# Maastricht University

Department of Advanced Computing Sciences

# Noise Detection and Uncertainty Estimation in ECGs

Internship Report

Parmenion Koutsogeorgos
i6328191

October 26, 2024

**Supervisors**

**University** - Linda Rieswijk
**Organisation** - Rutger van de Leur

Professional internship completed at

# Contents

# List of Figures

# List of Tables

# 1 Introduction

An *electrocardiogram* (ECG) is a simple non-invasive medical test that records the heart's electrical activity over multiple heartbeats, and is widely used in diagnosing and monitoring various cardiac conditions. The main components of an ECG are [1]:

- the *P wave*, which represents the electrical activity occurring in the upper chambers of the heart (atria);
- the *QRS complex*, which reflects the movement of electrical signals through the lower chambers (ventricles);
- the *ST segment*, which indicates the period when the ventricles are contracting without any electrical current passing through them, typically appearing as a flat, horizontal line between the QRS complex and the T wave;
- the *T wave*, which signifies the phase when the lower chambers of the heart are recharging electrically in preparation for their next contraction.

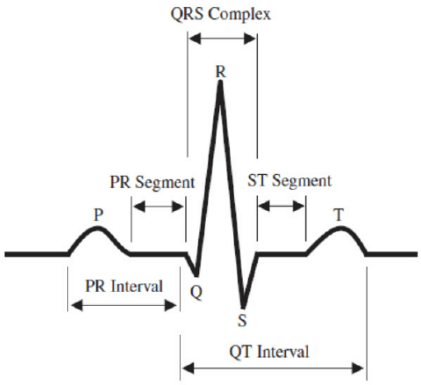



Figure 1: Representation of an ECG signal along with its main components. Source: [2].

An ECG typically consists of 12 channels known as *leads*, which are derived from electrodes placed on the body in positions that provide different views of the heart. Mathematically however, 8 independent leads are sufficient to convey the full information of a typical ECG, while the other 4 leads can be derived from them [3].

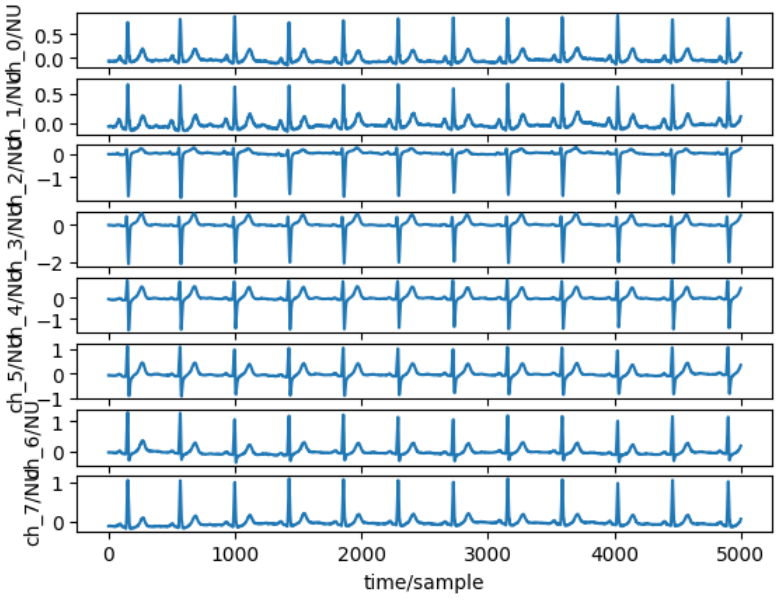



Figure 2: Example of an 8-lead ECG. Source: UMCU dataset (see Section 4.1, *UMCU dataset*).

On the one hand the ECG is easy to produce, but on the other hand it requires expert knowledge in order to be correctly interpreted. This imbalance has a significant negative impact on clinical practice, as there are too many ECGs that need interpretation, and not enough experts to interpret them quickly [4].

In recent years, efforts have been made to automate the process of ECG interpretation using various algorithms. One of the most urgent issues that diagnostic algorithms have to deal with is noise affecting the signal's interpretability. The main types of noise found in an ECG can be classified as follows [5]:

- *Baseline wander:* A low-frequency artifact caused by factors like respiration, body movements, and poor electrode contact. It can distort the ST segment and other low-frequency components of the ECG, leading to misdiagnoses of conditions like myocardial infarction.
- *Power-line interference:* This type of noise is caused by the coupling of power lines during ECG acquisition, resulting in narrowband noise that can distort ECG morphology, particularly the P wave, potentially leading to incorrect diagnoses of atrial arrhythmias.
- *Muscle artifacts* or *electromyogram noise*: Originating from electrical activities in muscles, such as those from movements near the head, these artifacts often overlap with normal ECG frequencies, complicating the recognition of arrhythmias.
- *Channel noise:* Induced when ECG signals are transmitted through channels with poor conditions, such as those affected by Additive White Gaussian Noise (AWGN), this type of noise can significantly affect the interpretability of the ECG.
- *Composite noise:* A combination of various types of noise.

This work aims to apply deep learning techniques in order to detect noisy or otherwise uninterpetable ECGs in an off-line manner, i.e. after they have been acquired. ECGs that fall into this category may be considered *Out-of-Distribution* (OOD) in comparison with the interpretable (in-distribution) ECG. Apart from implementing and evaluating traditional OOD detection techniques, we are interested in whether the uncertainty [6] in a trained diagnostic classifier's predictions can be used as a measure of noise or unintepretability.

This study was conducted as a part of an internship in association with the company *Cordys analytics*[1]. Any decisions made throughout the experimentation process that require medical knowledge were in consultation with the company's Chief Medical/Technical Officer, Rutger van de Leur[2], who is a trained cardiologist.

### 1.1 Research questions

Through this work we aim to answer the following research questions:

**RQ1.** What deep learning OOD detection techniques can be used to detect noisy / uninterpretable ECGs before they enter a diagnostic classifier?

**RQ2.** How can a deep learning model be optimised to distinguish noise from diagnostically important abnormalities in an ECG?

**RQ3.** Once an ECG enters a trained diagnostic classifier, can high uncertainty in the classifier's predictions be used to declare the ECG as noisy / uninterpretable?

## 2 Related work

Noise detection and/or removal in ECGs has traditionally been done by heuristics [7] or signal processing methods [8, 9, 10]. Deep learning techniques appear in the literature for OOD detection in various domains [11], however to the best of the author's knowledge, not many of them have been applied specifically in the context of off-line detection of noisy ECGs.

Various deep learning methods have been applied to the similar task of anomalous rhythm detection, in works such as [12, 13, 14]. While these methods can be attempted also for noise detection,

[1] `https://cordys.health/`
[2] https://www.researchgate.net/profile/Rutger-Van-De-Leur

our task differs fundamentally in the following sense: an ECG that is anomalous because it contains an irregular heartbeat is not necessarily noisy, and if labelled as noisy by our pipeline, then it would be a false positive (cf. RQ2).

Specifically in the context of noise detection in ECGs, [15] introduces a CNN trained on a large private dataset which uses both time and frequency domain information for binary classification of ECGs as "clean" or "noisy", achieving a ROC-AUC of 0.93. On the other hand, [16] uses a CNN to distinguish between "usable" and "unusable" ECG segments by rejecting those with a level of noise that impacts the accuracy of the QRS complex detection, achieving a ROC-AUC of 0.96. The work [17] applies self-supervised feature extraction and feature-space clustering methods for noise detection without the need of labelled data, achieving varying but generally strong results. Other works such as [18, 19, 20] focus on denoising ECGs.

# 3 Methodology

In this work we compare various methods for OOD detection in the domain of ECGs. The methods can be summarised as follows:

- *Supervised methods,* which involve training a binary classifier on noise-labelled data in order to classify ECGs as clean or noisy;
- *Self-supervised methods,* which include:
  - training an autoencoder on clean ECGs and using the reconstruction loss as a measure of noise upon testing;
  - training a feature-extractor on clean ECGs and using clustering methods on the extracted features of the training data in order to detect outliers upon testing;
- *Uncertainty based methods,* which involve training a classifier for diagnostic purposes on clean ECGs, and using the estimated uncertainty in the classifier's predictions as a measure of noise upon testing.

We now explain what each of the above methods entails from a theoretical standpoint. Information about the particular architectures used can be found in Section 6, *Architectures*, while the implementation and experimentation procedure is detailed in Section 7, *Experiments and Results*.

## 3.1 Binary classification

The first method we attempt is simple binary classification. In particular, we train a binary classifier

$$BC : \mathbb{R}^{m \times N} \to \mathbb{R}$$

on $m$-lead noise-labelled ECGs of length $N$ with the aim to classify ECGs as clean or noisy. The input to the network is an $m$-lead ECG $x$ of length $N$, and the output is a single number $y = BC(x) \in \mathbb{R}$, on which the sigmoid function

$$\sigma(y) = \frac{1}{1 + e^{-y}}$$

is applied to obtain the probability of the ECG belonging to the "noisy" class. The class of each sample is determined by comparison with a fixed threshold, chosen based on the network's performance on the validation data.

## 3.2 Self-supervised methods

While binary classification is the natural method to use when labelled data is available, the performance of the model greatly depends on the quality of the labels. This can be problematic in the case of outlier detection in general, and more so when applied on signals like ECGs, because labelling itself can be quite hard (see more in Section 8.1, *Limitations*). Moreover, large labelled ECG datasets are in general scarce, and can often be found only in a private setting (e.g. [15] as well as this work use a private dataset). While labelled public datasets do exist, they might not

always be sufficient for large scale experiments, and artificial data generation is sometimes used to augment the available public datasets (e.g. [17] uses synthetic data).

