot data representation reconstructions are as expected, the convolutional model performs better.

In Figure 4.6, the distribution of different objects from the same patient is plotted. Here different dots that have the same color signify different objects from the same patient. It can be seen that none of the configurations are able to cluster objects from the same patient in the same location. During preliminary experimentation, it was already clear that the capability of these models to cluster based on

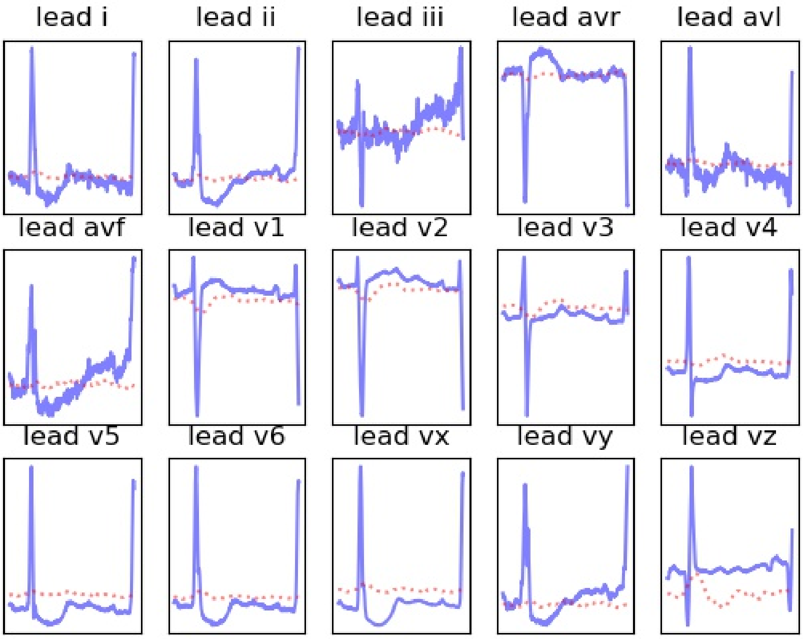


Figure 4.3: Input and reconstruction of the fully-connected autoencoder on the numeric data representation. Blue indicates the input, red indicates the reconstruction.

heart pathology was insufficient. It was hypothesized that perhaps the models were able to recognize the same patient because the preconception was that that would be easier for the model. Figure 4.6 indicates the opposite. It is important to note here that all three patients plotted, suffer from Myocardial infarction. This heart pathology is by far the most abundant heart pathology with 148 out of the 268 patients used in this experiment. This means that during the data augmentation using a sliding window, patients suffering from this pathology have had no sliding window applied to their ECG objects. During the discussion of Figures 4.8 and 4.9 it will become clear whether some configurations are able to identify the same patient in other circumstances.

The lower-dimensionality ECG representations are ten-dimensional, this means that the 1000 values that an ECG object consists of are decreased to only ten values. If those values were to be plotted, a ten-dimensional plotting space would be needed. This is impossible so the ten dimensions need to be reduced further. In this case, the aim is to provide insight into the data in two dimensions. This means that every ECG object will be represented by two values, one for the x-axis and one for the y-axis. A well-known technique that achieves this is called principal component analysis (PCA). In Figure 4.7, this two-dimensional space is visualized. The triangles signify the cluster centers obtained from the deep-k-means clustering algorithm. The placement of the centers explains why the finetuning results were so low. The deep-k-means algorithm uses the cluster centers as parameters in the same way deep neural networks use parameters. They are tuned by means of a variant of stochastic gradient descent.

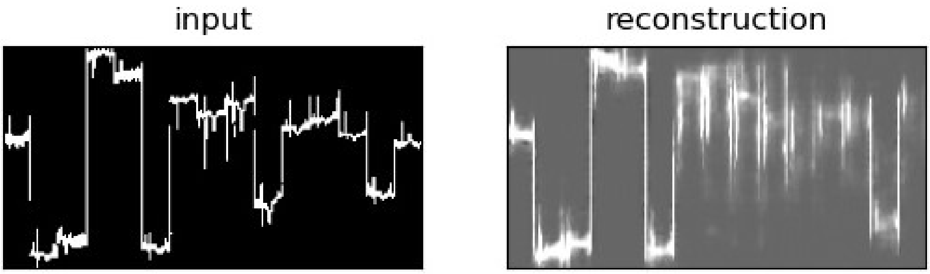


Figure 4.4: Input and reconstruction of the convolutional autoencoder on plot data representation. Left is the input, right is the reconstruction.

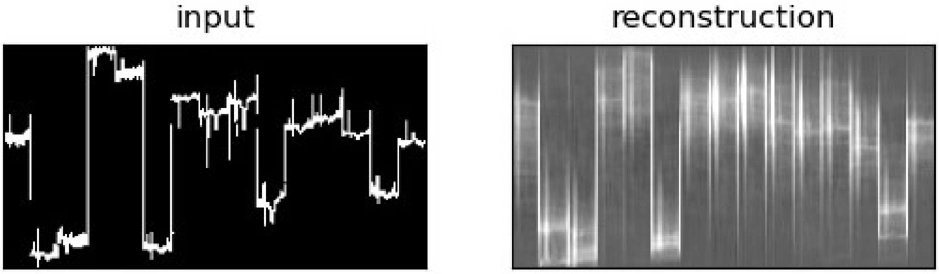


Figure 4.5: Input and reconstruction of the fully-connected autoencoder on plot data representation. Left is the input, right is the reconstruction.

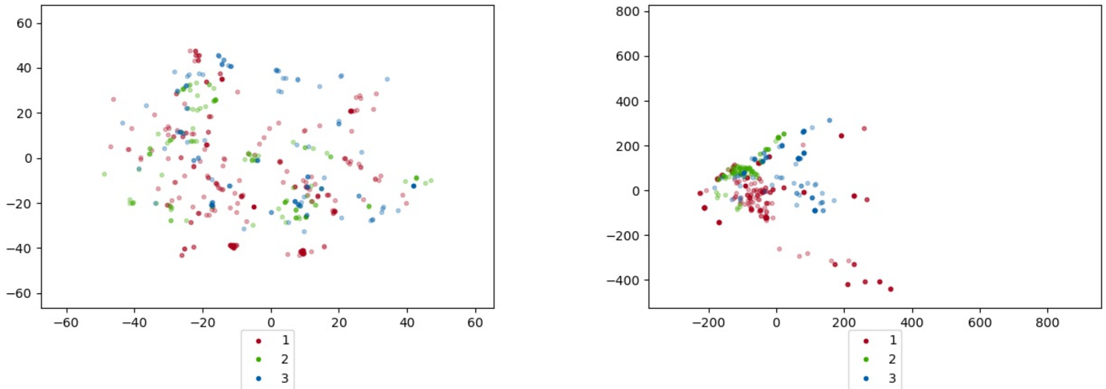
In the case of deep-k-means, the cluster center parameters are tuned at the same time as the parameters from the autoencoders. A possible explanation for the location of the cluster centers could be that the speed at which the cluster centers move through stochastic gradient descent is slower than the speed at which the data points move through stochastic gradient descent.

PCA is only one out of many methods that are used to reduce dimensionality. A different well-known dimensionality reduction algorithm is called t-SNE. This method is based on the assumption that the underlying structure of the ten-dimensional data is of lower dimensionality. This assumption is called the manifold assumption. T-SNE exploits this assumption resulting in a dimensionality reduction technique that is radically different from PCA. The resulting two-dimensional data is plotted in Figures 4.8 and 4.9. From the figures, it becomes apparent that the two convolutional configurations yield better discernible clusters than the fully-connected configurations with respect to heart pathology. This is in line with the visualizations of the reconstructions but not with the accuracy metrics. Judging from this visualization, the best low-dimensional representation is achieved with the convolutional model on the plotted data.

Returning to the subject of clustering the same patient in the same cluster, in Figure 4.9a one can roughly differentiate four blue clusters and six purple clusters. Table 4.1 states that the corresponding heart pathology is represented by four and six patients respectively. This indicates that in these two cases the model in question is able to recognize the same patient. The explanation here is that the mechanism that provides data augmentation and a balanced label distribution (the sliding window) has a very high degree of overlap as a direct result of the low number of patients that represent the heart pathology in question.

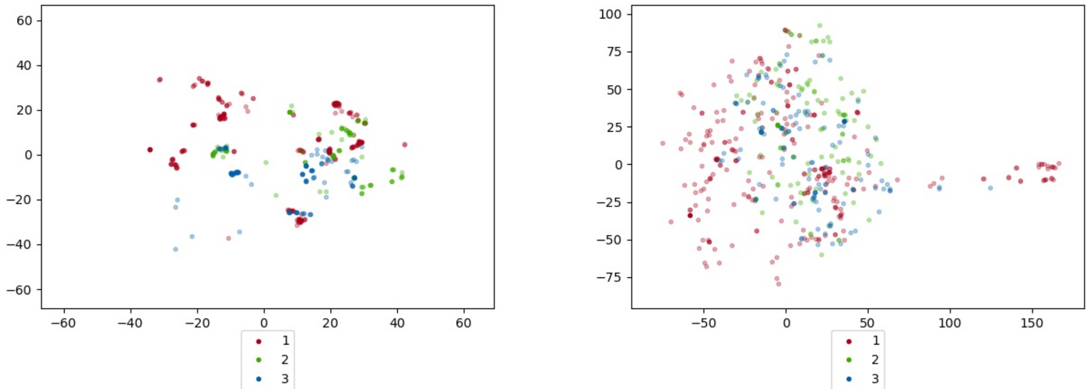
The models currently perform near complete randomness, which is the worst possible result and Figure 4.10 shows that training for more epochs is not going to result in better performance. In order to differentiate better between two bad-performing models, the task should be made easier. In consultation with the medical professionals from Erasmus Medisch Centrum (EMC), it is decided to focus only on ECG data from patients that are either healthy or suffer from Bundle Branch Block (BBB). In order to make this research reproducible in other data sets, the number of leads is reduced from fifteen down to the conventional twelve.

Figure 4.6



(a) Patient IDs from the convolutional autoencoder on the numeric data representation.

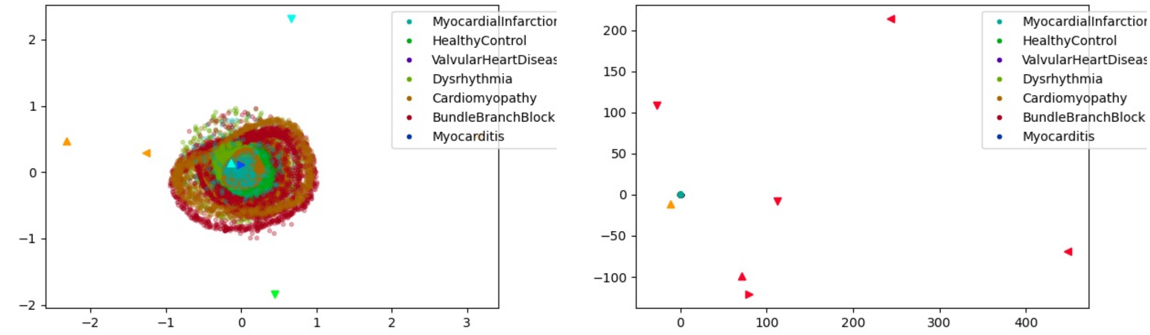
(b) Patient IDs from the fully-connected autoencoder on the numeric data representation.



(c) Patient IDs from the convolutional autoencoder on the plot data representation.

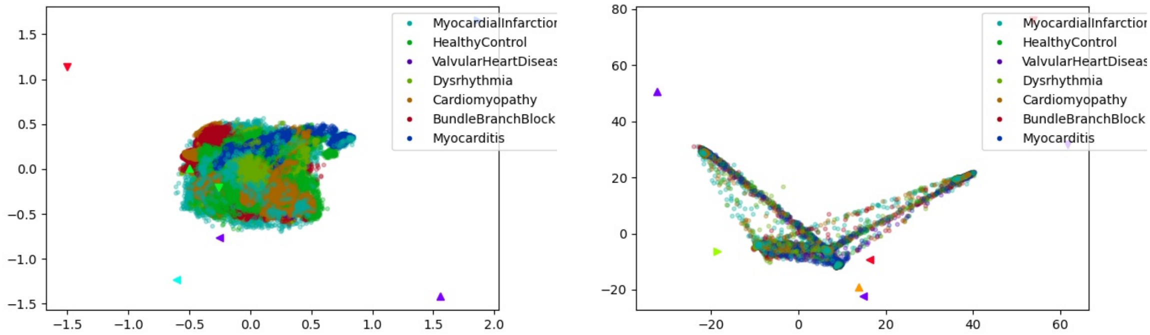
(d) Patient IDs from the fully-connected autoencoder on the plot data representation.

Figure 4.7



(a) PCA reduction of the convolutional autoencoder on the numeric data representation.

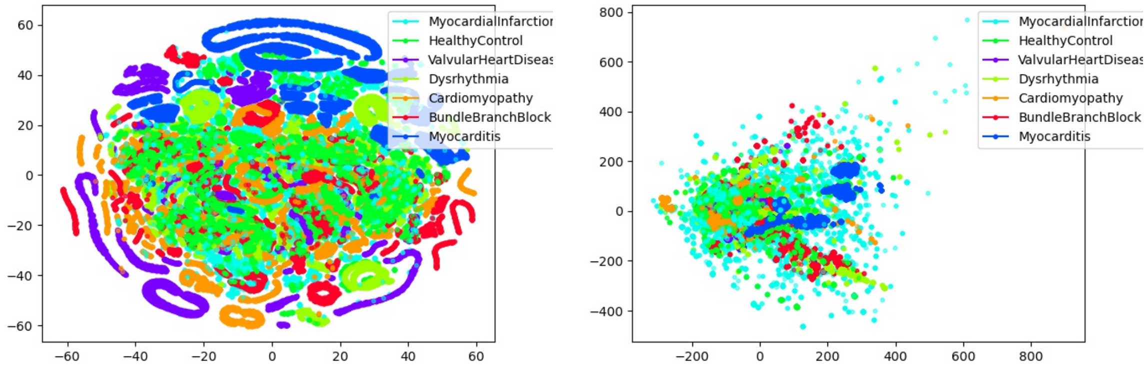
(b) PCA reduction of the fully-connected autoencoder on the numeric data representation.



(c) PCA reduction of the convolutional autoencoder on the plot data representation.

(d) PCA reduction of the fully-connected autoencoder on the plot data representation.

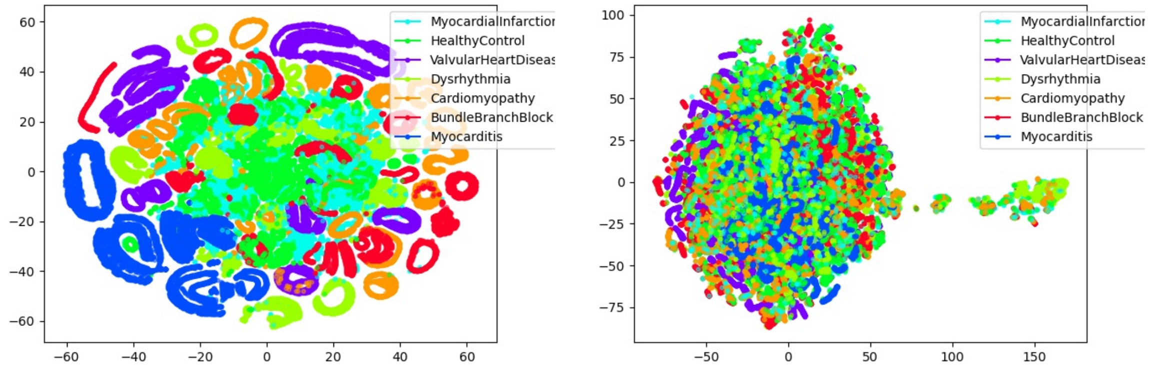
Figure 4.8



(a) t-SNE reduction of the convolutional autoencoder on the numeric data representation.