Self-supervised methods provide an alternative way to detect noisy ECGs without the strong reliance on labelled data. In this work we explore two types of self-supervised methods.

#### 3.2.1 Autoencoder reconstruction loss as a measure of noise

Similar to [12], this method involves training an autoencoder

$$AE : \mathbb{R}^{m \times N} \to \mathbb{R}^{m \times N}$$

for the task of reconstruction of clean $m$-lead ECGs of length $N$. Upon testing, the reconstruction loss $\mathcal{L}(x, AE(x))$ for a test sample $x$ is calculated. Samples with low/high reconstruction loss are classified as clean/noisy respectively. Classification happens according to a pre-determined threshold based on the pipeline's performance on the validation data.

#### 3.2.2 Feature space clustering for OOD detection

Similar to [17], in this method a neural network

$$N = G \circ F : \mathbb{R}^{m \times N} \to \mathbb{R}^n$$

consisting of two parts $F, G$ in sequence, where

$$F : \mathbb{R}^{m \times N} \to \mathbb{R}^d \quad \text{and} \quad G : \mathbb{R}^d \to \mathbb{R}^n$$

(for $d \ll m \times N$) is trained on clean ECGs for a particular task (e.g. classification, reconstruction, contrastive learning, etc.). The second part $G$ (which may be a classification head, a decoder, etc.) is then discarded, and the first part $F$ is used as a feature extractor, i.e. a mapping into the lower dimensional space $\mathbb{R}^d$ which maintains important information about the input. After training, the features extracted from the training set $X_{train}$ are clustered using a *Gaussian Mixture Model (GMM)*. Given a test sample $x$, the minimum *Mahalanobis distance* between $F(x)$ and the training feature clusters is calculated. Samples with low/high Mahalanobis distance are classified as clean/noisy respectively.

In this work, the following tasks were used to obtain feature extractors:

- An autoencoder was trained for reconstruction of clean ECGs. The encoder part was then used as a feature extractor.
- A classifier was trained on various diagnostic labels. The classification head was then removed and the rest of the network was used as a feature extractor.
- Contrastive learning (SimCLR) was used to directly learn feature representations.

We now briefly explain some of the technical terms introduced in this section.

**Gaussian Mixture Model.** A GMM is a probabilistic model which assumes that all data-points are generated from a finite number of possibly correlated Gaussian distributions. Popular algorithms that fit GMMs to data (such as the one in `scikit-learn`, which was used in this work) try to estimate the parameters of each cluster and the interaction between the clusters using an expectation-maximization algorithm [21].

**Mahalanobis distance.** The Mahalanobis distance [22] is a measure of distance between a point and a distribution. Given a point $x \in \mathbb{R}^d$ and a distribution $D$ over $\mathbb{R}^d$ with mean $\mu$ and covariance $\Sigma$, the Mahalanobis distance between $x$ and $D$ is given by:

$$M(x, D) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)}$$

Under the assumption that $D$ is Gaussian, $M$ can be shown to follow the chi-squared distribution with $d$ degrees of freedom. As a result, the CDF of the chi-squared distribution can be used to determine outliers at a certain confidence threshold of our choice. Since however $D$ being Gaussian is generally too strict of an assumption, the threshold can also be chosen once again empirically.

**SimCLR.** SimCLR [23] is a self-supervised framework for contrastive learning of low-dimensional representations of a dataset. The framework works as follows:

1. During each epoch, two (or more) random augmentations are produced for each data-point in a training batch.
2. The augmented samples are embedded through the feature extractor $F$ (often an additional projection head $G$ is added after $F$) into a low dimensional feature space.
3. For each batch, the InfoNCE loss [24] is calculated. In short, the InfoNCE loss tries to minimise the cosine similarity between augmentations of the same sample, and maximize that between augmentations of different samples.
4. After training, $F$ is used as a feature extractor.

## 3.3 Uncertainty based methods

It has been observed that DNNs can produce incorrect predictions with high confidence on OOD data [25]. To counter this, various methods have been developed in order to estimate the *uncertainty* associated with a DNN's predictions. Uncertainty can be a result of two causes:

- *Aleatoric uncertainty* arises from the inherent randomness in data, which can be background noise, measurement errors, or inconsistencies that can't be eliminated by gathering more data.
- *Epistemic uncertainty* is due to the model's lack of experience with data-points outside its training, and can be reduced by providing the model with more diverse data points that better represents the domain distribution.

In a medical setting, uncertainty estimation is crucial in order to avoid confident but erroneous predictions from an automated diagnostic system. The comprehensive study [6] investigates various estimation techniques for both types of uncertainty in the context of ECG classification. Works such as [26] also suggest that uncertainty estimation can be used for OOD detection.

In the context of ECG noise detection, the hypothesis explored in this work is that given a variety of training data, any sample which is determined to be OOD compared to them through uncertainty-based methods may be considered noisy, or unsuitable for diagnostic purposes (RQ3). The method is implemented as follows. A classifier

$$C : \mathbb{R}^{m \times N} \to \mathbb{R}^n$$

is trained on clean $m$-lead ECGs for prediction of various diagnostic labels. Upon testing, the (epistemic) uncertainty of the model's predictions is estimated. Samples with low/high uncertainty scores are classified as clean/noisy respectively, by comparison to a chosen threshold. The methods used to estimate uncertainty in this work are the following:

- *Ensemble methods,* which involve training the same architecture a number of times using different weight initializations, resulting in an ensemble of networks. The variance in the ensemble's predictions for any given input can be used as a measure of (epistemic) uncertainty [6].
- *Monte-Carlo dropout methods,* which apply on any network that is trained using dropout. Dropout is also kept upon testing, thus masking different weights of the network every time a prediction is made. The variance in the network's predictions for the same input over different dropout masks can be used as a measure of (epistemic) uncertainty [6].

## 3.4 Evaluation metrics

All of our models output a value which is to be compared with a threshold in order to make a classification decision. In order to assess the ability of our models to distinguish between the positive and the negative class over all classification thresholds, we use the ROC-AUC score, which stands for **R**eceiver **O**perating **C**haracteristic - **A**rea **U**nder **C**urve. The ROC curve is the plot of the True Positive rate (TPR) against the False Positive rate (FPR) at each distinct classification threshold, and its area is equal to the probability that the classifier ranks a randomly

chosen positive example higher than a randomly chosen negative example [27]. A perfect classifier receives a ROC-AUC score of 1, whereas a random classifier receives a score of 0.5. The ROC-AUC score is a standard metric often used in medical settings for comparing different diagnostic tests and determining their overall diagnostic performance [28].

Suppose now that for a specific model the threshold is chosen to produce a classifier which makes predictions $\hat{y}$ for each input $x$ with true label $y$. We are interested in the following metrics:

- *True Positive rate* (TPR), aka *sensitivity*, or *recall*:
$$P(\hat{y} = 1 \,|\, y = 1) = \frac{TP}{TP + FN}$$
- *True Negative rate* (TNR), aka *specificity*:
$$P(\hat{y} = 0 \,|\, y = 0) = \frac{TN}{TN + FP}$$
- *False Positive rate* (FPR), i.e. 1 - specificity:
$$P(\hat{y} = 1 \,|\, y = 0) = \frac{FP}{TN + FP}$$
- *False Negative rate* (FNR), i.e. 1 - sensitivity:
$$P(\hat{y} = 0 \,|\, y = 1) = \frac{FN}{TP + FN}$$

The above metrics are reported by means of confusion matrices, the rows of which are normalised by the number of samples with positive true label:

$$\begin{array}{c|c} \text{TNR} & \text{FPR} \\ \hline \text{FNR} & \text{TPR} \end{array}$$

We are also interested in other standard classification metrics:

- *Accuracy*:
$$P(\hat{y} = y) = \frac{TP + TN}{TP + TN + FP + FN}$$
- *Precision*:
$$P(y = 1 \,|\, \hat{y} = 1) = \frac{TP}{TP + FP}$$
- *F1 score*:
$$2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Any confidence intervals (CI) reported on the metrics are calculated by bootstrapping [29] on the test set.

# 4 Datasets

In this work we use 8-lead ECGs of length 5,000 for training and testing our algorithms. We mainly use two datasets, one of which is private, while the other is public.

## 4.1 UMCU dataset

Our private dataset is provided by Cordys Analytics in collaboration with the University Medical Center Utrecht (UMCU)[3]. The dataset consists of 386,413 8-lead ECGs sampled at 500 Hz over 10 seconds. This dataset is labelled by a panel of experts on more than 100 diagnostic criteria, including on existence of noise, and is the main dataset used for training, validation and testing.

[3] `https://www.umcutrecht.nl/en`

### 4.1.1 Noise label

The UMCU dataset contains two noise labels:

- *quality_to* (technisch onvolwaardig - technically defective), which indicates the presence of morphological noise in the ECG, such as significant baseline wander, high-frequency noise, missing leads, or other signal interference. This type of noise can be caused by many reasons including power line interference, muscle artefacts, movement of the patient, poor electrical connection, or other external sources.
- *quality_tf* (technische fout - technical error), which indicates that the order of leads in the ECG is incorrect. This type of noise is normally caused by misplaced electrodes during testing.

Both labels include different levels of noise (0-3 for quality_to, 0-2 for quality_tf). However, fewer than 0.01% of the total samples were found to have labels of level greater than 1. Therefore we opt for a binary label system instead, with 1 indicating any level of noise and 0 indicating lack of noise. The resulting label distributions are:

| **quality_to** | |
|---|---|
| 0 | 373,452 |
| 1 | 12,961 |

| **quality_tf** | |
|---|---|
| 0 | 385,607 |
| 1 | 806 |

A sample in which either label is positive is labelled as "noisy". The resulting *Noise* label has the following distribution:

| **Noise** | |
|---|---|
| 0 | 372,761 |
| 1 | 13,652 |

Figure 3 presents randomly selected examples of ECGs with different noise labels from the UMCU dataset.