(b) t-SNE reduction of the fully-connected autoencoder on the numeric data representation.

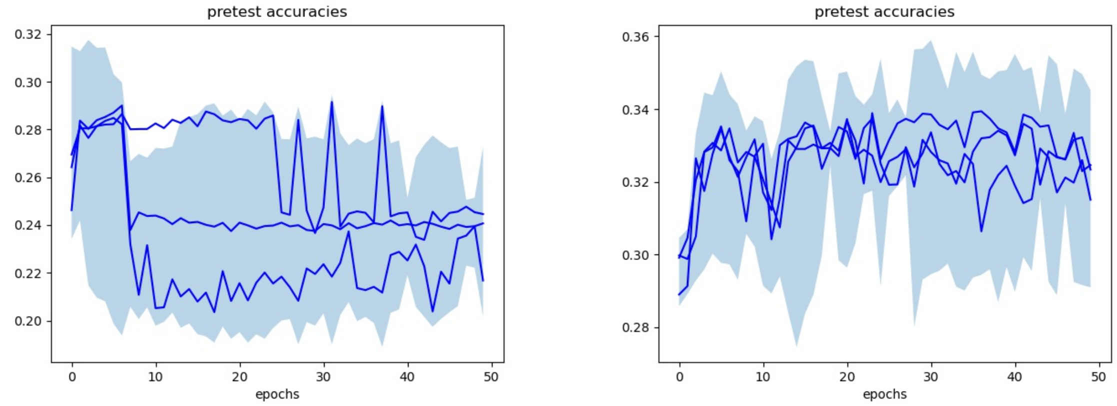
Figure 4.9



(a) t-SNE reduction of the convolutional autoencoder on the plot data representation.

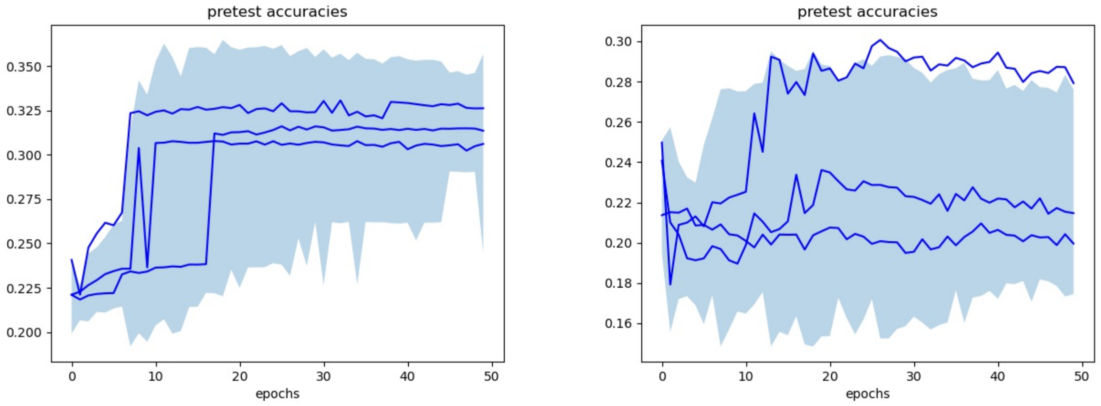
(b) t-SNE reduction of the fully-connected autoencoder on the plot data representation.

Figure 4.10



(a) accuracy of the convolutional autoencoder on the numeric data representation.

(b) accuracy of the fully-connected autoencoder on the numeric data representation.



(c) accuracy of the convolutional autoencoder on the plot data representation.

(d) accuracy of the fully-connected autoencoder on the plot data representation.

5

ECG Object Representations

5.1. Introduction

Different input representations highlight different attributes of the ECG signals. One possible explanation for the bad performance metrics of the setups that have been experimented with up until now is that the data representations highlight features of the ECG signals that are not correlated with heart pathological outcomes. The autoencoders make use of these features for reconstruction, fulfilling their objective, but the extracted features are not correlated with heart pathology, which is what this research is looking for. The hypothesis is that other data representations enable the autoencoder to extract salient features with respect to heart pathology, resulting in higher performance metrics from experimentation.

5.2. Extra Literature

In their research, [10] first decompose the ECG signals using empirical mode decomposition (EMD). The resulting signals, called intrinsic mode functions (IMFs), are used to reconstruct the original signal. This procedure is used as a method to denoise the data. The original signal is an element-wise summation of the twelve traditional ECG leads. After the denoising step, the signal is fed into a one-dimensional convolutional neural network that is designed for supervised classification. The model is trained on different data sets; MIT-BIH, St. Petersburg, and the dataset used for this research up until now; PTB. The resulting accuracies attained after testing are 97.7%, 99.71%, and 98.24% respectively. The code published in this research was used in order to reproduce. Unfortunately, the EMD technique failed for a significant number of objects. This was the reason to deem this research not reproducible. A different novel contribution of this research is that the ECGs are processed in a one-dimensional data representation.

[26] Were motivated by the recent advances in deep learning on computer vision, to propose a framework for encoding time series as images. They encode the time series as an image of three channels; the first channel is a Grammian angular summation field, the second a Grammian angular difference field and the third channel is a Markov transition field. The Grammian angular fields have several advantages according to the authors; the mapping from time series to Grammian angular fields is bijective, meaning there is one and only one unique Grammian angular field for any time series and the

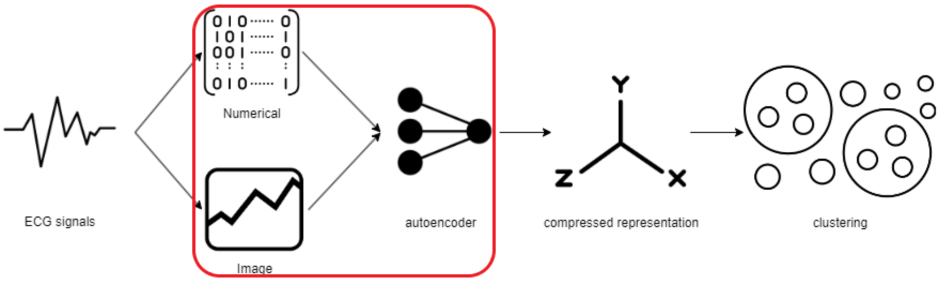


Figure 5.1: Subject of experiments for this section.

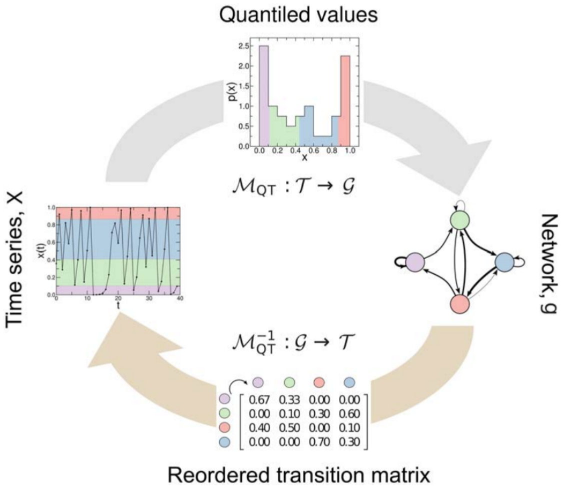

Quantiled values
p(x)
x
$\mathcal{M}_{QT} : \mathcal{T} \rightarrow \mathcal{G}$
Time series, X
x(t)
t
Network, g
$\mathcal{M}_{QT}^{-1} : \mathcal{G} \rightarrow \mathcal{T}$
0.67 0.33 0.00 0.00
0.00 0.10 0.30 0.60
0.40 0.50 0.00 0.10
0.00 0.00 0.70 0.30
Reordered transition matrix

Figure 5.2: Relation between time series and complex networks from [5].

reverse is also true. Secondly, as an intermediate step, the Cartesian coordinates from the time series are mapped to polar coordinates. According to the authors, this has the advantage of preserving absolute temporal relations.

The process of converting a time series to a Markov transition field is illustrated in Figure 5.2. As a first step, the time series is split into vertical bins as indicated by the colors in the left image. Data points from each bin are assigned to a node in a network. A transition matrix is then constructed by looking at the transitions of two consecutive points in the time series. For example, if a point in the blue bin is followed by a point from the red bin, one transition from blue to red is then counted. In order to turn these transition counts into a proper Markov transition matrix, the rows are normalized, to sum up to one. In their evaluation, [26] compare the classification performance of this three-channel representation to other techniques used to represent time series in different ways with consecutive classification. The different techniques are evaluated on 20 data sets. Most of these techniques were state-of-the-art prior to this research. From the 20 data sets, this research yields the lowest misclassification rate on nine, making this the best-performing data representation that was tested.

5.3. Methodology

In this experimentation setup, different input representations are used while the rest of the experimentation setup is kept the same. The dataset used is the PTB dataset. From this dataset only the objects are used that are related to patients suffering from bundle branch block and the healthy controls. The autoencoder used has an encoder that consists of three convolutional layers with same padding, a stride of two, and kernel size 5x5, 3x3, and 3x3 respectively. All convolutional layers are followed by ReLU activation layers, and the last ReLU layer is followed by a fully-connected layer that reduces the input dimensionality to the desired low-dimensional autoencoder output of ten. The decoder is a mirror image of the encoder that has its last ReLU activation layer before the reconstruction part removed. After the autoencoder, the data is clustered using the traditional k-means clustering algorithm as implemented by the scikit-learn python library. The number of clusters or hyperparameter k is fixed at two, the same as the number of unique labels. As first input representation, the best-performing input representation from the previous experiment is used, which is a plotted image of the twelve leads concatenated after each other. Inspired by [10] the next two leads are a plotted image of the twelve leads summed element-wise and a one-dimensional vector of data of that same summed-up vector. The following five experiment setups are plotted images of signals that are fast-Fourier transformations of the summed-up numeric data representation. All variants of this representation plot the first 40 frequencies and the intensities. The difference lies in the time interval over which the fast-Fourier is done. This interval is fixed at 1 second, 2 seconds, 3 seconds, 4 seconds, and 30 seconds.

Table 5.1: Performance metrics of the different data representations.

model name	train acc	test acc	train loss	test loss	train nmi	test nmi
plot FFT 1sec	0.55 ± 0.01	0.53 ± 0.01	1.24 ± 0.08	1.10 ± 0.05	0.09 ± 0.00	0.13 ± 0.01
plot FFT 2sec	0.53 ± 0.00	0.51 ± 0.00	1.49 ± 0.09	1.41 ± 0.05	0.09 ± 0.00	0.13 ± 0.00
plot FFT 3sec	0.54 ± 0.00	0.52 ± 0.00	1.41 ± 0.04	1.46 ± 0.02	0.09 ± 0.00	0.14 ± 0.00
plot FFT 4sec	0.54 ± 0.01	0.52 ± 0.00	1.30 ± 0.06	1.35 ± 0.03	0.09 ± 0.00	0.14 ± 0.00
plot FFT 30sec	0.55 ± 0.02	0.53 ± 0.01	0.35 ± 0.04	0.74 ± 0.01	0.14 ± 0.01	0.18 ± 0.01
plot concatenated	0.56 ± 0.05	0.55 ± 0.05	6.58 ± 0.34	8.49 ± 0.17	0.07 ± 0.08	0.08 ± 0.09
plot summed-up	0.55 ± 0.02	0.54 ± 0.02	3.98 ± 0.08	4.17 ± 0.06	0.20 ± 0.07	0.21 ± 0.07
matrix conv 1D	0.53 ± 0.01	0.55 ± 0.03	1.23 ± 0.10	2.35 ± 0.28	0.05 ± 0.02	0.08 ± 0.04
vector conv 1D	0.51 ± 0.00	0.51 ± 0.00	1.06 ± 0.06	1.39 ± 0.29	0.00 ± 0.00	0.01 ± 0.01
image fields	0.52 ± 0.01	0.52 ± 0.01	1.90 ± 0.11	1.93 ± 0.10	0.00 ± 0.00	0.00 ± 0.00

5.4. Discussion

The overall performance metrics are disappointing. The results will be discussed in four sections. The first section will discuss all the FFT plot representations, the second all the other plot representations, the third will discuss the 1-dimensional convolutional representations, and finally, the last section will discuss the image fields.

5.4.1. Plotted FFT representations

The FFT representations are experimented on by taking FFTs of different time intervals. For all other representations, the duration of each object is one second. This means that for the FFTs that have longer intervals than one second, there is an increase in the overlap between the subsequent objects. This invalidates the results from FFTs derived from longer intervals because potential good performance could be caused by the overlap. To see if there is any potential at all, these FFTs are investigated nonetheless. Intuitively the increase in overlap should result in clusters from the same patient being clustered in the same cluster more often. The metric that reflects this type of phenomenon is NMI. Apart from the remarkable NMI of the plot summed-up data representation, the NMIs from the FFT representations reflect this intuition by having the highest NMIs. The increase in NMI versus the increase in time interval, and with that, the increase in overlap is quite low. For example, the difference between the test NMIs of the 1-second FFT and the 4-second FFT is only 0.01. Given the fact that the consecutive 4-second FFTs from the same patient have more than 75% of underlying ECG signals in common, this increase in NMI of only 0.01 is puzzling.

The reconstruction visualizations of the FFT representations can be found in Figures 5.3 and 5.4. The quality of these reconstructions seems decent, but there is much empty space, which gets only worse in FFT plots derived from longer intervals. Possible improvements could be made by changing the range of the vertical axis, or changing the scale of the vertical axis, to a logarithmic scale for example.

The two-dimensional plots of the low-dimensional space can be found in Figures 5.5, 5.6, and 5.7. In

input
reconstruction
input
reconstruction

(a) Reconstruction of an FFT over a 1 second interval.

(b) Reconstruction of an FFT over a 2 second interval.

input
reconstruction
input
reconstruction

(c) Reconstruction of an FFT over a 3 second interval.

(d) Reconstruction of an FFT over a 4 second interval.