### 4.1.2 Expert-annotated set

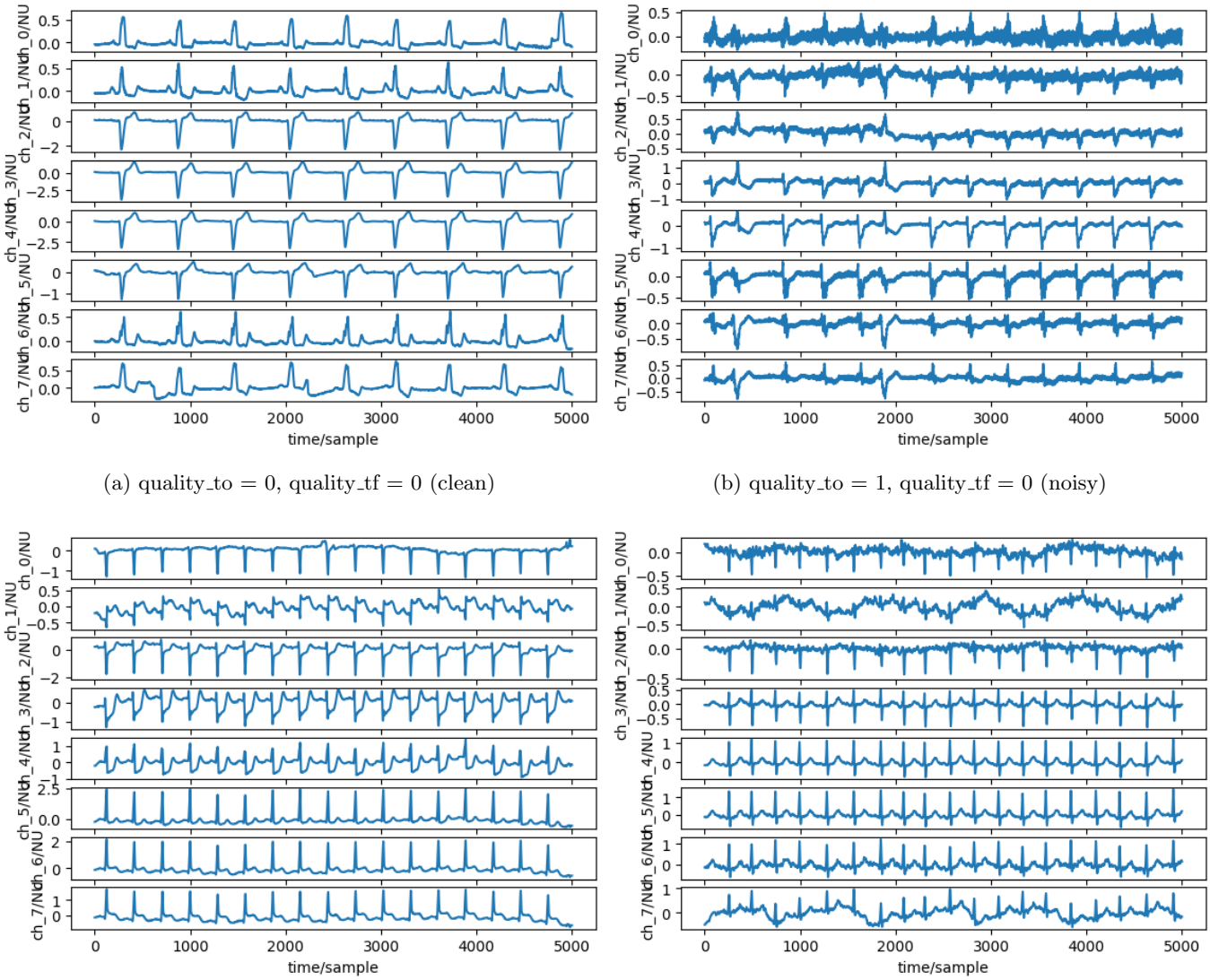
During the internship, a panel of experts in collaboration with Cordys Analytics further labelled a subset of 992 UMCU ECGs. The new labels are deemed to be of higher quality than those available before, and contain detailed annotations about how each ECG should be segmented into its distinct components, while also including noisy segments. Due to the fact that the main algorithms and pipelines had already been developed by the time that the new annotations became available to the author, the segmentation information was not used (see more in Section 8.1, *Limitations*). Instead, for the purposes of this work, every ECG which contained noise annotations was classified as noisy, resulting in the following distribution:

- 976 clean ECGs;
- 16 noisy ECGs.

This subset is used for testing purposes.

## 4.2 PhysioNet 2011 dataset

The PhysioNet 2011 Challenge [30, 31] aimed to create an efficient algorithm that can quickly provide feedback regarding the diagnostic interpretability of ECGs recorded in any setting, possibly by a non-expert. The public dataset associated with the challenge can be downloaded from [32]. It contains 1,000 noise-labelled 12-lead ECG recordings sampled at 500 Hz over 10 seconds, which were made by various experts and non-experts. The samples were reviewed by a group of independent annotators with various degrees of expertise, and the average label was calculated. The quality of 2 samples is labelled as "undetermined" due to lack of agreement between the annotators, while the other 998 ECGs have the following label distribution:




(a) quality_to = 0, quality_tf = 0 (clean)

(b) quality_to = 1, quality_tf = 0 (noisy)

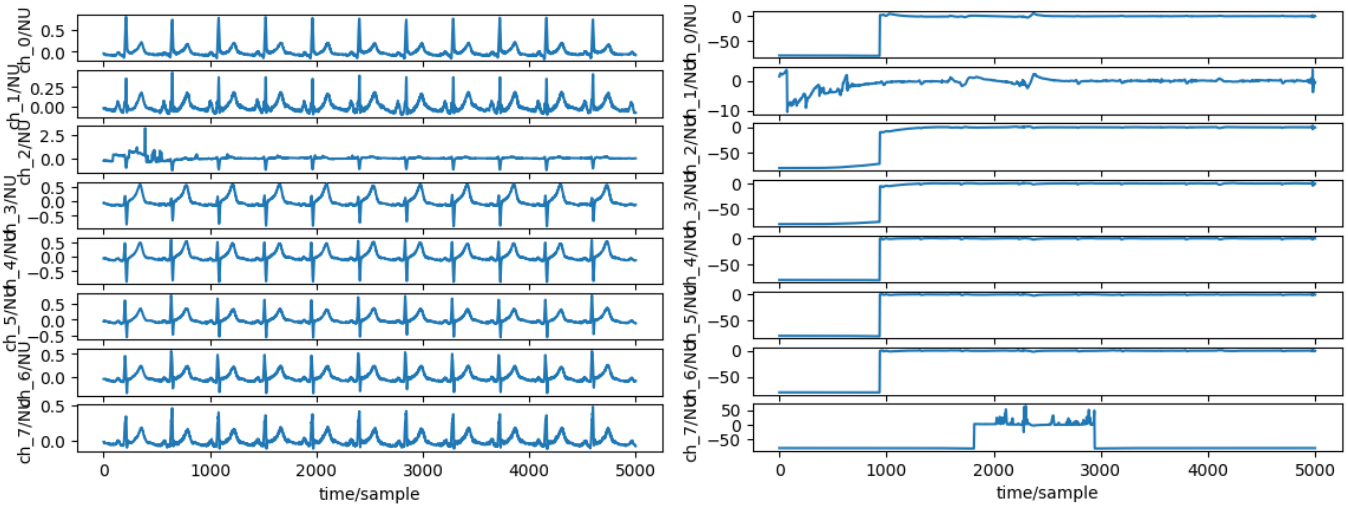
(c) quality_to = 0, quality_tf = 1 (noisy)

(d) quality_to = 1, quality_tf = 1 (noisy)

Figure 3: Examples of ECGs from the UMCU dataset with different noise labels.

- 773 "acceptable" (clean);
- 225 "unacceptable" (noisy).

The 998 ECGs with unambiguous labels are used in this work for testing purposes. Figure 4 shows randomly selected examples of clean and noisy ECGs from the PhysioNet 2011 dataset.




(a) PhysioNet 2011 dataset, clean example

(b) PhysioNet 2011 dataset, noisy example

Figure 4: Examples of ECGs from the PhysioNet 2011 dataset with different noise labels.

# 5 Data preprocessing

We now explain the preprocessing that was applied on the data. First, we discuss the issue of data normalisation.

## 5.1 Data normalisation

When training a DNN for example in computer vision, it is typical to normalise the data so that all the values are in the range $[0, 1]$ or $[-1, 1]$. This is not advisable in the case of ECGs for the following reasons:

- The scale of an ECG has a diagnostic meaning, i.e. different conditions can produce heart rhythms of different amplitudes. Therefore, any model used for diagnostic purposes must take the scale of an ECG into account.
- Noise in the signal is also scale-dependent, and normalising the values of the ECG would also normalise the noise. This is undesirable for types of noise that may present with amplitudes outside the normal ECG range, such as large baseline drift. On the other hand, if the ECG contains Gaussian noise with small amplitude, then normalisation would make this noise harder to detect. The above points are particularly relevant for the reconstruction experiments, where normalising the noise would lead to better reconstructions of noise itself, leading to an increase of FNR (see Section 7.2.1, *Experiments*).

Instead of normalisation, the gain value of 0.00488 was applied on each signal in order to convert it from the scale in which it was stored in the dataset, which is of order $\mu$Volt ($\mu$V), to the order of mVolt (mV). The value itself is obtained from the dataset's metadata, and indicates an average gain value that should be appropriate for the vast majority of the ECGs in it (since individual gain values are not stored). Thus, most ECG signals in the dataset end up being within the range of $[-5, 5]$ (in mV), which is normal for ECGs [33]. More on the statistical distribution of the clean signals of the UMCU dataset in particular is analysed in Section 5.2.2, *Clean split A*.

## 5.2 UMCU data

The UMCU data is split in three ways for the different experiments.

### 5.2.1 Binary classification split

For the binary classification experiments, the data is split into train/validation/test sets, with relative sizes 0.8/0.1/0.1. The split is performed at the patient ID level, in order to avoid ECGs from the same patient to end up in different sets.

### 5.2.2 Clean split A

For the rest of the experiments we aim to use only clean ECGs for training, hence the noisy ECGs are all removed from the training set. Additionally, we want to remove ECGs with morphological properties that possibly indicate noise, but can be easily detected by a rule-based system. Therefore, ECGs are additionally removed if:

- The *maximum absolute value* in the ECG (after applying gain) is greater than 5 (3,264 ECGs are flagged).
- The *maximum standard deviation* over all leads of the ECG (after applying gain) is greater than 1 (4,675 ECGs are flagged).
- The *minimum standard deviation* over all leads of the ECG (after applying gain) is less than 0.02 (560 ECGs are flagged).

Note that the above rules may overlap. In total, along with the 13,652 noisy ECGs, another 5,564 ECGs are removed to make a total of 19,216 OOD ECGs, leaving a set of 367,197 in-distribution ECGs. Once again, the data is split into train/validation/test sets at the patient ID level, with relative sizes 0.8/0.1/0.1.

Figure 5 shows examples of ECGs that are removed by our rule-based filters, but are not labelled as noisy.

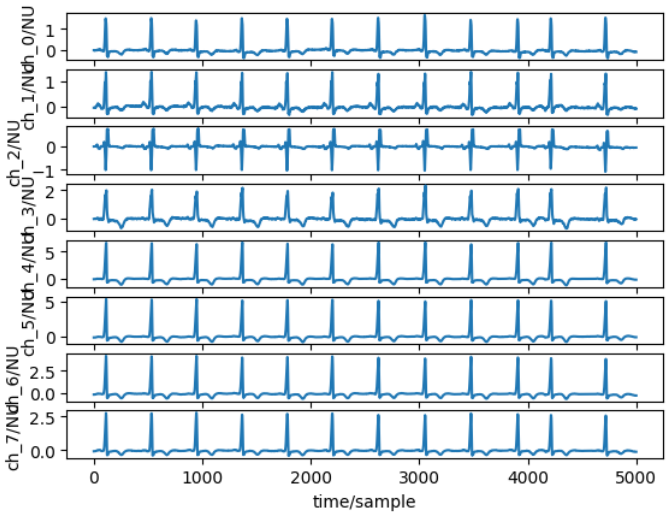
**Noise detection test set.** Out of the 13,652 noisy ECGs, we discard 1,040 ECGs that are easily flagged by the above rules, and we are left with 12,612 noisy ECGs. Along with the in-distribution test set, which consists of 36,593 clean ECGs, we form a set of 49,205 ECGs on which the reconstruction and the feature space methods are tested for ECG noise detection. Note that the ECGs that were removed based on morphological properties are not included in the noise detection test set, since they can be detected anytime by the same simple rules.

### 5.2.3 Clean split B

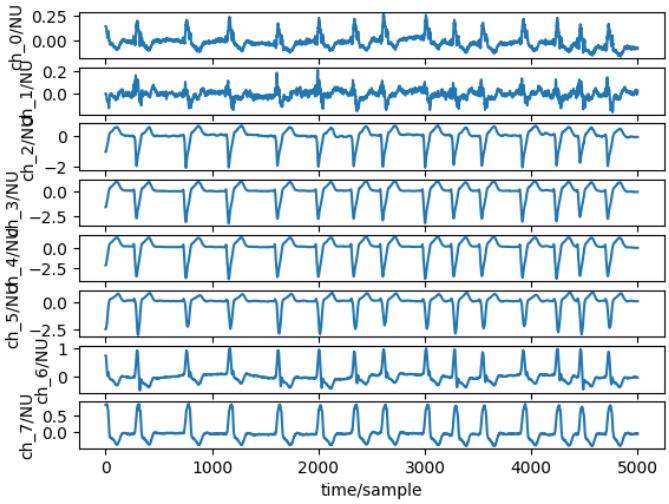
For the uncertainty based experiments, along with noisy ECGs we also aim to detect "difficult" ECGs, which are either rare in real-world scenarios, or challenging even for human experts to interpret. We would also like to evaluate our algorithms on a test set that does not contain such difficult examples in the clean label. For the above purposes, we use the diagnostic labels available in the UMCU dataset to determine the following categories of interest:

- *Rhythm type:* 12 mutually exclusive labels (multi-class);
- *AV conduction:* 4 mutually exclusive labels (multi-class);
- *Ventricular conduction:* 4 mutually exclusive labels (multi-class);
- *Ischemia:* 3 mutually exclusive labels (multi-class);
- *Other:* 6 independent labels (multi-label).

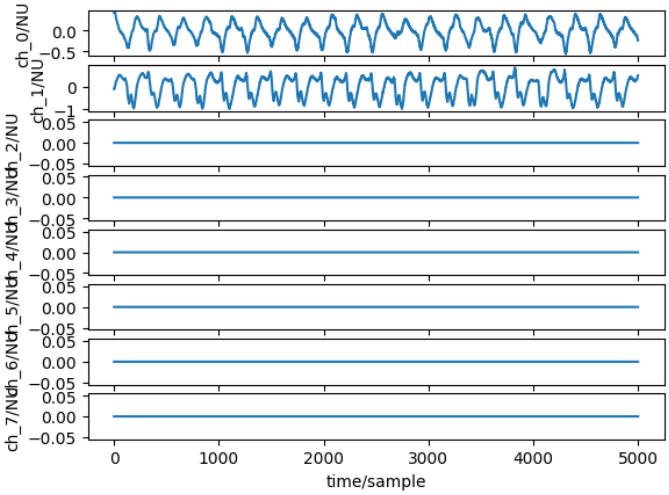
While it is not impossible in a real life clinical scenario for some labels to co-exist in the above multi-class categories, it is quite rare for this to happen, and might be indicative of severe pathology, or of an ECG that is hard to interpret. Our hypothesis is that such ECGs should result in increased uncertainty in a diagnostic model's predictions. To test this hypothesis, out of the 367,197 ECGs of Clean split A, we keep only those that are labelled with a single class in the multi-class categories. In total, 18,543 more ECGs are removed in this manner, leaving a set of 348,654 in-distribution




(a) Max abs. value $> 5$, noise label $= 0$




(b) Max SD $> 1$, noise label $= 0$




(c) Min SD $< 0.02$, noise label $= 0$

Figure 5: Examples of ECGs that are labelled as clean but are flagged as OOD by our filters.

ECGs. Train/validation/test splits are constructed by filtering the corresponding splits of Clean split A. The corresponding noise detection test set is constructed by combining the 12,612 noisy ECGs that are left behind by Clean split A, along with the 35,097 ECGs that are left in the test set of Clean split B after filtering the test set of Clean split A. Note that we do not further filter the noisy part of the noide detection test set through the diagnostic rules, since detecting noise by first figuring out the diagnostic labels defeats the purpose.

#### 5.2.4 Expert-annotated set

The expert annotated-set is ensured to be entirely contained within the test sets of all the other UMC splits. After filtering this set with the morphological rules of Section 5.2.2, *Clean split A*, we are left with 976 ECGs that pass our filters, out of which:

- 962 are clean;
- 14 are noisy.

### 5.3 PhysioNet 2011 data

The PhysioNet 2011 is used exclusively for testing. After removing outliers based on our morphological rules, only 299 ECGs remain, out of which:

- 276 are clean;
- 23 are noisy.

While a lot of ECGs that are labelled as clean are excluded, this is justified since it is important to keep a consistent distribution across all test sets.

### 5.4 Summary of data splits

This section summarises the information about the different datasets, their usage, the manner in which they were split during experimentation, and their label distribution.

**Usage of each split.** Table 1 shows how each split is used in the different methods tested in this work.

| | **Binary class.** | **Reconstruction** | **Feature space** | **Uncertainty** |
|---|---|---|---|---|
| **Binary class.** | **train, val, test** | *n/a* | *n/a* | *n/a* |
| **Clean split A** | *n/a* | **train, val, test** | **train, val, test** | *n/a* |
| **Clean split B** | *n/a* | test | test | **train, val, test** |
| **Expert-ann.** | test | test | test | test |
| **PhysioNet** | test | test | test | test |

Table 1: Usage of each split.

**Binary classification split.** Table 2 shows the label distribution over all splits used for training and evaluation of the binary classification algorithm.

| | **Training set** | **Validation set** | **Test set** | **Sum** |
|---|---|---|---|---|
| **Clean** | 298,865 | 36,827 | 37,069 | 372,761 |
| **Noisy** | 10,952 | 1,360 | 1,340 | 13,652 |
| **Sum** | 309,817 | 38,187 | 38,409 | 386,413 |

Table 2: Summary of binary classification data split.

| Clean split A | Pass rules | Fail rules | Sum |
|---|---|---|---|
| **Clean** | 367,197 | 5,564 | 372,761 |
| **Noisy** | 12,612 | 1,040 | 13,652 |
| **Sum** | 379,807 | 6,604 | 386,413 |

Table 3: Construction of Clean split A.