Figure 5.3

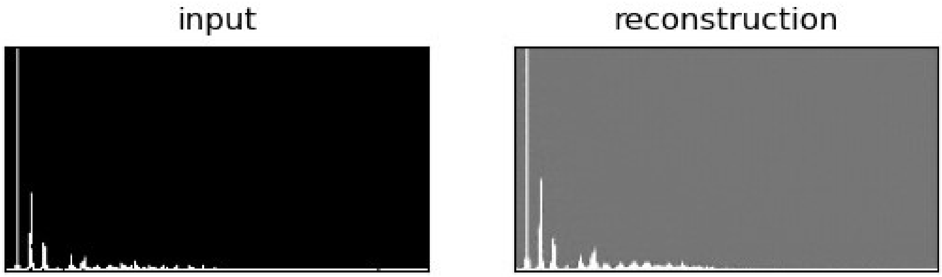
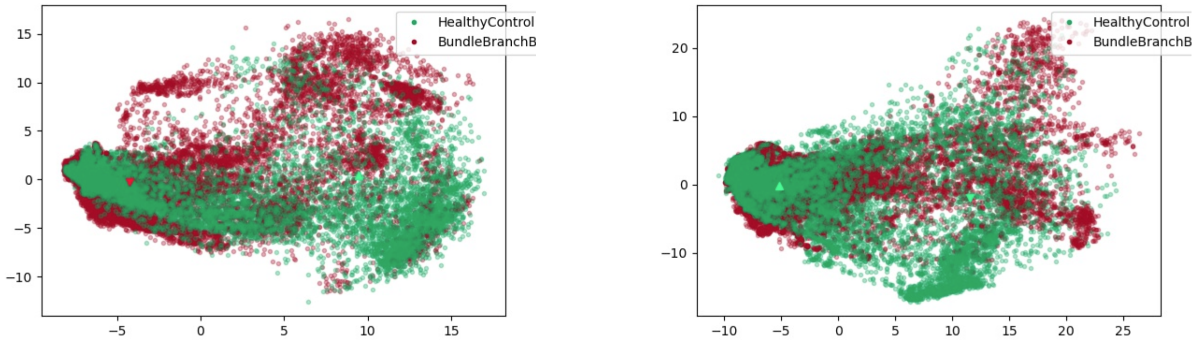
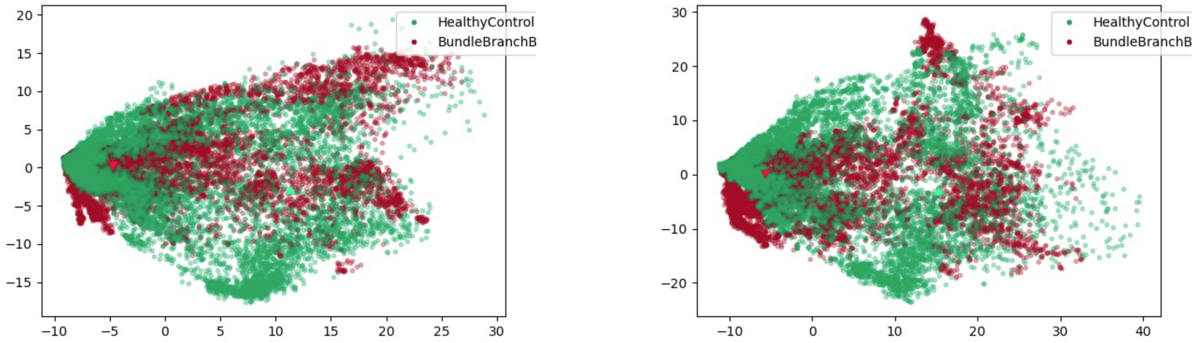


Figure 5.4: FFT reconstruction of a 30 second interval.



(a) PCA of an FFT over a 1 second interval.

(b) PCA of an FFT over a 2 second interval.



(c) PCA of an FFT over a 3 second interval.

(d) PCA of an FFT over a 4 second interval.

Figure 5.5

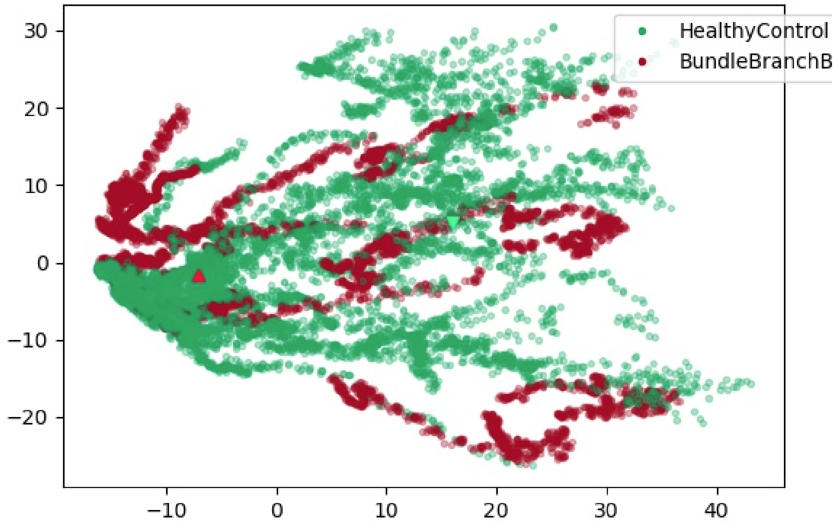
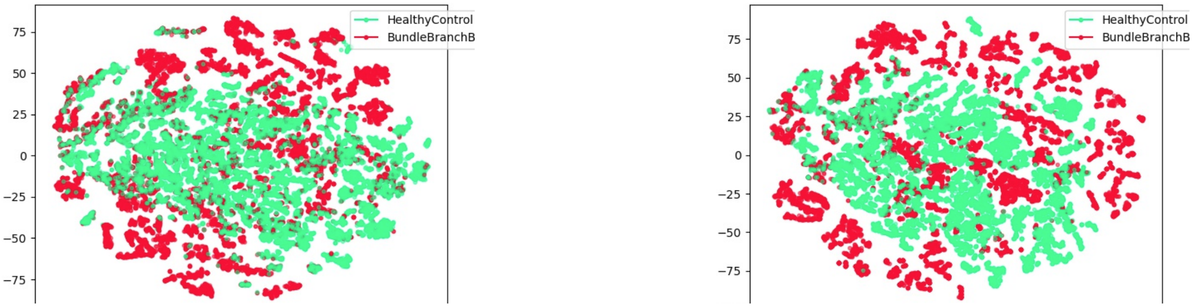
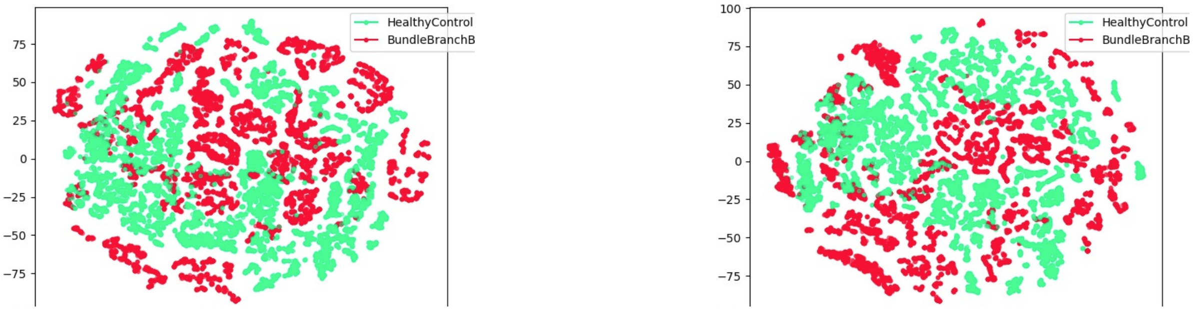


Figure 5.6: PCA of an FFT over a 30 second interval.



(a) t-SNE of an FFT over a 1 second interval.

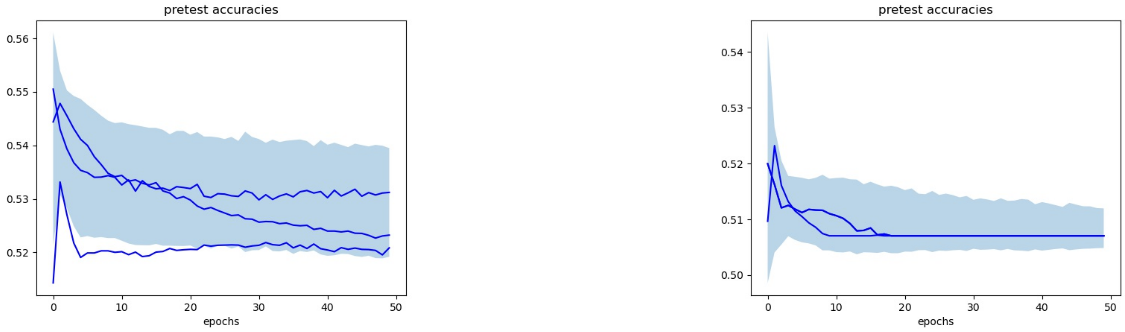
(b) t-SNE of an FFT over a 2 second interval.



(c) t-SNE of an FFT over a 3 second interval.

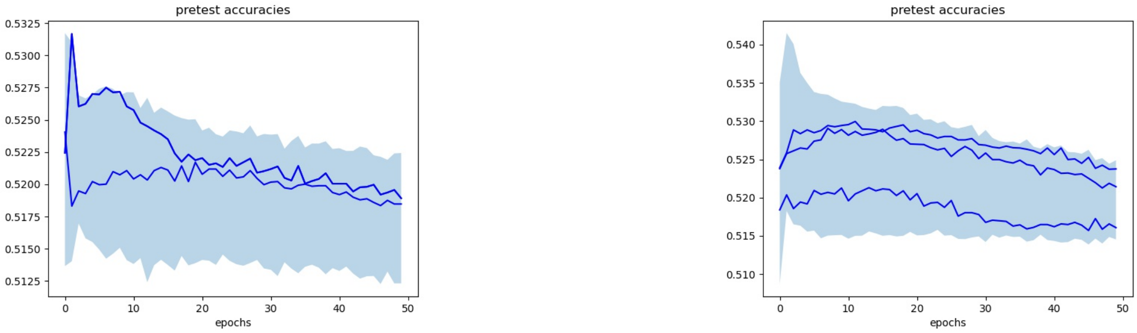
(d) t-SNE of an FFT over a 4 second interval.

Figure 5.7



(a) Accuracies of an FFT over a 1 second interval.

(b) Accuracies of an FFT over a 2 second interval.



(c) Accuracies of an FFT over a 3 second interval.

(d) Accuracies of an FFT over a 4 second interval.

Figure 5.8

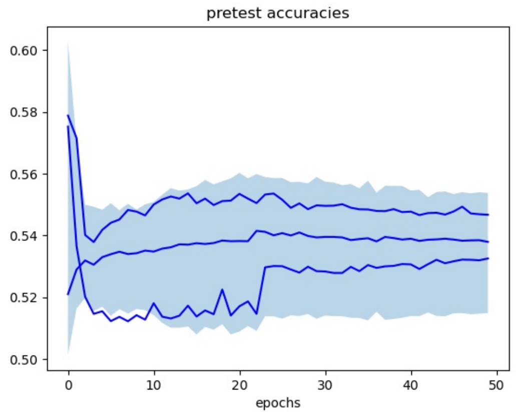
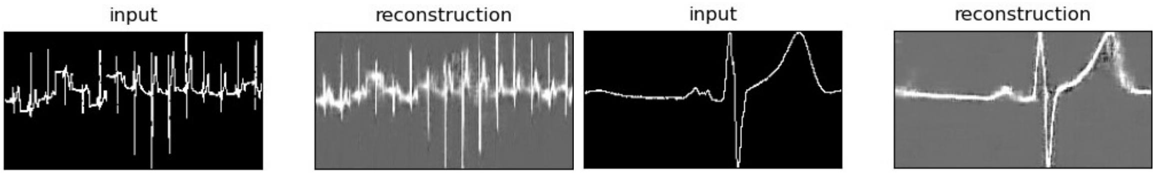


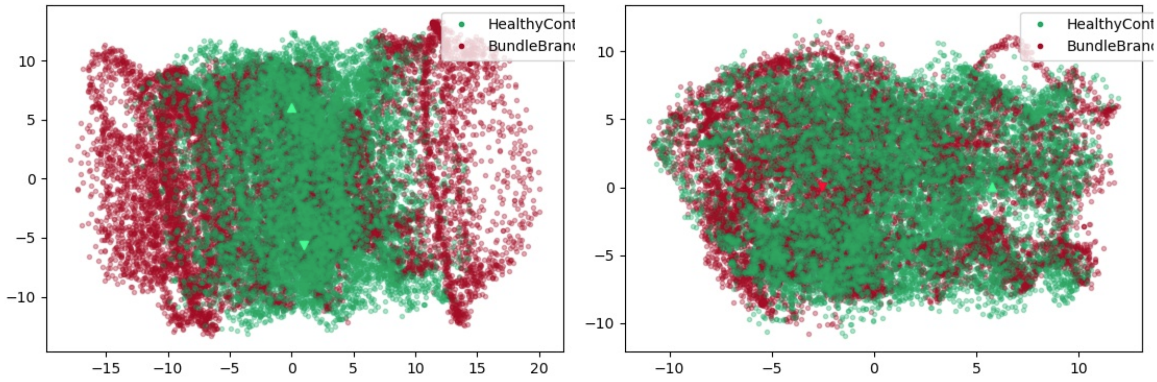
Figure 5.9: Accuracies of an FFT over a 30 second interval.



(a) Reconstruction of the concatenated plot representation.

(b) Reconstruction of the element-wise summed up plot representation.

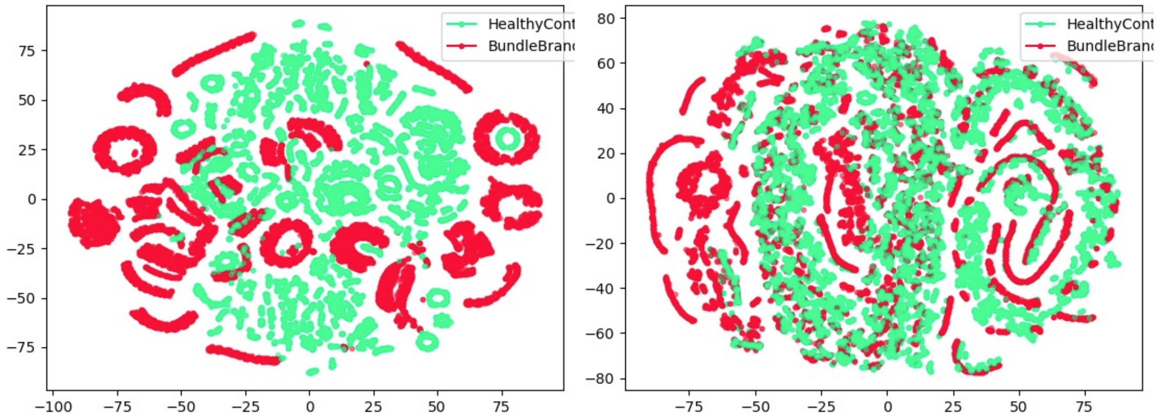
Figure 5.10



(a) PCA of the concatenated plot representation.

(b) PCA of the element-wise summed up plot representation.

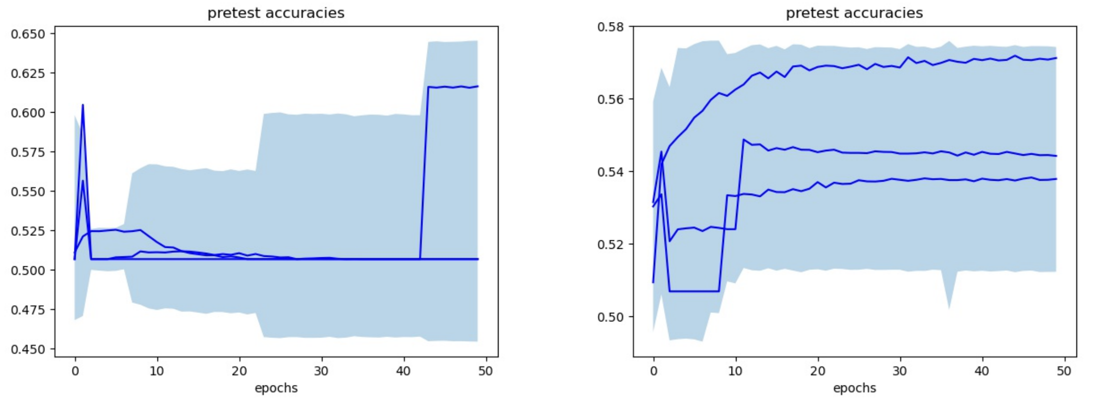
Figure 5.11



(a) t-SNE of the concatenated plot representation.