**Clean splits.** Table 3 summarises how the data is split in order to form Clean split A according to the chosen morphological rules.

Now each sub-split of Clean split B is simply a subset of the corresponding sub-split of Clean split A that was filtered using the diagnostic rules. The noisy ECGs are not subject to the diagnostic rules. Table 4 shows how Clean split B is formed according to the morphological as well as the diagnostic rules.

| Clean split B | Pass rules | Fail rules | Sum |
|---|---|---|---|
| **Clean** | 348,654 | 24,107 | 372,761 |
| **Noisy** | 12,612 | 1,040 | 13,652 |
| **Sum** | 361,266 | 25,147 | 386,413 |

Table 4: Construction of Clean split B.

Table 5 summarises how Clean split A and Clean split B are used for training and testing. Here, *training* refers to the actual training set, *validation* refers to the set used for validation during training in order to select the best model, and *testing* refers to the set kept aside to be used along with the noisy data for evaluation of the algorithms on noise detection.

| | Clean split A | Clean split B |
|---|---|---|
| **Training** | 294,046 | 278,719 |
| **Validation** | 36,558 | 34,838 |
| **Testing** | 36,593 | 35,097 |
| **Sum** | 367,197 | 348,654 |

Table 5: Summary of data that is used for training/testing purposes on each split.

**Noise detection test sets.** Table 6 shows the label distribution of all the sets that are used for testing our algorithms on the task of noise detection.

# 6 Architectures

In total, 4 different model architectures are used during the experimentation process:

1. a *binary classifier* (BinCl);
2. an *autoencoder* (AE);
3. a *diagnostic classifier* (DiagCl);
4. a *SimCLR* architecture.

In this section we go into details about each architecture.

## 6.1 Binary classifier

The main BinCl architecture consists of a double residual 1-D convolutional encoder, based on [34], followed by a classification head. Dropout layers are used both in the encoder as well as before the classifier, while non-linearity is added to the classification head by means of the RELU activation

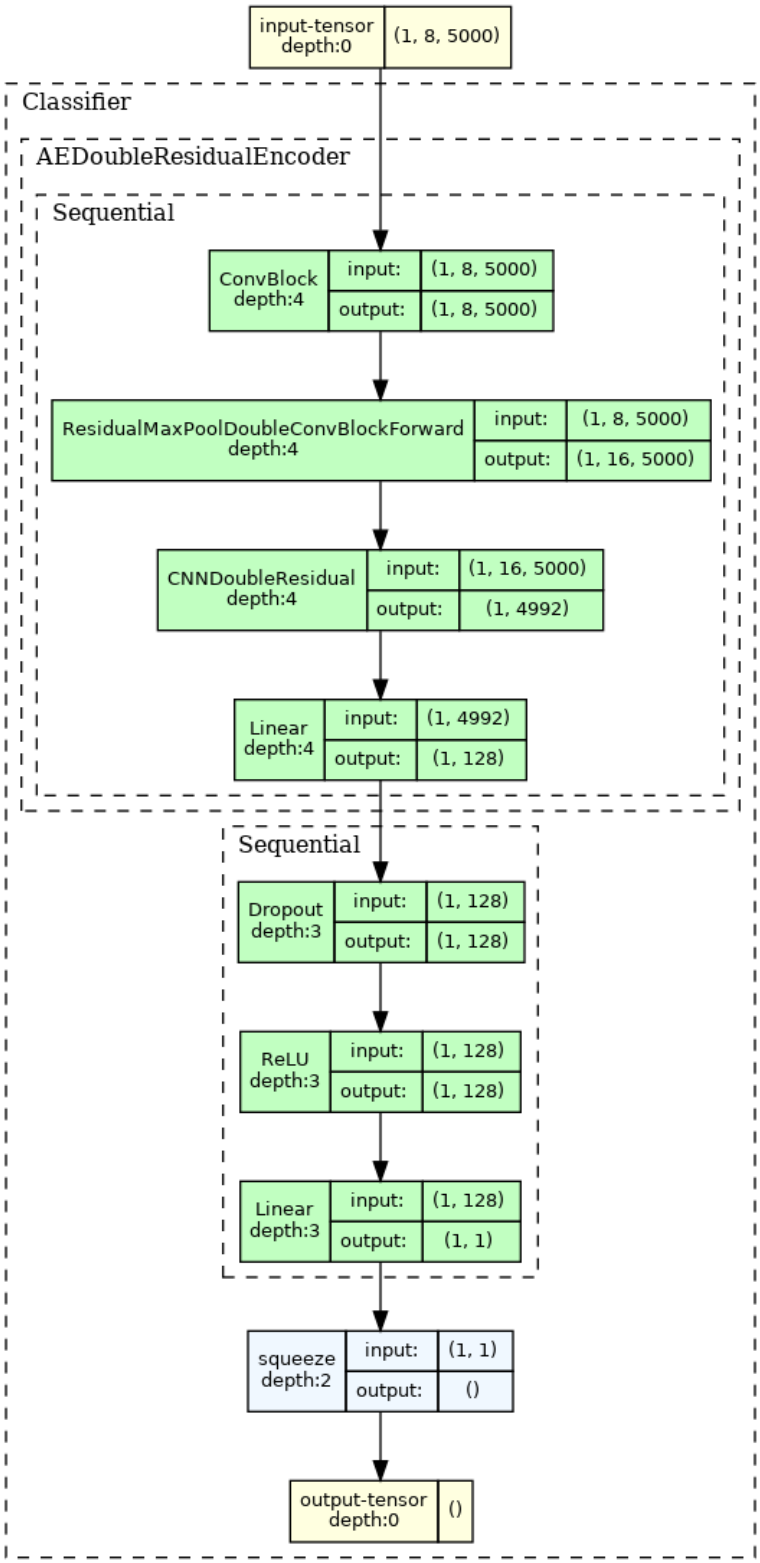



Figure 6: Binary classifier architecture.

| | Binary class. | Clean A | Clean B | Expert-ann. | PhysioNet 2011 |
|---|---|---|---|---|---|
| **Clean** | 37,069 | 36,593 | 35,097 | 962 | 276 |
| **Noisy** | 1,340 | 12,612 | 12,612 | 14 | 23 |
| **Sum** | 38,409 | 49,205 | 47,709 | 976 | 299 |

Table 6: Summary of sets that are used for testing the noise detection algorithms.

function. An example of the architecture can be found in Figure 6. More detailed views of the components of the encoder can be found in Section 10, *Appendix*, Figures 20, 22 and 23.

### 6.2 Autoencoder

The main AE architecture used for our experiments consists of a double residual encoder and a double residual decoder, based on [34]. Figure 7 shows an example of the architecture. The decoder architecture mirrors the one of the encoder (see Section 7.1.1, *Experiments*), containing the same components but in a reverse order. The architectures of the mirrored components can be seen in Section 10, *Appendix*, Figures 23 and 24.

### 6.3 Diagnostic classifier

The DiagCl architecture consists of a double residual encoder [34], along with 5 independent classification heads, one for each diagnostic class of interest as defined in Section 5.2.3, *Clean split B*. Each classification head consists of a dropout layer followed by RELU and a linear layer. Figure 8 shows an example of the architecture of the DiagCl model.

### 6.4 SimCLR

Our implementation of the SimCLR architecture consists of a double residual encoder [34] followed by a projection head. The use of projection heads in contrastive learning is studied extensively in [35], where it is shown that using a non-linear projection head during training, and then discarding it in order to obtain embeddings from the layer just before it, produces better representations compared to training by applying the InfoNCE loss on the feature extractor directly.

### 6.5 Summary of models and methods

Each method uses specific architectures as its main tools. However, some architectures may be used as secondary tools during the experiments of each method in order to assist training, e.g. for better weight initialization. Table 7 shows a summary of which models were used for which method. Details about how each model was used can be found in Section 7, *Experiments and Results*.

| | Binary class. | Reconstruction | Feature space | Uncertainty |
|---|---|---|---|---|
| **Binary class.** | **main** | *n/a* | *n/a* | *n/a* |
| **Autoencoder** | *n/a* | **main** | **main** | *n/a* |
| **Diagnostic class.** | *n/a* | secondary | **main** | **main** |
| **SimCLR** | *n/a* | secondary | **main** | secondary |

Table 7: Summary of models and methods in which they were used

## 7 Experiments and Results

We now go into details about the experiments conducted with each of the methods described in Section 3, *Methodology*. Plots of the training metrics, as well as confusion matrices and classification reports for all experiments mentioned in this section can be found in Section 10, *Appendix*.

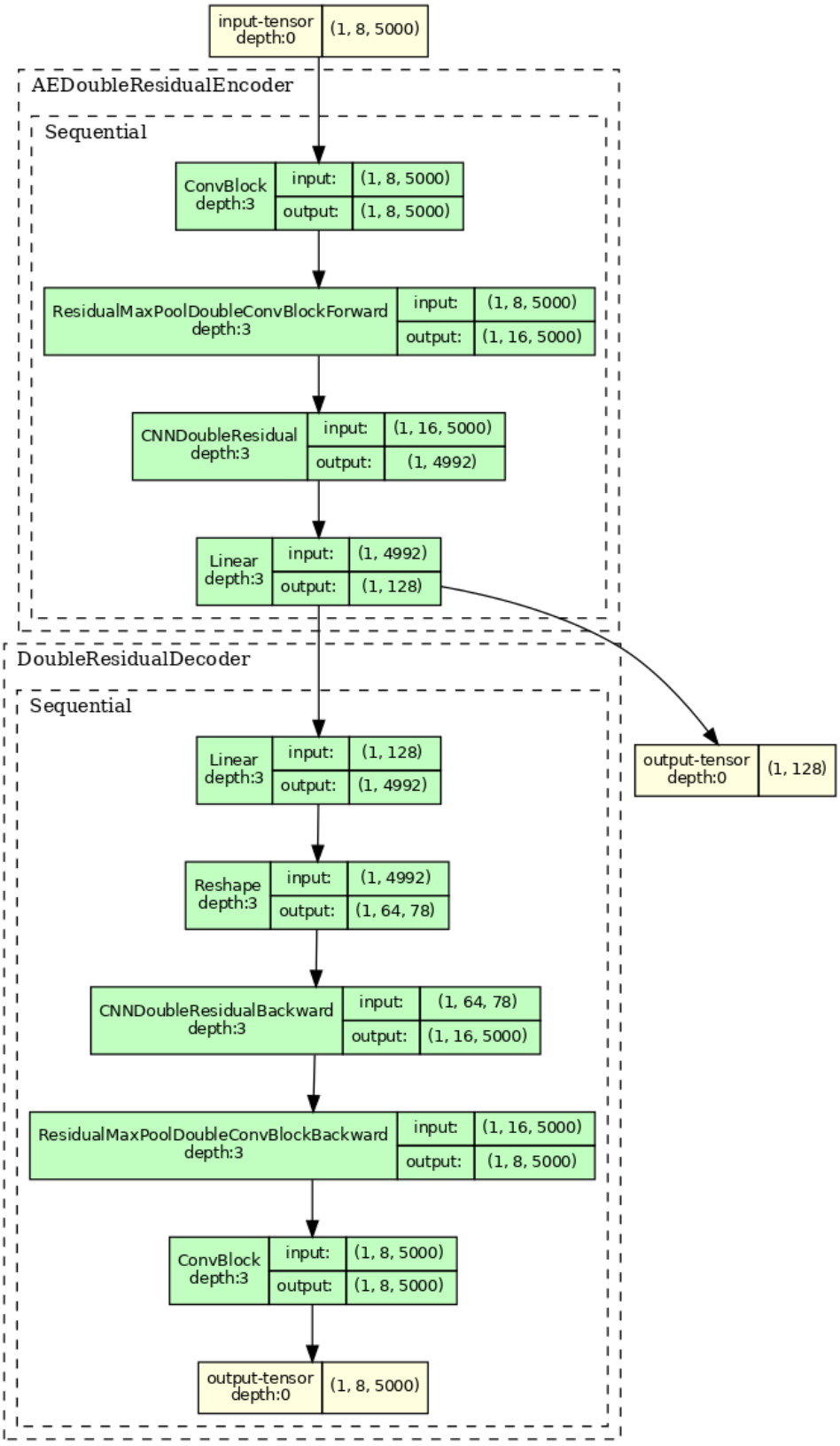



Figure 7: AE architecture.

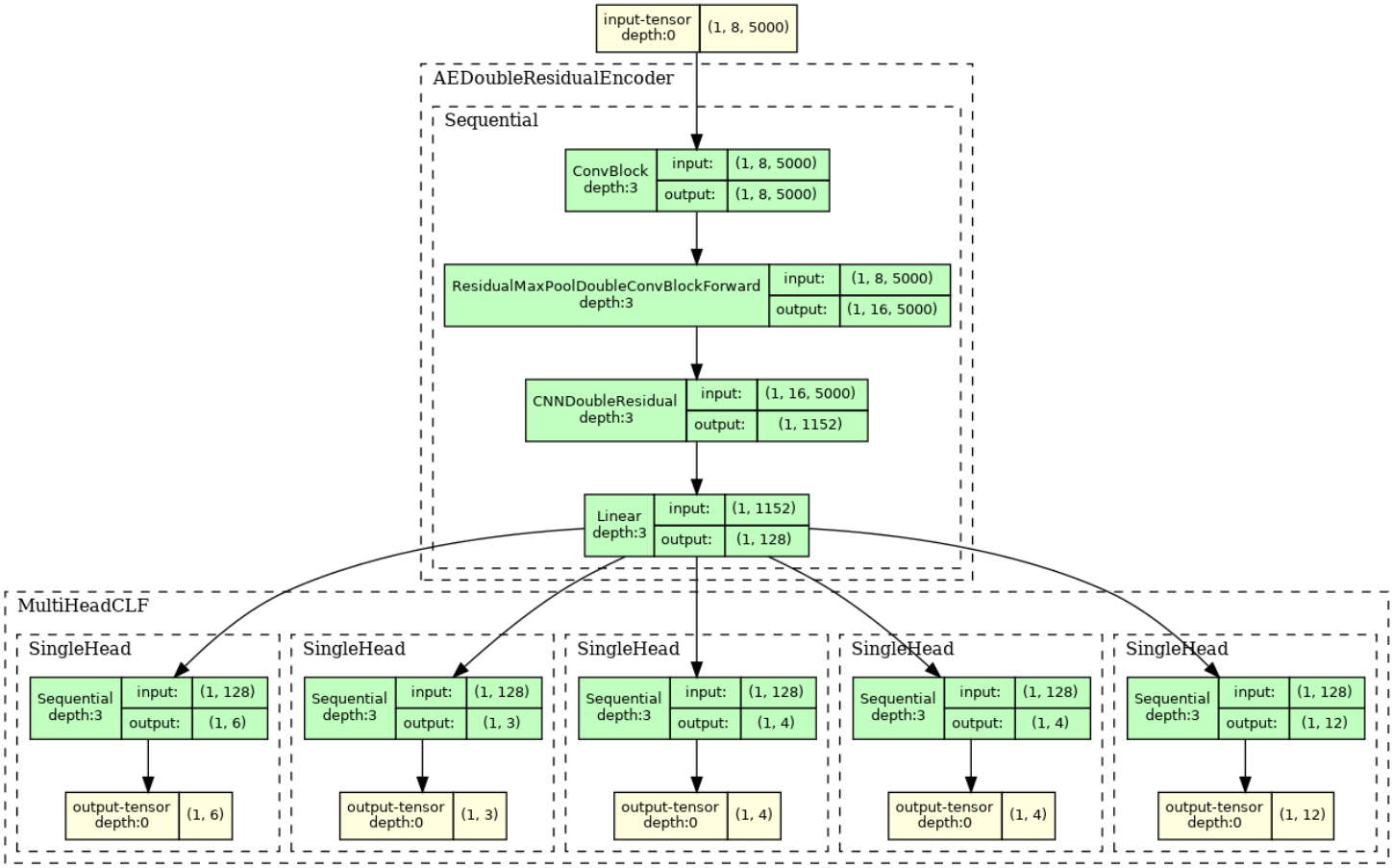



Figure 8: DiagCl architecture.

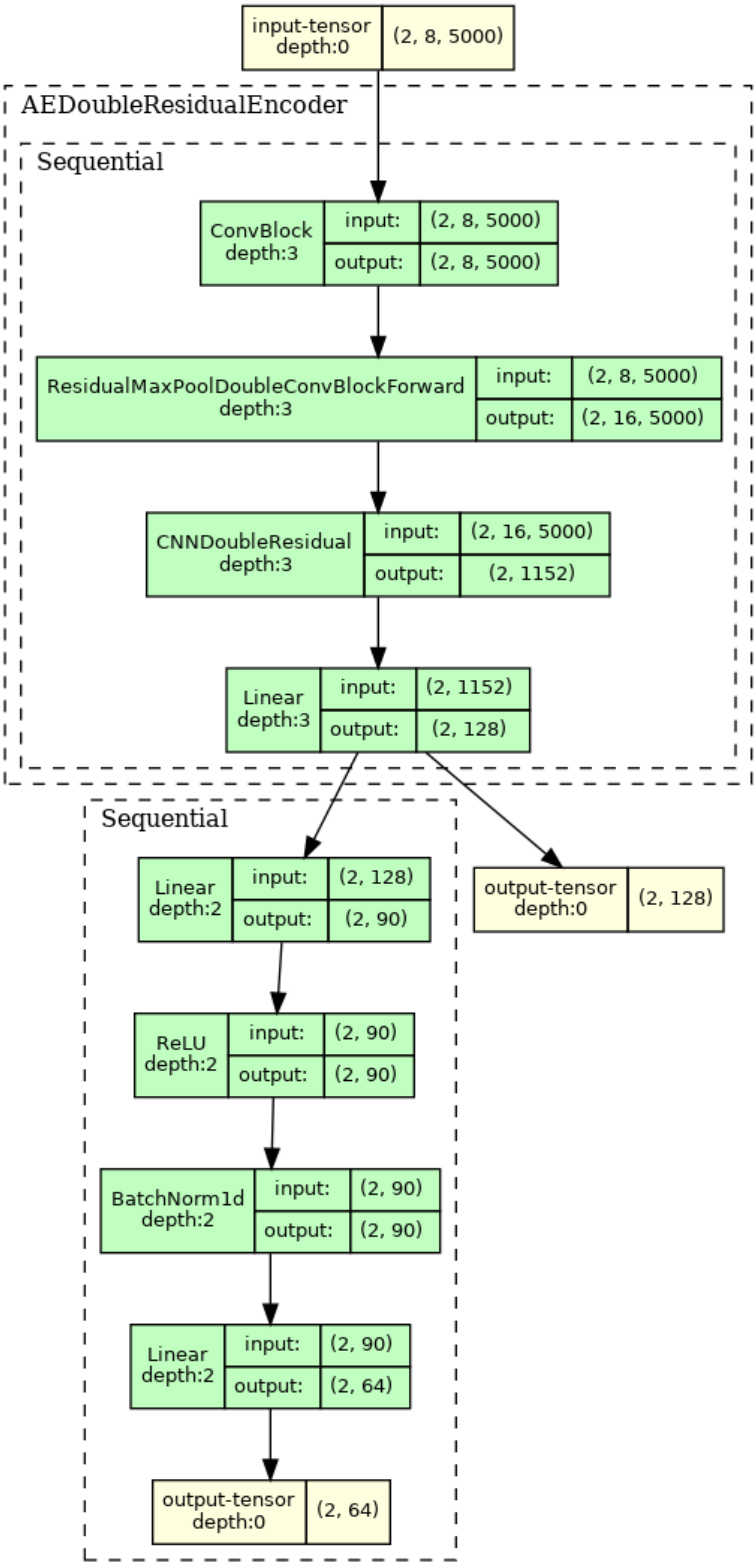



Figure 9: SimCLR architecture.