(b) t-SNE of the element-wise summed up plot representation.

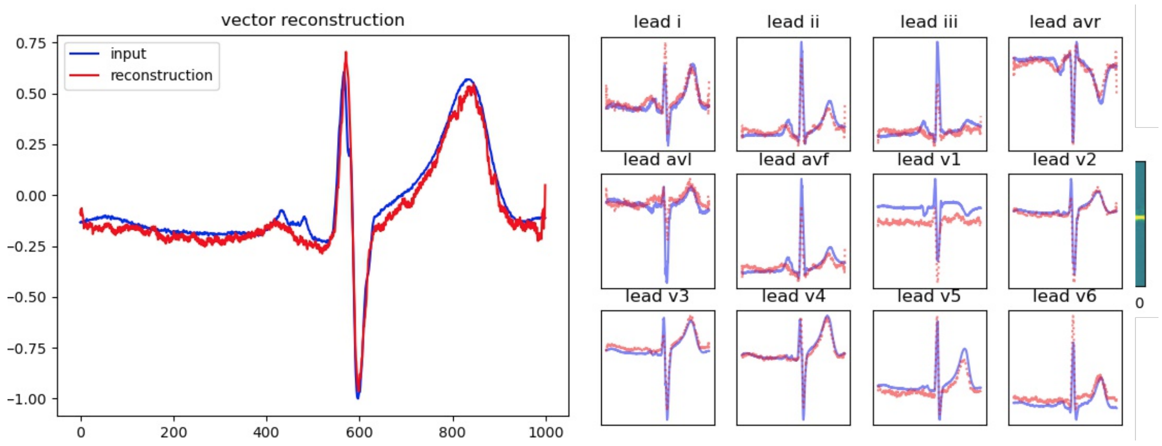
Figure 5.12



(a) Accuracies of the concatenated plot representation

(b) Accuracies of the element-wise summed up plot representation.

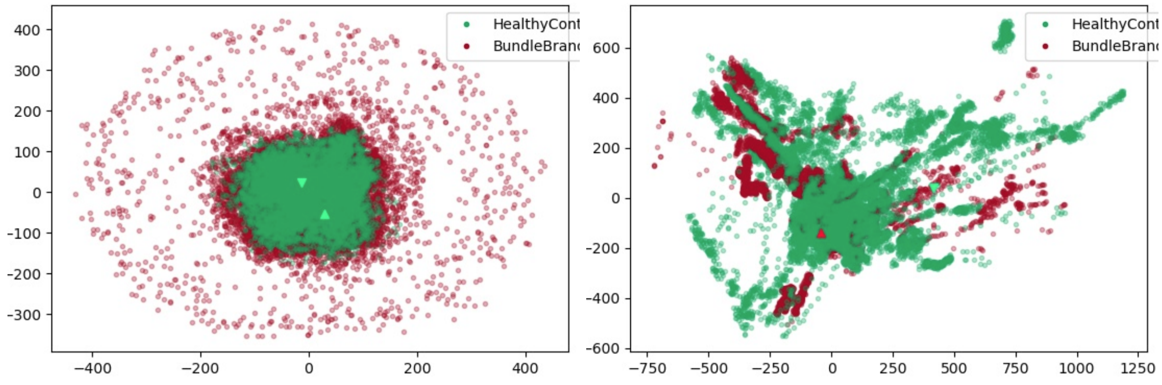
Figure 5.13



(a) Reconstruction 1D convolution summed-up element-wise representation.

(b) Reconstruction 1D convolutions 12-channel representation.

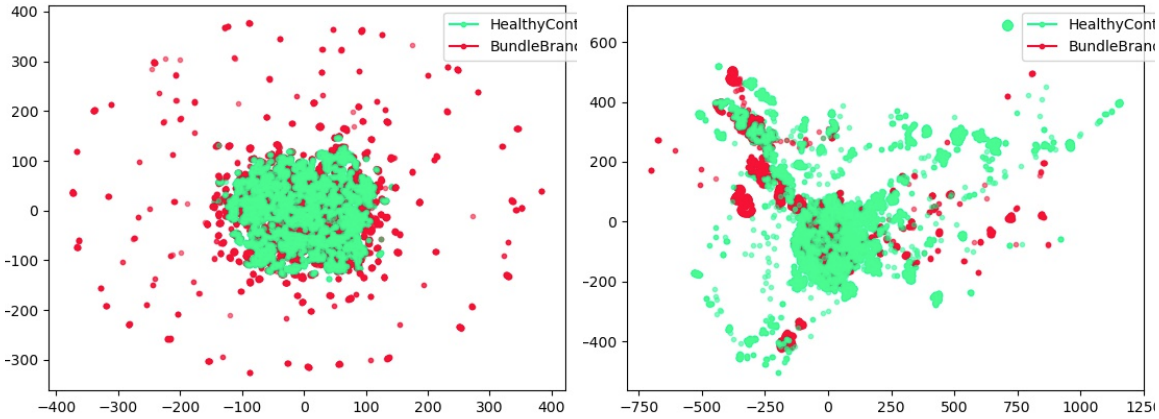
Figure 5.14



(a) PCA of the 1D convolution summed-up element-wise representation.

(b) PCA of the 1D convolutions 12-channel representation.

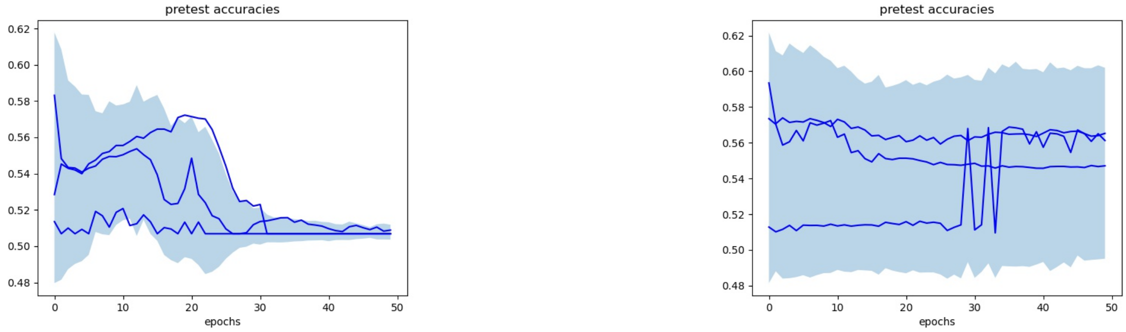
Figure 5.15



(a) t-SNE of the 1D convolution summed-up element-wise representation.

(b) t-SNE of the 1D convolutions 12-channel representation.

Figure 5.16



(a) Accuracies of the 1D convolution summed-up element-wise representation.

(b) Accuracies of the 1D convolutions 12-channel representation.

Figure 5.17

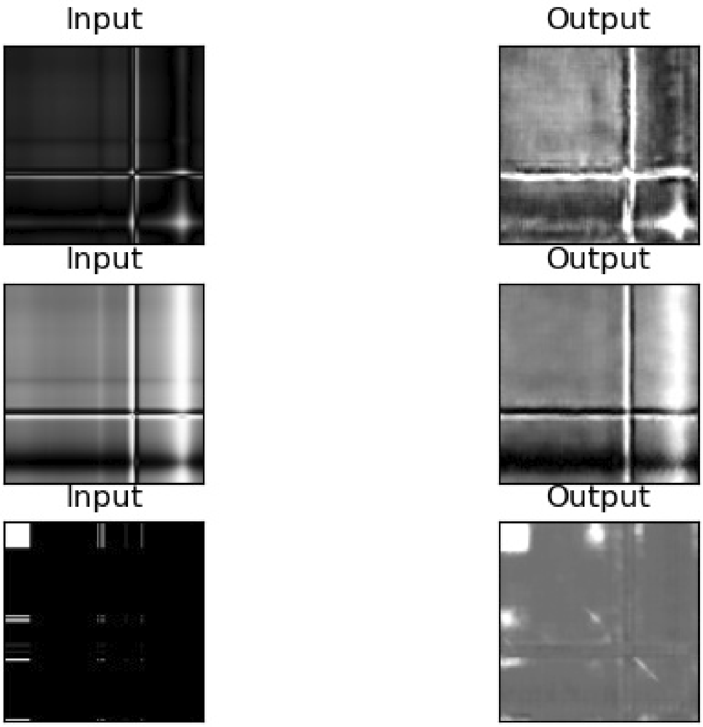


Figure 5.18: Reconstruction of the image fields.

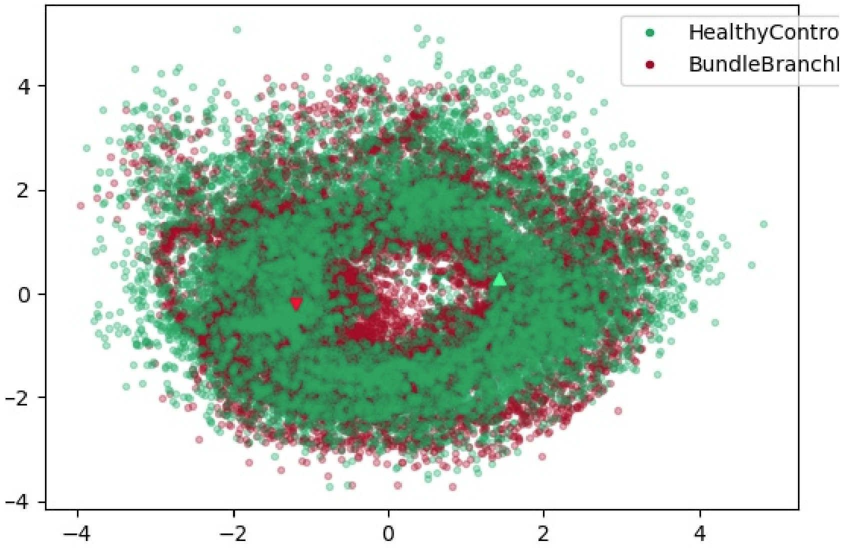


Figure 5.19: PCA of the image fields.

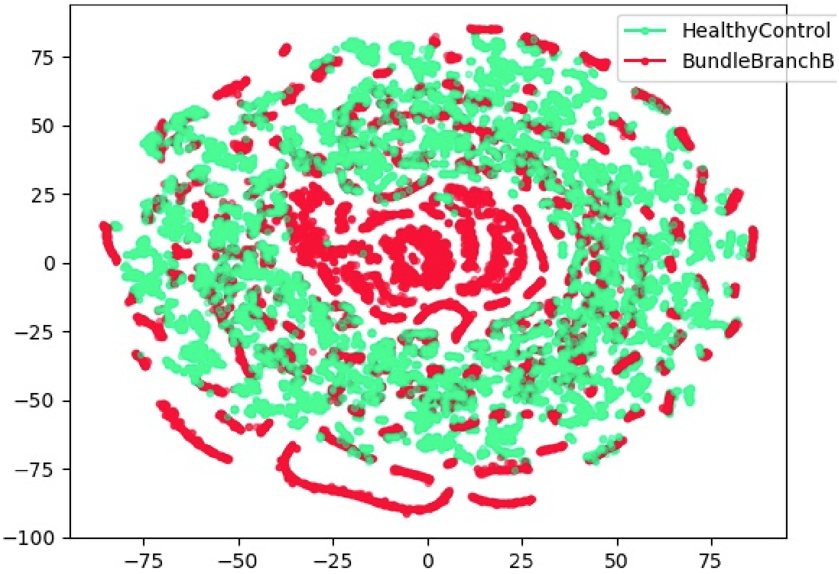


Figure 5.20: t-SNE of the image fields.

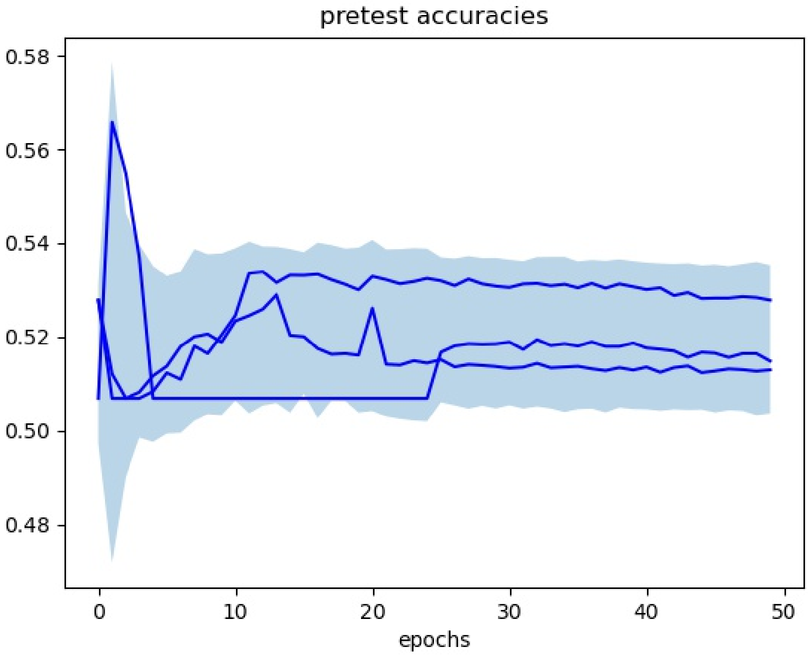


Figure 5.21: Accuracies of the image fields.

Figure 5.5 the healthy and BBB patients seem to be somewhat discernible, making them already seem better than expected. The t-SNE plots from Figure 5.7 give the impression that only FFTs from longer time intervals have the healthy controls and BBB patients somewhat separated, and although it seems decent, there are better discernible data representations in this same experiment.

An interesting phenomenon can be seen in Figure 5.6, the data points form long lines. A potential explanation is that the sliding window from consecutive objects of the same patient, in the case of FFTs from an interval of 30 seconds, share about 97% of the underlying ECG signals. This means that every consecutive point is plotted very close to its predecessor, which in turn means that every line of points seen in Figure 5.6 represents the objects derived from the same patient. The relatively large distances that these lines make then indicate that there is very little information retained in the low-dimensional representation of the patient's identities.

Finally, from Figures 5.8 and 5.9 it can be seen that training the model for additional epochs will not result in better performance metrics, in some cases this even results in worse metrics.

5.4.2. Plotted representations

The plotted representations consist of two representations; the concatenated plotted representation and the element-wise summed-up plotted representation. The first representation is the representation that is also used in the first experiment. Quantitatively, when comparing the ARIs between this representation in this experiment against the previous experiment it shows that the performance has degraded. This means that discerning healthy controls from BBB patients is harder than discerning all diagnostic classes used in the first experiment.

Upon inspection of Figure 5.10 it looks like the reconstruction quality of both representations is about equal. Quantitatively, Table 5.1 reads that the test losses of the concatenated representation are about twice as high as the summed-up representation. This can be explained by arguing that from the reconstruction visualizations, it looks like the concatenated representation is a more chaotic signal, making it harder to reconstruct. The strength of the concatenated signal is that the model is able to see the difference in the baseline of the different leads, this can be used to determine in what general direction the heart is beating, which is an indication medical experts use to determine abnormal heart functioning. In the summed-up representation, this information is lost. The summed-up representation however, shows a much clearer image of the course of the heartbeat. From Table 5.1 it looks like both models score about even on performance metrics, with the concatenated model having a larger range of uncertainty. When inspecting the PCA embeddings in Figure 5.11, however, the low dimensional representations of the concatenated signal are much better discernible. The cluster density seems to have a radial shape with the BBB patients on the outside and the healthy controls on the inside. The K-means clustering is at fault here, this algorithm is bad at clustering this type of distribution of clusters. From both the PCA visualization Figure 5.11 and the t-SNE visualization Figure 5.12 it can be seen that the quality of the low dimensional representation with respect to heart pathology is better for the concatenated representation.