## 7.1 Binary classification

We briefly discuss the experiments done for the binary classification task, as explained in Section 3.1, *Binary classification*.

### 7.1.1 Experiments

The main architecture used for these experiments is the BinCl (see Section 6.1, *Binary classifier*).

**Loss.** As only around 3.5% of the ECGs belong to the positive class, the dataset suffers from severe class imbalance. To counter this, we optimise with respect to the Focal Loss [36], defined as:

$$FL_{\alpha,\gamma}(p) = \begin{cases} -\alpha(1-p)^\gamma \log(p), & y = 1 \\ -(1-\alpha)p^\gamma \log(1-p), & y = 0 \end{cases}$$

for prediction $p \in (0, 1)$ and target $y \in \{0, 1\}$. We use parameters $\alpha = 0.8$, $\gamma = 2$.

**Training.** We report on the training of the most successful model. The network was trained on the binary classification split over 30 epochs using the AdamW optimiser [37] starting from a learning rate of $10^{-4}$. We used the ReduceLROnPlateau learning rate scheduler, which divides the learning rate by a factor of 10 whenever the validation loss stops decreasing over 3 epochs. Training terminated at 24 epochs because the validation loss stopped improving over 10 epochs. The best model was chosen as the one with the smallest validation loss, at epoch 14. Figure 25 shows the training and validation losses as well as the training and validation ROC-AUC scores per epoch.

### 7.1.2 Results

Our most successful model contains 749k parameters.

**Binary classification test data.** The model achieved a ROC-AUC of 0.907 (95% CI: [0.898, 0.916]) on the binary classification data. The ROC curve is shown in Figure 29.

A classification threshold of 0.3 is chosen empirically based on the classification performance on the validation data. Using this threshold, the model achieves an accuracy of 0.85 on the test data. Figure 30 shows the scaled confusion matrix, whereas Table 10 reports on the other classification metrics.

**Expert-annotated set.** The model achieves even better results on the expert annotated set, with a ROC-AUC of 0.978 (95% CI: [0.954, 0.995]). The ROC curve is shown in Figure 31. Using the same threshold of 0.3, the model achieves accuracy of 0.85. The scaled confusion matrix is found in Figure 32 and the rest of the metrics in Table 11.

**PhysioNet 2011 data.** Performance on the PhysioNet 2011 data is similarly strong, with a ROC-AUC of 0.909 (95% CI: [0.860, 0.947]). The ROC curve is shown in Figure 33. Using the same classification threshold of 0.3 the model achieves accuracy of 0.76. Figure 34 shows the scaled confusion matrix and Table 12 the rest of the metrics.

## 7.2 Reconstruction

We now detail the experiments conducted with the AE reconstruction loss method, which is described in Section 3.2.1, *Autoencoder reconstruction loss as a measure of noise*.

### 7.2.1 Experiments

The main architecture used for these experiments is the AE (see Section 6.2, *Autoencoder*).

**Loss.** The AE was optimised with respect to the MSE loss:

$$MSE(x, y) = \frac{1}{m \cdot N} \sum_{c=1}^{m} \sum_{i=1}^{N} (x_{c,i} - y_{c,i})^2,$$

where $x$ is the input and $y$ is the reconstruction of an ECG of size $m \times N$. Here $m = 8$, $N = 5,000$.

**Issues.** Initial experiments had little success, achieving a ROC-AUC score of 0.630 (95% CI: [0.617, 0.643]). The following issues were observed:

- Rare abnormal rhythms are hard for the AE to reconstruct, resulting in greater reconstruction loss, which increases the FPR. This issue is related to RQ2. Figure 10 presents an example of a clean but diagnostically abnormal ECG which is flagged as noisy by the AE.
- High frequency Gaussian noise is often small in amplitude, resulting in small reconstruction loss, which increases the FNR. Figure 11 presents an ECG which contains high-frequency noise, but its overall form is approximated well by the AE, resulting in small reconstruction loss.
- The scale of the ECG can affect the scale of the loss. In particular:
  - Noisy ECGs with small amplitudes might also produce reconstructions with small amplitudes, and therefore small reconstruction loss, increasing the FNR.
  - Autoencoders tend to produce small neural activations when presented with unfamiliar inputs, whether those are genuinely OOD, or rare conditions that are under-represented in the training data [38]. The former case increases the FNR as explained above, while the latter case increases the FPR.

**Dealing with FP.** The following techniques were used to help counter the issue of abnormal rhythms being flagged as FP (RQ2):

- *Random augmentations* were introduced to the training set with the aim of better familiarizing the autoencoder with different ECG presentations. Data augmentation techniques are widely used in many domains in order to produce more robust models. However, our options for augmentations are more limited in the noise detection task. Indeed, it is important not to introduce noise to the training data, otherwise we are running into the risk of the model overfitting and producing good reconstructions for noise as well. The study [39] investigates a number of augmentations (albeit for the different task of contrastive representation learning of ECGs). In the end the following augmentations were applied:
  - *Random shift*: The signals were shifted in the time dimension by a random factor. Mirroring was used to fill in the blank space left behind by the shifting.
  - *Baseline drift:* Small amounts of baseline drift were randomly applied on the signal. Large baseline drift may be classified as noise, but small amounts still leave behind an interpretable signal.
  - *Permutation:* Following [39], the ECG are split into $n$ parts of equal size in the time dimension. The parts were then permuted randomly in order to create a new ECG. This technique can actually introduce noise, so it is important to apply it sparingly and to ensure that $n$ is approximately equal to number of heartbeats expected to be found in the ECG based on its length (e.g. $n = 10$ in our case, see [39]).
- *Weighted sampling* was used in order to increase the model's familiarity with rare conditions. The weights applied for each sample were determined as follows:
  - For each sample, its least frequent diagnostic label of interest is determined.
  - The sample receives a weight inversely proportional to the frequency of its least frequent label.
  - Additionally, when testing on the validation set, the classes that most often appear as FP receive an increased weight.

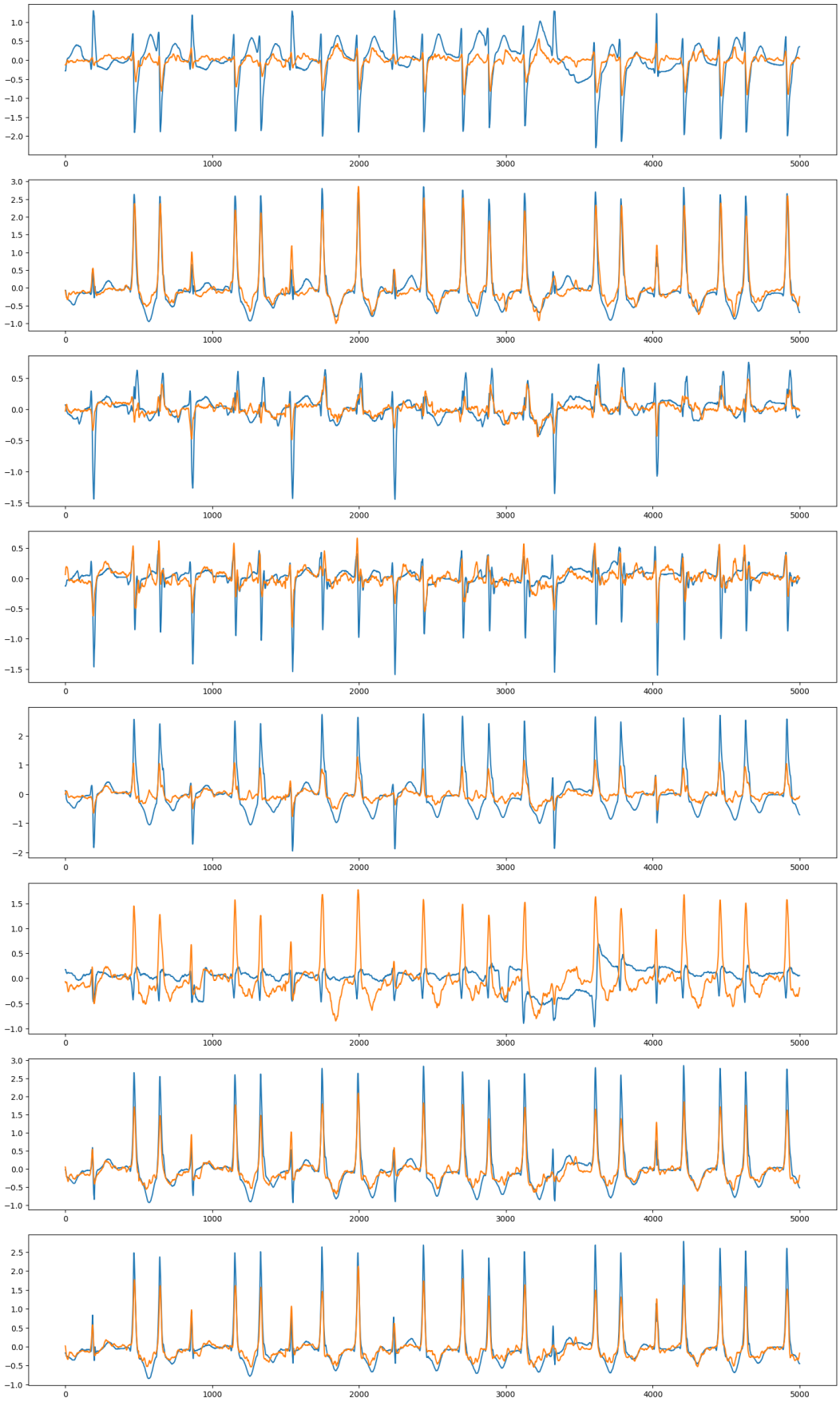


Figure 10: Clean example with high reconstruction loss (FP). Blue is the original signal, while orange is the AE reconstruction.