5.4.3. One-Dimensional convolutional representations

The one-dimensional convolutional representations consist of two representations; the element-wise summed-up representation, which is a single signal consisting of the summed-up elements of all the leads, and the other representation, which is one where the leads are fed into the one-dimensional convolution layers as separate channels.

Table 5.1 shows that quantitatively, the 12-channel representation outperforms the summed-up representation by a large margin. The reconstruction plots in Figure 5.14 show similar reconstructions, which is confirmed by the losses noted in Table 5.1. The PCA plots of the low dimensional representation however show that the clustering for the summed-up representation is again one of a radial shape. The k-means algorithm is bad at clustering these types of clusters, skewing the quantitative results. The t-SNE plots of the low dimensional representations underline this phenomenon, but they also show that a very low number of BBB patients are discernible from the radial core of healthy controls mixed with BBB patients. The accuracy over training epoch visualizations seen in Figure 5.17 show two things; firstly these visualizations suffer from the same skewing that the quantitative accuracies from Table 5.1 suffer from, secondly, there is no trend visible that indicates better performance with longer training.

5.4.4. Image fields representation

The image fields representation is a single representation. Visualizations of the reconstructions show that the trends are properly reconstructed, but the overall images are a bit washed out compared to the input. Both the PCA and t-SNE visualizations show again a radial structure in the low-dimensional representation. This means that the performance metrics, again, are skewed by the bad combination of the clustering algorithm and cluster structure. In this case, the t-SNE visualization reveals that the core of the radial structure consists of only BBB patients, while the surroundings are a poorly discernible mix of both classes.

Finally, an important note here is that the image fields are produced from an element-wise summed-up lead. From previous data representations in this experiment, it is known that this transformation removes the information needed to determine the general direction the heart is beating in, in three dimensions. An image fields representation produced by concatenating all leads into one single lead could produce better results.

6

Deep Vision Models

6.1. Introduction

The previous experiments show that even with the best-performing model and data input representation, the low-dimensional ECG representations are not sufficiently discernible. The cluster plot indicates that the degree of overlap is too high for any clustering algorithm to cluster different heart pathology in different clusters consistently. It is hypothesized that this is because the feature extraction model can not capture the complex non-linear properties that define heart pathology. This leads the search for an unsupervised ML model to an autoencoder that is capable of capturing more complex properties. One such model is the research by [11]. Their model called Resnet is truly a deep learning model because of the large number of layers. Because of the deep nature of Resnet, it is hypothesized that features that define heart pathology can be extracted, hypothetically leading to improved performance on the task of unsupervised classification of heart pathology.

6.2. Extra Literature

The only problem is that ResNet[11] is not suited for the reconstruction objective. In order to use it as an unsupervised neural network, an autoencoder needs to be constructed following the main principles and findings as in [11]. [11] Found that when forming larger neural networks, a problem occurs. The models are not capable of being trained. This is because of the vanishing and exploding gradient problems. The intuition behind these problems is that if there are many layers, gradients get multiplied many times. In the cases where individual gradients are larger than 1.0, this means that gradients for parameters near the input part of the network explode. The consequence of this is that the learning mechanism of gradient descent does not converge and the network will consequently not learn to minimize its loss objective. In the cases where the individual layer gradients are smaller than 1.0, this means that because of the many multiplications the gradients for the parameters at the input side of the network will vanish, quickly stagnating the learning mechanism of gradient descent. The solution to these problems was found to be so-called skip connections or residual connections. These connections

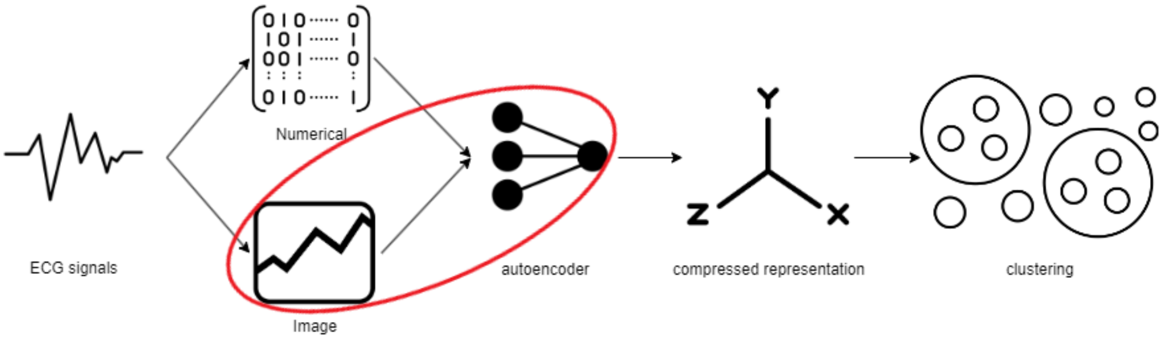


Figure 6.1: Subject of experiments for this section.

Table 6.1: Quantitative results of deep autoencoders.

autoencoder model	train accuracy	test accuracy	train loss	test loss	train NMI	test NMI
ResNet50	0.51	0.51	0.32	0.8	0.00	0.00
InceptionResNetv2	0.53	0.52	0.08	0.09	0.17	0.18

allowed an identity mapping of the input features parallel to the main structure of the network. The skip connections are recombined with the features flowing through the main structure of the network every two or three convolution layers, depending on the type of block used. A basic block consists of two convolution layers and is used in the shallower versions of ResNet, while a bottleneck block consists of three convolution layers and is used in the deeper versions of ResNet.

In [24] are inspired by the work of [19] and [2]. In [19], network blocks are defined that do dimensionality reduction within parts of the network in order to save computing resources. [19] Do this by means of 1x1 convolutions. In [2] sparse structures in the convolutions are explored. [24] Take the findings of both researches and come up with an architecture that makes use of the highly optimized dense convolutional layers to approximate sparse structures. They do this by making use of the 1x1 convolutional layers. In later research, [25] improve on this concept by constructing larger networks with the same principles. The two best-performing architectures of [25] are Inceptionv4 and InceptionResNetv2. The second network adds the residual layers from [11] to the Inception architecture. Eventually, both networks reach the same performance, but InceptionResNetv2 does so earlier. In this comparison, the residual connections seem to only speed up training, since Inception architectures in general are able to deal with the exploding and vanishing gradient problems.

6.3. Methodology

Different autoencoders are built with designs of different lengths of ResNet and a single version of InceptionResNetv2. They are constructed by using the ResNet and InceptionResNetv2 models as encoders and using inverted versions of those encoders as decoders. The autoencoders are then used and compared to the basic convolution autoencoders from the previous experiment on the concatenated plotted image representation.

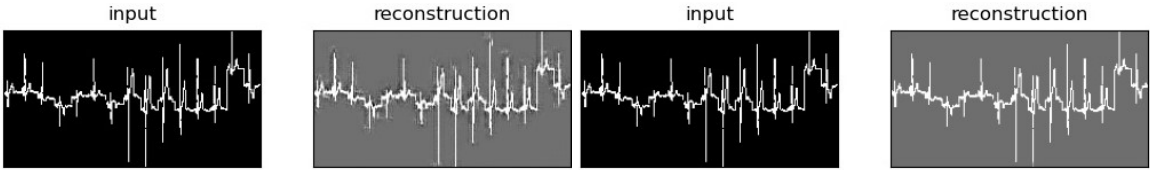
6.3.1. Hyper Parameters

The experiments are run with a replication factor of only one because a single training run takes about three days to complete. The other hyperparameters consist of an Adam optimizer with an initial learning rate of 1.0×10^{-3}, 120 epochs training, and an embedding size of 10.

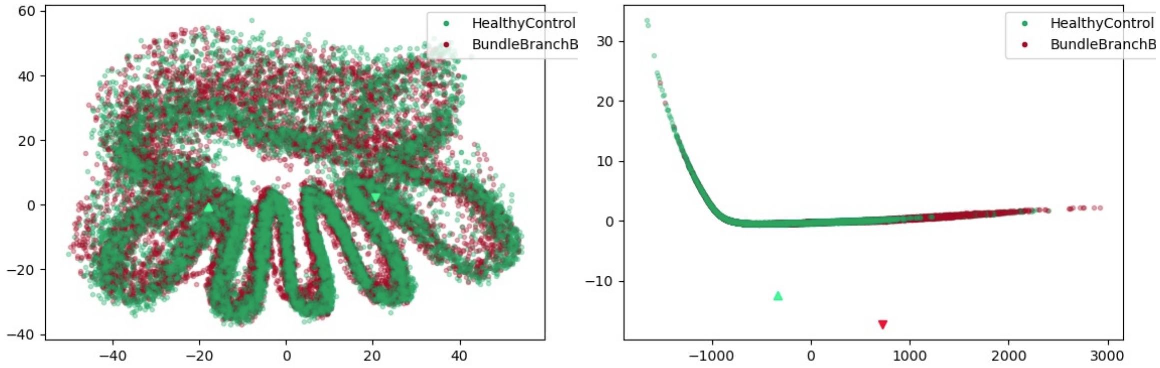
6.4. Discussion

Both the ResNet50 and the InceptionResNetv2 have excellent reconstructions, but also both models produce very poorly discernible clusters. For autoencoders, it is known that the dimensionality of the encoding is key for the bottleneck mechanism to produce salient features. A dimensionality that is too high will turn the bottleneck into an identity mapping, causing it to not produce salient features anymore.

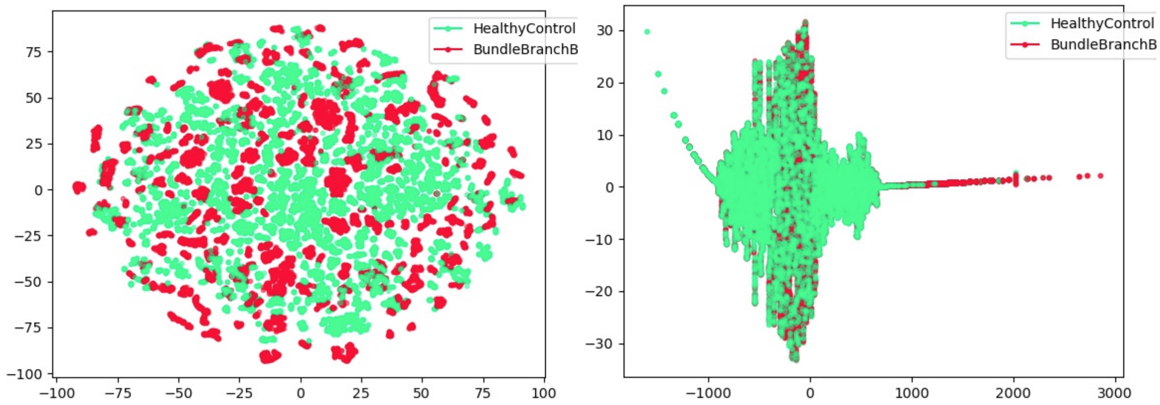
It is hypothesized that complex decoders in autoencoders have the same effect as large encoding dimensionalities. The reasoning is that more complex decoders are more flexible and can more easily reconstruct the input from the low dimensional encoding. For example, when a simple decoder would only just be able to reconstruct an input from an encoding with a certain dimensionality, a complex encoder could do the same with a yet lower-dimensionality encoding. When the encoding dimensionality would instead be kept the same, that difference in dimensionality needed for a good reconstruction results in the complex decoder having redundant encoding values compared to the simple decoder. These redundant values, in turn, negate the bottleneck mechanism.



(a) ResNet reconstruction **(b)** Inception reconstruction



(c) ResNet PCA **(d)** Inception PCA



(e) ResNet t-SNE **(f)** Inception t-SNE

Figure 6.2: ResNet50 vs InceptionResNetv2

7

Recurrent Numeric Models

7.1. Introduction

Recurrent neural networks (RNNs) are known to be able to better capture the underlying nature of sequential processes. They are known to work well on natural language problems and time series. ECGs are time series. The hypothesis is; autoencoders created by recurrent units are better able to extract the underlying features of the ECG signals, resulting in the extracted low-dimensional representations being more correlated to heart pathology.

7.2. Extra Literature

[7] Created a recurrent neural network (RNN) by first compressing an ECG vector of length 1024 to a feature vector of length 32. This vector is fed into a long-short-time-memory (LSTM) unit of eight stacked layers and 32 time steps. The output of the LSTM layer is interpreted as the encoding of the autoencoder. In the decoder, only convolutional layers, upsampling layers, and one fully connected layer are used. No LSTM is used in the decoder, making it different from the usual inversion of the encoder. The network is trained in a way that it also does denoising. The purpose of the research is to create a model for compression in order to alleviate transmission costs and eliminate some of the noise that comes with the transmission of ECG data. The resulting compression ratio is 64 and the performance evaluation of the decompressed ECG as a consequence of noise and compression is expressed in a metric called quality score with a value of 15.61. This quality score is higher than the state-of-the-art at the moment of publication, which is 2021.

7.2.1. LSTM-based arrhythmia classification

[13] Created a supervised LSTM-based autoencoder with a support vector machine (SVM) attached to the encoding. In their research, the authors argue that the autoencoder is capable of extracting relevant features. This is motivated by the high accuracy attained of 99.7%.

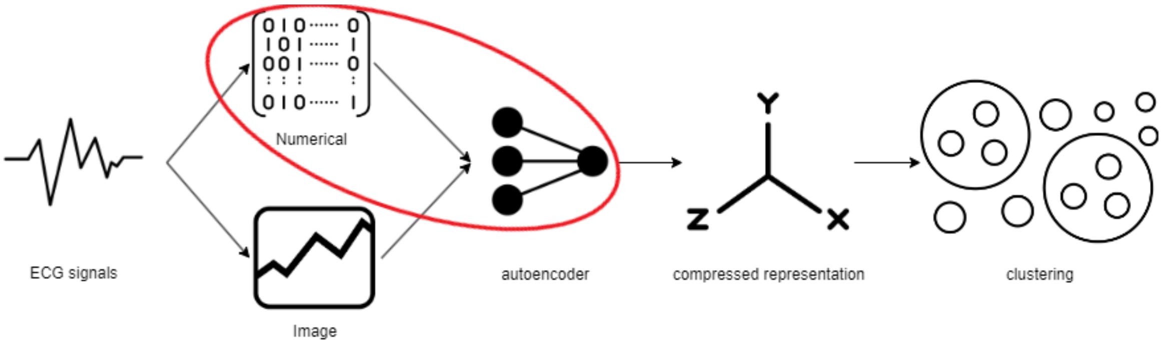


Figure 7.1: Subject of experiments for this section.

Table 7.1: LSTM autoencoders with different sample rates of the input signals.

sample rate (hz)	train accuracy	test accuracy
100	0.53 ± 0.02	0.53 ± 0.02
40	0.54 ± 0.02	0.54 ± 0.01
20	0.55 ± 0.03	0.54 ± 0.03
10	0.53 ± 0.03	0.51 ± 0.01

Table 7.2: LSTM autoencoders with different teacher forcing probability.

teacher forcing probability (%)	train accuracy	test accuracy
0	0.51 ± 0.01	0.53 ± 0.02
20	0.52 ± 0.01	0.55 ± 0.01
40	0.51 ± 0.01	0.53 ± 0.02
60	0.52 ± 0.02	0.54 ± 0.01
80	0.52 ± 0.01	0.51 ± 0.01
100	0.53 ± 0.02	0.54 ± 0.02

7.2.2. LSTM unsupervised video representations

In their research [23] find a solution for feature extraction of videos. [23] came up with a convolutional LSTM-based autoencoder where they experiment with different loss objectives. The authors experiment with the reconstruction of the input sequence. They find that inverting the sequence of the reconstruction improves performance as the model better learns to remember the end of the sequence. The authors experiment with the loss objective of future prediction, which is a loss objective more natural to sequential data and, has not been done on autoencoders that are not based on RNNs. The last loss objective that [23] created was a combination of the preceding two objectives: the model has one decoder reconstructing the reversed input, and one decoder predicting the future. The performance of their research is qualitatively expressed by inspection of the reconstructed and future frames of the video. The authors also quantitatively evaluate their model by means of supervised video classification. One of their findings is that the best features are extracted when using the two decoders.

7.3. Methodology

There are multiple configurations to explore when evaluating RNN-based autoencoders. The objectives of reconstruction, prediction, and the hybrid of both are explored.

Different types of input for the decoding part are experimented with as well. This consists of inputting the output of the previous time step recursively, inputting the ground truth of the previous time step, or a combination of both. The combination is made by drawing from a Bernoulli distribution every time a next time step is performed. For this distribution one needs to define the chance for success parameter p. On a successful draw from the Bernoulli distribution, the input for the next time step will be the ground truth. On a failed draw from the Bernoulli distribution, the input for the next time step will be the output of the previous time step.

For the last category of configurations, it is observed in both [23] and exploratory runs of this research, that the maximum number of time steps that the decoder is able to produce somewhat relevant predictions is 32. After this number of time steps the outcome of the decoder time steps is completely arbitrary. In order to predict larger parts of the ECG, one can lower the sample rate of the input vector. In the PTB database the sample rate is fixed at 1000 hertz. In this research, experiments are performed with sample rates of 100 hertz, 40 hertz, 20 hertz, and 10 hertz.

Table 7.3: LSTM autoencoders with different loss objectives.

loss objective	train accuracy	test accuracy
reconstruction	0.52 ± 0.01	0.54 ± 0.03
prediction	0.53 ± 0.01	0.53 ± 0.03
hybrid	0.51 ± 0.01	0.51 ± 0.00

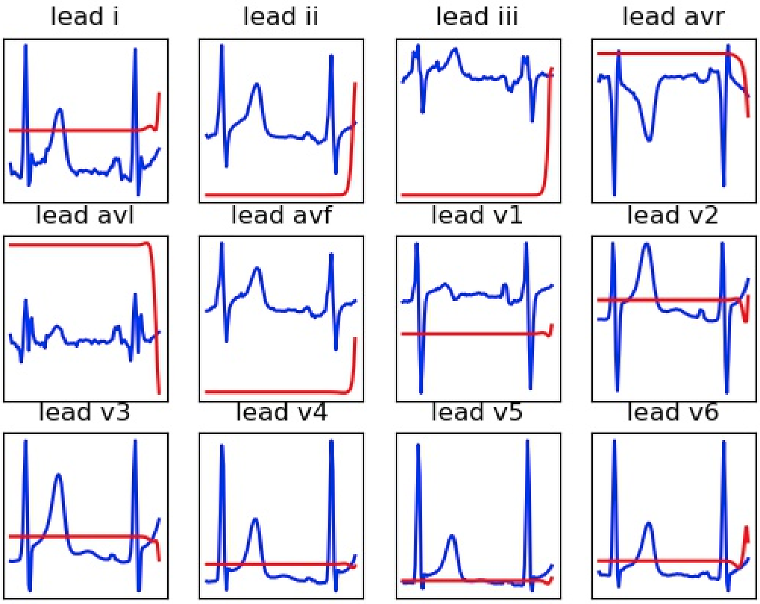


Figure 7.2: Reconstruction of the entire ECG object.

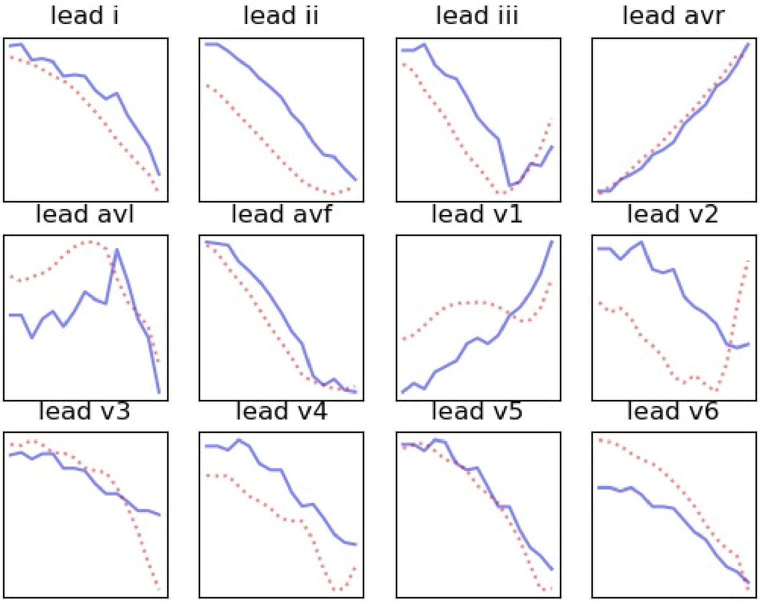


Figure 7.3: Reconstruction of the last 15 steps of the ECG object.

7.4. Discussion

In the first set of experiments, the effect of the sample rate of the signal on the accuracy performance of the classification task is researched. Results show that varying this part of the signal has little effect on the eventual accuracy reached, but the best-performing sample rate is 40 hertz. The reasoning is that this sample rate results in the joint best test classification accuracy, but with a more narrow spread in test accuracy, making it more reliable than its competitor of 20 hertz.

In the second set of experiments, the effect of the teacher forcing probability is researched. Similar to the first set of experiments, the different options do not result in a dramatically different outcome of evaluation metrics. The best performing teacher forcing probability from the available options here is 20%. Not only does this result in the highest test accuracy, but the same metric is also achieved with a joint lowest standard deviation of 0.01%.

In the last set of experiments, the different loss objectives are researched similar to the research by [23]. Similar to all preceding LSTM experiments, the different options make only a small difference in outcome of the evaluation metrics. The best-performing loss objective seems to be reconstruction, with prediction being a close runner-up. The hybrid reconstruction-prediction loss objective seems to be by far the worst in this comparison, which is surprising since it is the exact opposite of what [23] found.

From the first and second rounds of experiments, Tables 7.1 and 7.2 were produced. For these experiments, the entire ECG was used for input of the decoder. For the last set of experiments only 15 steps were used in the decoder part, this improves the decoder output as can be seen in Figures 7.2 and 7.3.

8

Variational Numeric models

8.1. Introduction

Recent research [17] on variational autoencoders shows that the design of a variational autoencoder can generate explainable features when applied to ECG data. Explainable features are generally regarded as good features in the machine learning community. In this case what is meant with explainable features is that when the authors fix 24 out of the 25 features of the encoded representation of an ECG and vary the feature that is not fixed, a visible quantity of the ECG is changed, for example when changing varying one feature, the height of the QRS complex varied in the same direction consistently.

It is hypothesized that features generated by the exact model from [17] are good in the sense they can be used to discriminate ECG data based on underlying heart pathology.

The key quantitative result that [17] found was a metric called maximum mean discrepancy (MMD). The MMD is a metric that describes dissimilarity between two populations of vectors. A lower value means the two populations are more similar. In the research by [17], the value of this metric was found to be 3.83×10^{-3}. Generation of this metric requires one hyperparameter. The way [17] set this hyperparameter is said to be the same as described in [8]. In [8], the generation of artificial ECGs is done by adversarial neural networks. They achieve an MMD of 1.05×10^{-3}.

8.2. Methodology

In order to find the answer to the aforementioned hypothesis, first, an attempt will be made to reproduce the research [17]. Following the attempted reproduction is an evaluation that grants insight into the quality of the extracted features with respect to the discernibility of heart pathology.

8.2.1. Data Set

The data set used in this experiment is the Lobachevsky university electrocardiogram database (LUDB). In this database ECG signals are segmented. The segmentation marks part of the ECGs as P-wave, QRS

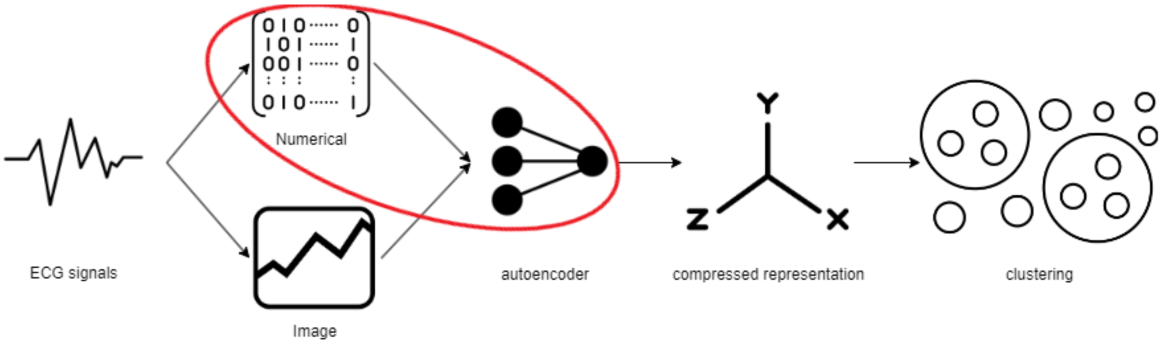


Figure 8.1: Subject of experiments for this section.

complex, and T-wave. This allows the authors of experiments to center the QRS wave in every ECG object, which is also the case for [17]. Instead of element-wise summing of corresponding leads, every lead is used as an ECG object by itself. Compared to other data sets like the PTB data set that is used for all preceding experiments, the LUDB data set contains more annotation information. The heart pathology annotations are split into three different groups; the electrical axis of the heart, hypertrophy, and heart rhythm. Plots are made for all these three different categories, and one for examining whether the low dimensional features can discern male from female.

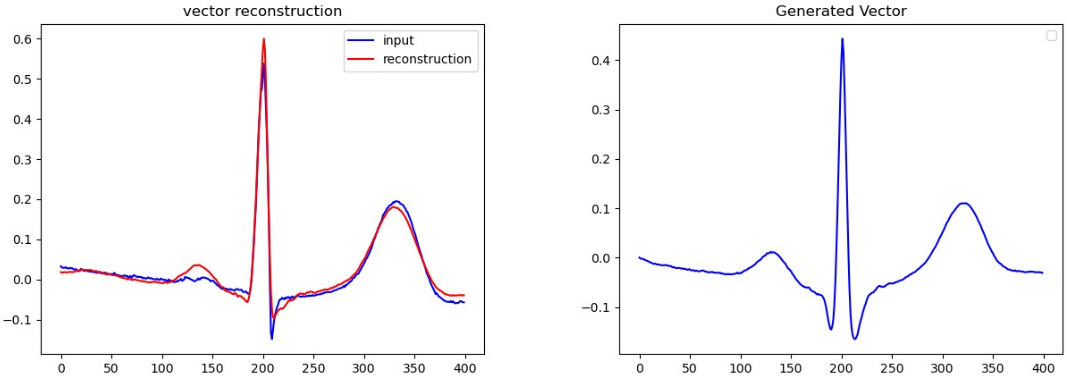
8.2.2. Hyperparameters

The model is trained and evaluated on 750 epochs of training, a low-dimensional representation size of 25, and the optimizer is chosen to be Adam just like the research by [17]. The initial learning rate for the optimizer is not mentioned, which is why it is set to be a default value of 1.0×10^{-3}. The batch size was not specified by [17] and is set to 256 since that is often used in preceding experiments in this research. The replication factor is three.

8.3. Discussion

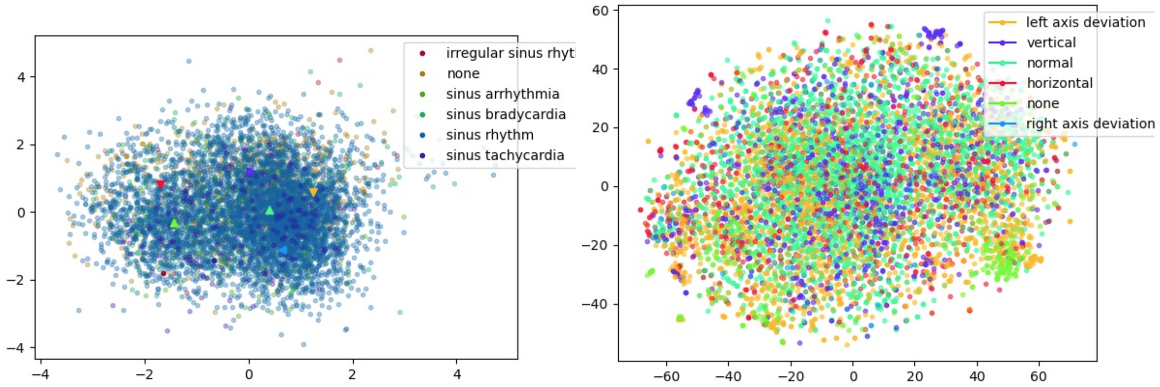
The reconstruction and generation of the ECG vectors as seen in Figure 8.2 look good. The reconstruction shows a low error and the generated ECG vector looks like a real ECG signal which is confirmed by medical experts at the Erasmus medical center. The MMD value attained is 2.4×10^{-3}, which is better than [17], which is being reproduced by this experiment, and worse than [8], which is expected. Since the batch size and the initial learning rate of the optimizer are not known, an exact reproduction was never possible. The authors are of Russian nationality, meaning that current geopolitical circumstances prevent additional communication about hyperparameters.

Plots of the low-dimensional representations of the ECGs, which are colored according to the available annotations show that the hypothesis that the low-dimensional representations would be able to discern heart pathology is not true. The additional plot showing discernibility of sex shows that sex is not discernible either.



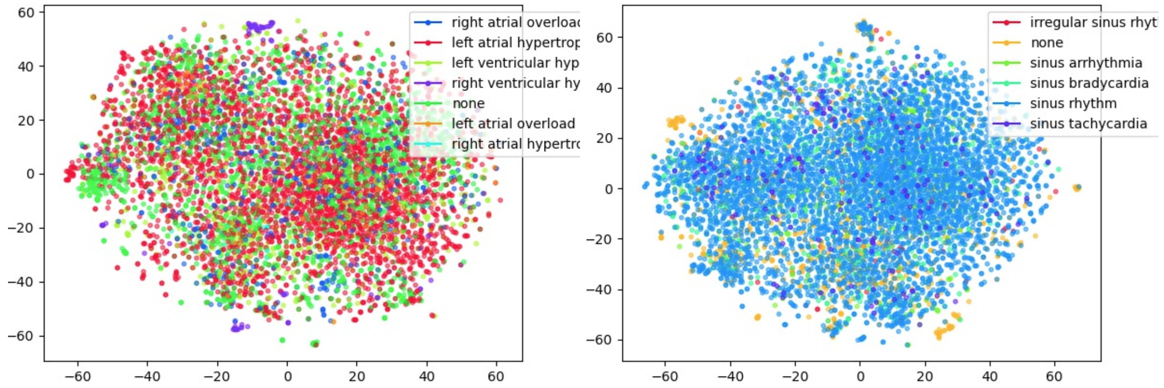
(a) Reconstruction.

(b) An artificial ECG signal constructed by the trained VAE.



(c) PCA visualization of different heart rhythm problems.

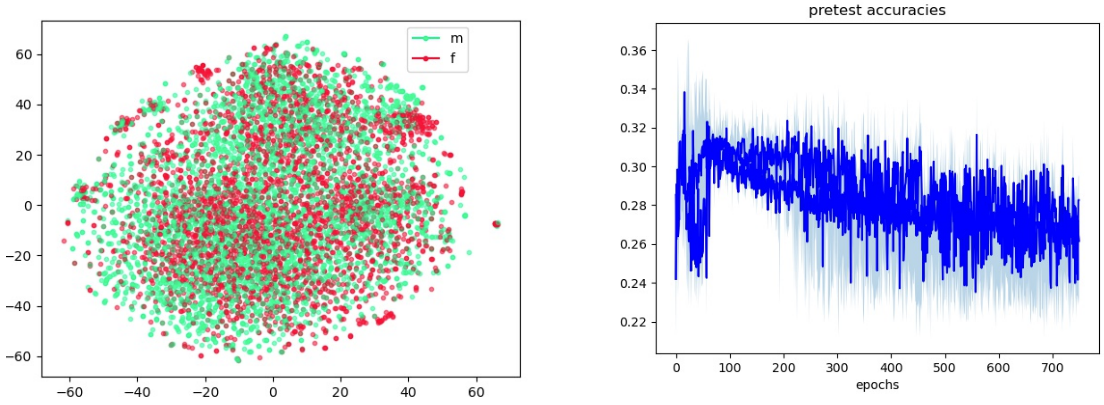
(d) t-SNE visualization of different axes of the heart problems.



(e) t-SNE visualization of different hypertrophy problems.

(f) t-SNE visualization of different heart rhythm problems.

Figure 8.2



(a) t-SNE visualization of different sexes of the patients.

(b) Test accuracy vs epoch trained.

Figure 8.3

9

Clustering Techniques

9.1. Introduction

Previous experiments yielded very poor results up until now. The hypothesis of this design space is that the clustering algorithm with its parameters is not a good fit with the structure of the actual clusters in the low-dimensional space. The low-dimensional data points produced by autoencoders contain clusters that are not radial in shape, but much more complex. These oddly shaped clusters are called manifolds. Three clustering strategies that are more adept at this type of clustering are DBSCAN, OPTICS, and Hierarchical clustering. Clustering can be further improved by assuming a many-to-one mapping from cluster labels to heart pathology labels. The best number of clusters could be inferred from the hierarchical clustering.

9.2. Methodology

In the experiments, multiple clustering algorithms are examined, while varying their hyperparameters. [18] argues that if there are more than 25 clusters, the entire pipeline will stop being meaningful. This has to do with the way the pipeline is evaluated. Like before, the data is fed into an autoencoder, which produces low-dimensional data. This low-dimensional data is used to cluster. This process yields cluster labels, which are only numbers, data points from the same cluster get the same cluster label. The aim of the evaluation is to see if the formed clusters are related to heart pathology. In order to get any meaningful metric at all, the cluster labels are mapped to the heart pathology annotations based on the heart pathology that is most abundant in each cluster. This means that the evaluation pipeline makes use of the heart pathology annotations as labels, making it supervised. The emphasis is on the fact that neither the clustering algorithm nor the autoencoder makes use of the heart pathology annotations during training, in other words, the model is trained completely unsupervised, but the evaluation is supervised. This is used in all similar research, where unsupervised feature extraction is evaluated [18, 21]. A direct consequence of the supervised nature of the evaluation is that if the clustering algorithm produces many clusters, the accuracy will always converge to 1.0. In the extreme case that all data points

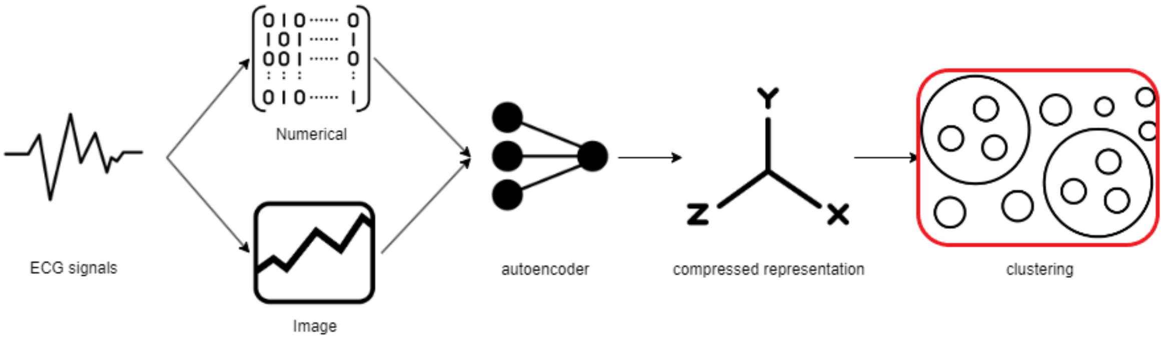


Figure 9.1: Subject of experiments for this section.

are their own cluster, the data points will all get mapped to their most abundant heart disease, which is the heart pathology annotation of their only data point, resulting in a perfect accuracy of 1.0. This makes clusterings with high numbers of clusters yield only trivial information. With that in mind and the fact that [18] uses 25 clusters as the upper limit, the experiments here will only look at clusterings with less than 50 clusters. This is more than 25 because the objective of this experiment is to see if there is any potential in the different clustering techniques at all.

The experiment will investigate five different clustering techniques while varying hyperparameters. The first two techniques are k-means clustering and hierarchical clustering. For these two techniques, the number of clusters will be varied from 2 to 50. Then the grid-based clustering algorithm of self-organizing maps will be researched. The hyperparameters are the number of rows and columns that make up the grid. These will be taken all square, so they will be 4, 9, 16, and 25 grid units. As last group of clustering algorithms, 2 manifold clustering algorithms will be investigated. They are called DBSCAN and OPTICS. Both algorithms have two hyperparameters, which are the maximum distance between two data points and the minimum number of neighbors needed for a data point to become a core point.

All these five clustering techniques will be performed on the low-dimensional data produced by two of the best-performing data representations from the previous experiment. These are the concatenated image plot format and the 12-channel 1-dimensional convolution format.

9.3. Discussion

For the one-dimensional convolutional format, it is clear that a larger number of clusters results only in marginal gains of accuracy. The heat maps created for the hyperparameter searches of DBSCAN and OPTICS show that even when hyperparameters are chosen by looking at the test labels, there is no potential for high accuracy. In the case of the one-dimensional convolutional format, the legend on the right of the heat maps shows that the maximum attainable accuracy is quite poor; there are no hyperparameter combinations that result in an accuracy of even 0.6.

For the concatenated plotted format the results are very different. Between k-means, agglomerative clustering, and self-organizing maps (SOM), agglomerative clustering performs best. This is in line with the observation that the clusters are of complex shape. The DBSCAN seems to be doing best with accuracy values of around 0.9 according to the legend on the right of the heat map. This is somewhat misleading, because when hyperparameters are chosen accordingly; the number of clusters is higher than 7000. This is not an interesting result since the focus here is on clusterings that have 25 clusters or fewer.

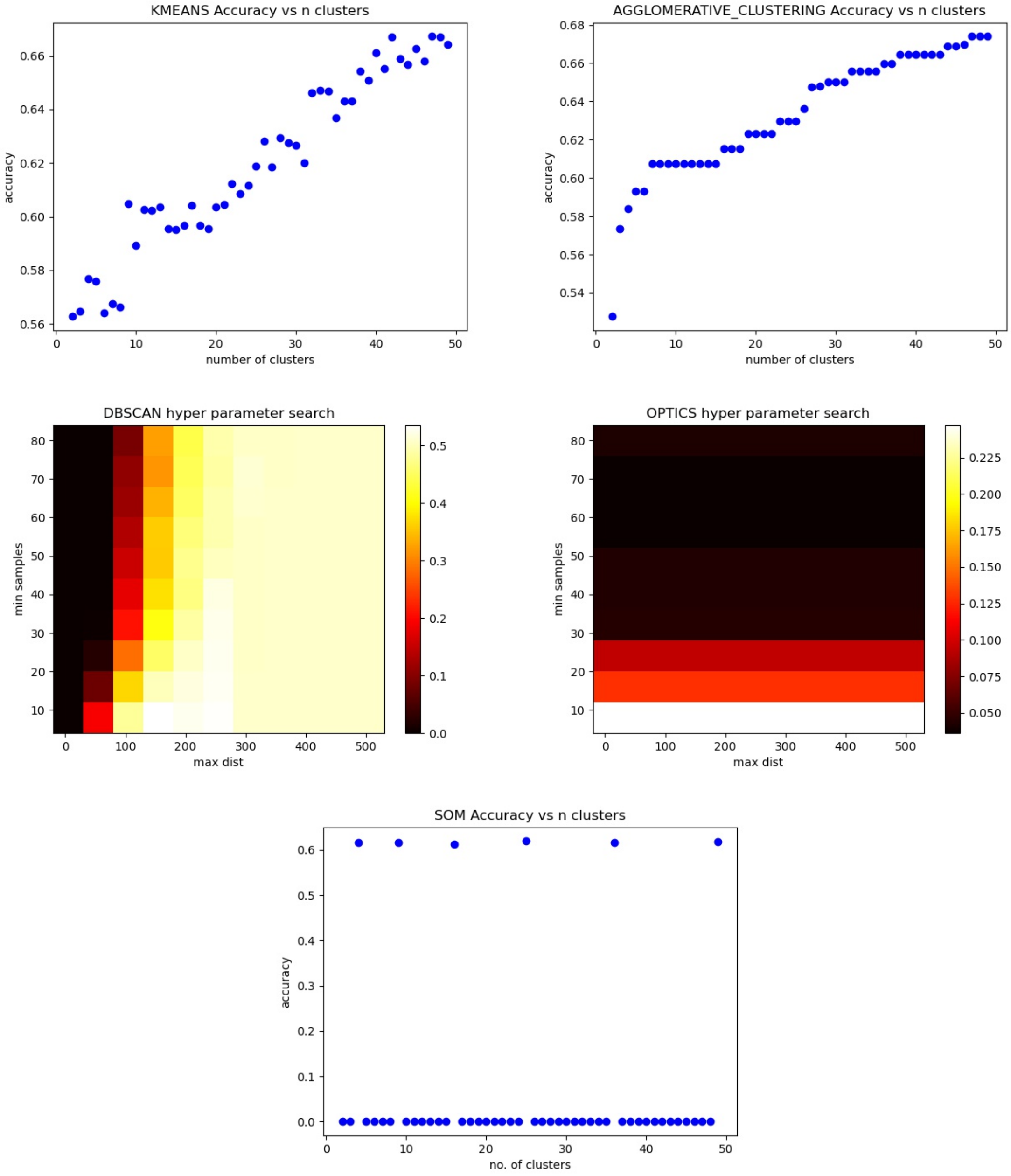


Figure 9.2: 12-channel 1-D convolutional format clustering

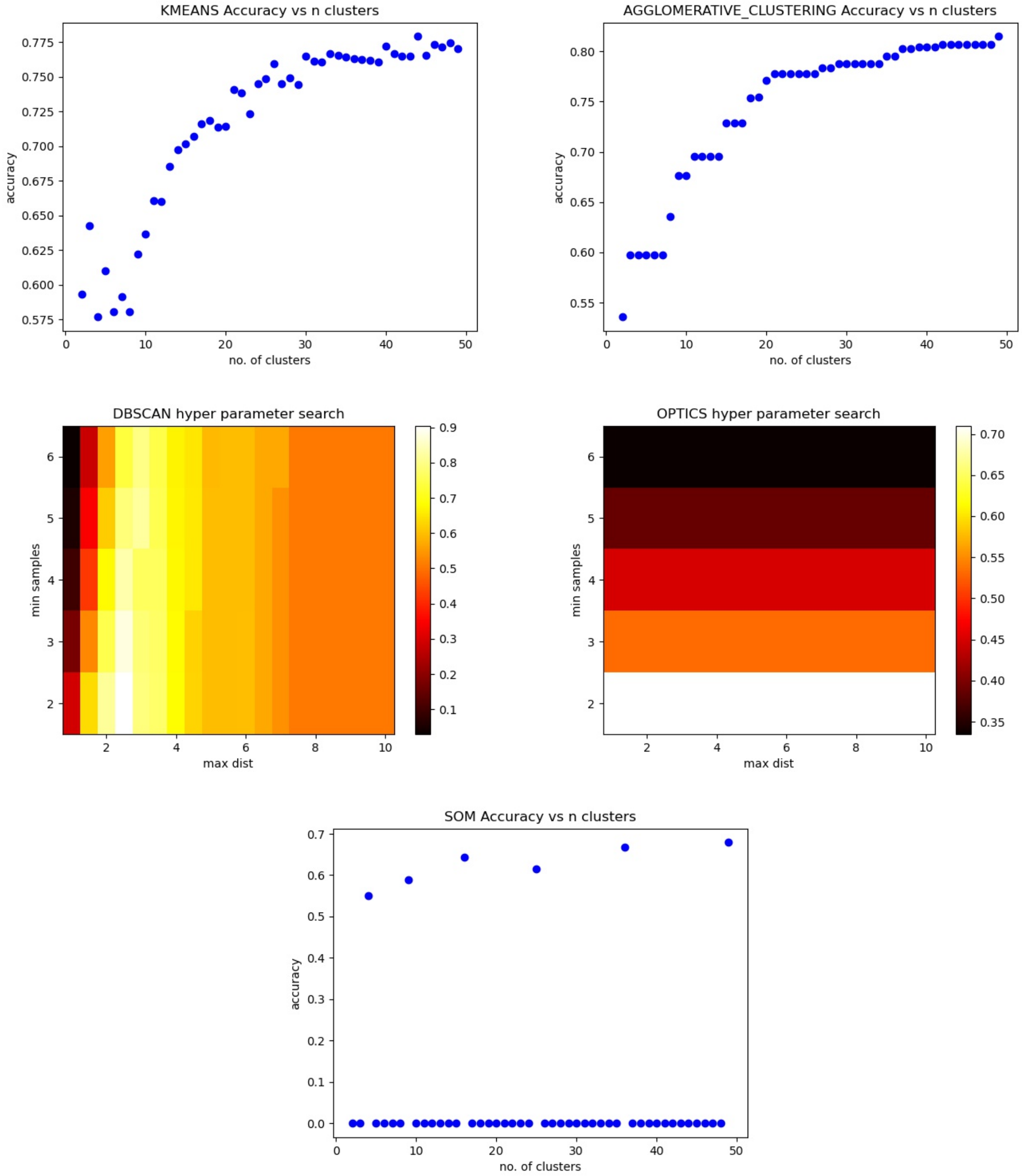


Figure 9.3: concatenated plot format clustering

10

MIT Dataset

10.1. Introduction

The research by [18] manages to run its model on the MIT BIH ECG beat data set and score an accuracy of 98.5%. Their model uses unsupervised manual feature extraction followed by SOM with 5 rows and 5 columns. The hypothesis of this research is that automatic unsupervised feature extraction by autoencoder produces features that are at least as good as the manually extracted features by [18] resulting in the same or better accuracy metrics in the classification task on the MIT data set.

10.2. Methodology

The MIT data set is an ECG data set where every heartbeat is annotated separately, in contrast to the data set previously used in this research, where the heart pathology annotations are only available per patient. The MIT data set contains 16 classes of heart pathology and one class that represents healthy controls. [18] remove all beats annotated with the 'paced beat' annotation, because this is visible in the p wave, while their manual feature extraction method only uses the QRS complex. Contrary to what is done in this research, [18] does not correct for the skewed class distribution in the MIT data set. This means that about 70% of the beats in the MIT data set are annotated as healthy beats. The setup used to compare this research to [18] consists of parts that are either the best according to previous experiments of this research, or the same as in [18] for the best comparison. The input representation is the concatenated plotted image representation, the autoencoder is the basic convolutional autoencoder, and the clustering algorithm will be SOM with 5 rows and 5 columns as implemented by the SuSi python package. Similar to preceding experiments the low dimensional representation will be of size 10 and the experiment reproduction factor will also be 10. The number of training epochs is fixed at 50 and the optimizer is chosen to be ADAM with an initial learning rate of 0.001.

10.3. Discussion

The automatic unsupervised feature extraction described in the preceding methodology chapter scores a disappointing 0.76 ± 0.01. From the t-SNE visualization in combination with the found NMI of

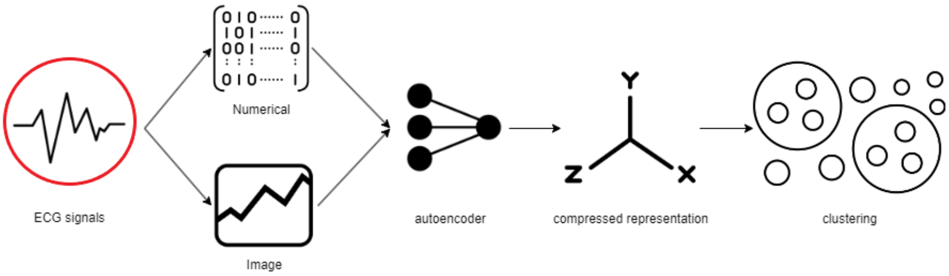


Figure 10.1: Subject of experiments for this section.

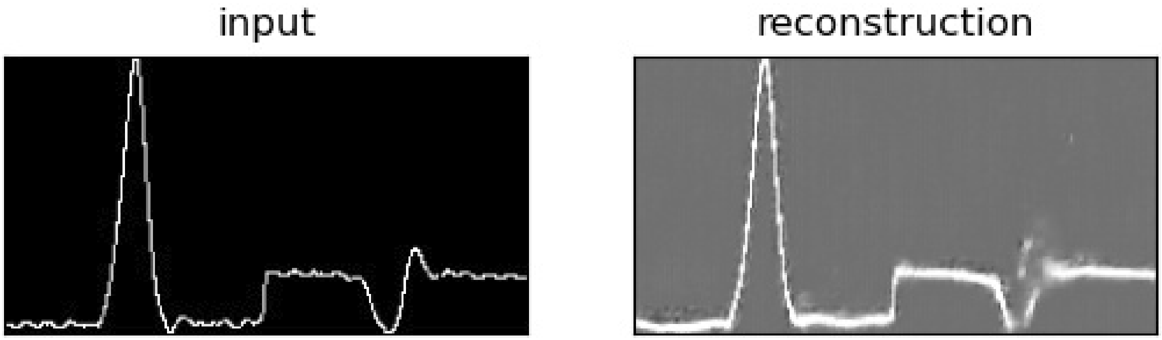


Figure 10.2: Reconstruction of the concatenated plot representation on the MIT data set.

0.53 ± 0.01, it becomes clear this at least for the healthy beats, the model is able to recognize and cluster together different beats from the same patient relatively well. The ARI is a performance metric which is closely related to accuracy, but always on a scale from 0 to 1 regardless of the label distribution. The ARI of this experiment comes down to a meagerly 0.29 ± 0.05. This means that this model is better at recognizing the same patient than it is at recognizing the same heart pathology.

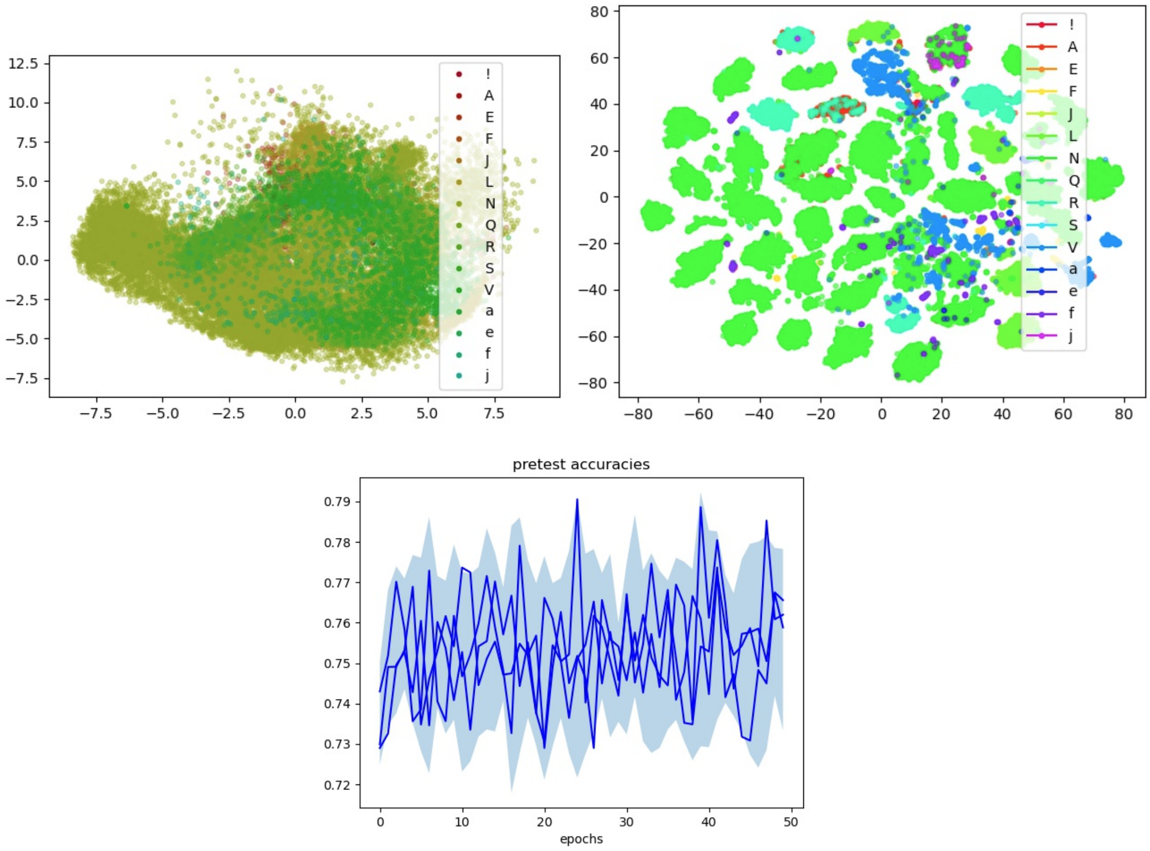


Figure 10.3: MIT visualizations

11

Conclusion

In this concluding section, the main contributions and findings are presented. Finally, suggestions will be made for future research.

11.1. Contributions

In this research unsupervised automatic feature extraction of ECG signals is compared to both supervised automatic feature extraction by classification neural networks and unsupervised manual feature extraction followed by SOM clustering. The main finding is that unsupervised automatic feature extraction performed by various types of autoencoders on various input representations followed by various types of clustering does not perform as well on the task of classification as the aforementioned machine learning pipelines. Before coming to this conclusion, extensive experimentation has been done with the aforementioned parts of the machine learning pipeline.

Furthermore, there are two minor contributions. The first is a contribution in the context of 1-dimensional convolutional neural networks and ECG signals. The idea that different ECG leads can be used as different channels for 1-dimensional neural networks is a novel idea that could also be used in supervised deep learning on ECG data because in this research it yielded better results than the alternative known from the literature [10], which does the element-wise summation of the different leads. It is hypothesized that this alternative way to use different leads conserves the directional information.

Similar to the previous contribution, the idea of concatenating the ECG leads for the plotted image representation, instead of element-wise summation before plotting has the advantage of keeping the directional information intact. This is an additional contribution to the existing ([14]) idea of plotting signals, mapping this signal classification task to the domain of computer vision.

11.2. Recommendation Future Research

In this section, three recommendations for future research are done. The first possibility for future research originated from the idea that a combination of both manual feature extraction and automatic feature extraction might be better than either of them in isolation. In their research [18] combine the features extracted by fitting the polynomials to the QRS complex with two manual features. These are two measures related to the RR interval. What these two measures are exactly is not known since this is exactly how it is framed in [18], but including these features in the features found by the autoencoders might yield much better performance on the task of heart pathology classification.

The second recommendation has to do with the question of what features are extracted from the ECG by the autoencoders. In this research, attention is given to comparing the low-dimensional features to heart pathology annotations and patient id annotations. The autoencoders work by first extracting features into a low dimensional representation and from that low dimensional representation reconstructing the input. The bottleneck mechanism is said to force the encoder to extract salient features that uniquely encode information that allows the decoder to reconstruct unique inputs. This research tried to find the degree to which these extracted features are correlated with heart pathology of the patients, but those are found to be quite weak. The question now remains, what is this kind of information that the autoencoders extract from the ECGs from which it is able to reconstruct the input? Some possible alternatives are patient sex and patient age.

The third and last recommendation is based on the research by [12]. The contributions here are twofold; [12] come up with autoencoders that are asymmetrical, where the encoder is complex and the decoder is relatively simple. The second contribution is that [12] mask 75% of the input while keeping the reconstruction completely intact. [12] mention that transfer learning using this approach as pretraining outperforms related supervised pretraining. Taking the very poor performance found by this research, the models by [12] are not expected to perform better than supervised heart pathology classification or manual feature extraction, but the possibility still exists.

References

[1] Anonyma. *12 Lead ECG*. `https://medizzy.com/feed/4178130`. [Online; accessed 08-August-2022]. 2018.

[2] Sanjeev Arora et al. "Provable bounds for learning some deep representations". In: *International conference on machine learning*. PMLR. 2014, pp. 584–592.

[3] Agateller (Anthony Atkielski). *SinusRhythmLabels*. `https://commons.wikimedia.org/w/index.php?curid=1560893`. [Online; accessed 09-August-2022].

[4] R Bousseljot, D Kreiseler, and A Schnabel. "Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet". In: (1995).

[5] Andriana SLO Campanharo et al. "Duality between time series and networks". In: *PloS one* 6.8 (2011), e23378.

[6] Cburnett. *Own work in Inkscape*. `https://commons.wikimedia.org/w/index.php?curid=1840682`. [Online; accessed 09-August-2022].

[7] Evangelin Dasan and Ithayarani Panneerselvam. "A novel dimensionality reduction approach for ECG signal via convolutional denoising autoencoder with LSTM". en. In: *Biomedical Signal Processing and Control* 63 (Jan. 2021), p. 102225. ISSN: 17468094. DOI: `10.1016/j.bspc.2020.102225`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S1746809420303554` (visited on 12/02/2021).

[8] Anne Marie Delaney, Eoin Brophy, and Tomas E Ward. "Synthesis of realistic ECG using generative adversarial networks". In: *arXiv preprint arXiv:1909.09150* (2019).

[9] Xifeng Guo et al. *Deep Clustering with Convolutional Autoencoders*. Pages: 382. Oct. 2017. ISBN: 978-3-319-70095-3. DOI: `10.1007/978-3-319-70096-0_39`.

[10] Nahian Ibn Hasan and Arnab Bhattacharjee. "Deep Learning Approach to Cardiovascular Disease Classification Employing Modified ECG Signal from Empirical Mode Decomposition". en. In: *Biomedical Signal Processing and Control* 52 (July 2019), pp. 128–140. ISSN: 17468094. DOI: `10.1016/j.bspc.2019.04.005`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S1746809419301028` (visited on 08/20/2021).

[11] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[12] Kaiming He et al. "Masked autoencoders are scalable vision learners". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16000–16009.

[13] Borui Hou et al. "LSTM-Based Auto-Encoder Model for ECG Arrhythmias Classification". en. In: *IEEE Transactions on Instrumentation and Measurement* 69.4 (Apr. 2020), pp. 1232–1240. ISSN: 0018-9456, 1557-9662. DOI: `10.1109/TIM.2019.2910342`. URL: `https://ieeexplore.ieee.org/document/8688435/` (visited on 12/13/2021).

[14] Tae Joon Jun et al. "ECG arrhythmia classification using a 2-D convolutional neural network". In: *arXiv:1804.06812 [cs]* (Apr. 2018). arXiv: 1804.06812. URL: `http://arxiv.org/abs/1804.06812` (visited on 10/27/2021).

[15] Anubha Kalra, Andrew Lowe, and Ahmed Al-Jumaily. "Critical review of electrocardiography measurement systems and technology". In: *Measurement Science and Technology* 30 (Nov. 2018). DOI: `10.1088/1361-6501/aaf2b7`.

[16] D Kreiseler and R Bousseliot. "Automatisierte EKG-Auswertung mit hilfe der EKG-Signaldatenbank CARDIODAT der PTB". In: (1995).

[17] V. V. Kuznetsov et al. "Interpretable Feature Generation in ECG Using a Variational Autoencoder". In: *Frontiers in Genetics* 12 (Apr. 2021), p. 638191. ISSN: 1664-8021. DOI: `10.3389/fgene.2021.638191`. URL: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8049433/` (visited on 02/10/2022).

[18] M. Lagerholm et al. “Clustering ECG complexes using Hermite functions and self-organizing maps”. In: *IEEE Transactions on Biomedical Engineering* 47.7 (July 2000), pp. 838–848. ISSN: 00189294. DOI: `10.1109/10.846677`. URL: `http://ieeexplore.ieee.org/document/846677/` (visited on 10/13/2021).

[19] Min Lin, Qiang Chen, and Shuicheng Yan. “Network in network”. In: *arXiv preprint arXiv:1312.4400* (2013).

[20] MoodyGroove. *I captured this 12 lead ECG from a friend of mine during a training class at the fire station.* `https://commons.wikimedia.org/wiki/File:12-lead_ECG.jpg`. [Online; accessed 09-August-2022].

[21] Maziar Moradi Fard, Thibaut Thonet, and Eric Gaussier. “Deep k-Means: Jointly clustering with k-Means and learning representations”. en. In: *Pattern Recognition Letters* 138 (Oct. 2020), pp. 185–192. ISSN: 01678655. DOI: `10.1016/j.patrec.2020.07.028`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0167865520302749` (visited on 08/20/2021).

[22] Nicoguaro. `https://commons.wikimedia.org/wiki/File:Iris_dataset_scatterplot.svg`. [Online; accessed 09-August-2022].

[23] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. “Unsupervised Learning of Video Representations using LSTMs”. In: *arXiv:1502.04681 [cs]* (Jan. 2016). arXiv: 1502.04681. URL: `http://arxiv.org/abs/1502.04681` (visited on 12/14/2021).

[24] Christian Szegedy et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[25] Christian Szegedy et al. “Inception-v4, inception-resnet and the impact of residual connections on learning”. In: *Thirty-first AAAI conference on artificial intelligence*. 2017.

[26] Zhiguang Wang and Tim Oates. “Imaging Time-Series to Improve Classification and Imputation”. en. In: (), p. 7.