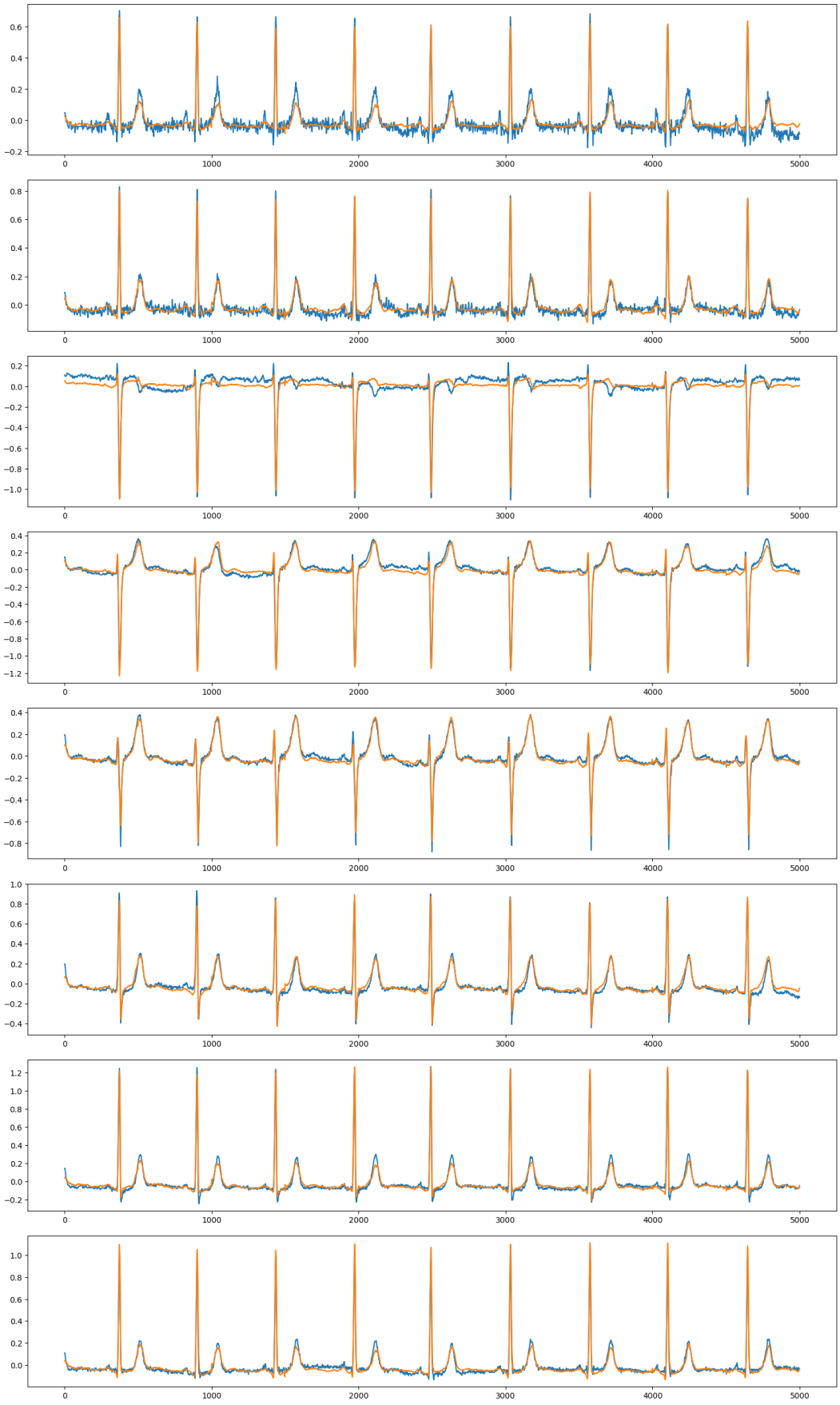


Figure 11: Noisy example with low reconstruction loss (FN). Blue is the original signal, while orange is the AE reconstruction.

Other ideas were also attempted in order to deal with the issue of FP, including:

- A DiagCl model (see Section 6.3, *Diagnostic classifier*) was trained on the diagnostic labels. The weights of the classifier were then frozen. During training of the AE, the reconstruction produce by the AE was input in the trained classifier. A weighted sum of the reconstruction loss and the classifier loss was used for optimisation. The idea here is to force the network to make representations that better capture the unique features of each diagnostic label.
- An encoder was pretrained using SimCLR (see Section 6.4, *SimCLR*), and then finetuned as a part of the AE system. The idea here is that better weight initialization as well as better encodings in the AE's latent space may lead to better reconstructions overall.

These ideas were abandoned however, because they did not seem to result in significant improvements.

**Dealing with FN.** Dealing with false negatives is a harder task, since more training leads to better reconstructions, which also applies to noisy ECGs, which then increases the FN rate. One idea is to try and deal with high-frequency noise separately, by examining the frequency components of each signal using a Fourier Transform. In particular, we apply a 4th-order low-pass Butterworth filter [40] with a cutoff frequency at 40 Hz, which should not affect most normal ECGs that are mainly concentrated in the 0.05-35 Hz range [33]. The Butterworth filter is a standard signal processing tool. In short, it works by multiplying each frequency found in the signal by a certain factor in order to leave the desired frequencies mostly unchanged, while minimising the contribution of unwanted frequencies. Figure 12 shows the frequency response of the Butterworth filter used in this work; the cutoff frequency (i.e. the frequency with voltage gain of $1/\sqrt{2} \approx 0.7071$, which is 40 Hz here) is highlighted in red. We attempt to detect ECGs with high-frequency noise

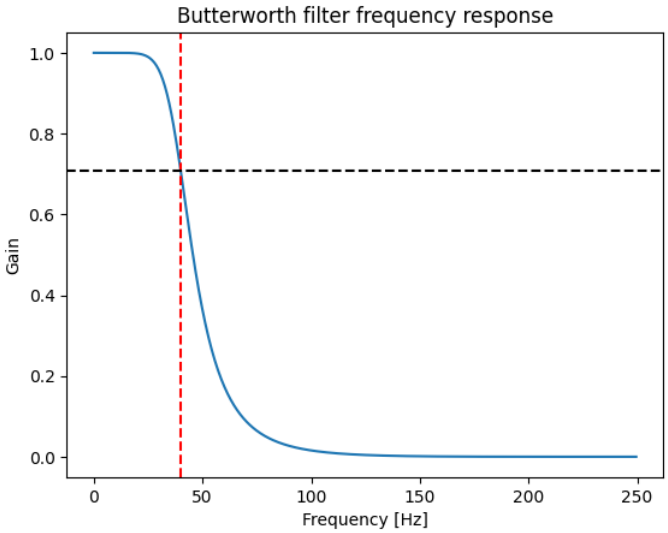



Figure 12: Frequency response for 4th-order low-pass Butterworth filter with a cutoff frequency at 40 Hz.

as follows. For each ECG $x$ let $x'$ be the resulting signal after $x$ passes through the filter. We calculate the $L_1$-norm of $x - x'$, i.e.:

$$L_1(x, x') = \sum_{c=1}^{8} \sum_{i=1}^{5{,}000} |x_{c,i} - x'_{c,i}|.$$

Signals with high $L_1$-norm are considered noisy. Classification happens based on a fixed threshold of 500. This rule filters 17% of the noisy data while still letting through 98% of the clean data.

**Dealing with scaling.** As explained in Section 5.1, *Data normalisation*, the data is not normalised during training, since the scale of the ECGs has diagnostic meaning, and the noise is scale-dependent. However, scaling can be used upon inference in order to help mitigate the effect

of the ECG's scale on the reconstruction loss. Following [38], instead of the MSE loss we use the following scaled Euclidean distance upon inference:

$$L_{2-scaled}(x, y) = \left\| \frac{x}{\|x\|} - \frac{y}{\|x\|} \right\|,$$

where $x$ is the input and $y$ is the reconstruction. Note that since any input $x$ that contain all 0s can easily be detected by our rule based system, in practice it the above distance should always be defined.

**Training.** We report on the training of the most successful model with respect to the noise detection task. The AE was trained over 30 epochs on Clean split A using the AdamW optimiser [37] and the OneCycle learning rate scheduler [41], which starts the learning rate from a small value, increases it over 3 epochs to its max value of $10^{-3}$, and then gradually decreases to almost 0 over the rest of the epochs via a cosine annealing strategy. The model with the smallest validation loss, achieved at epoch 28, is kept for further testing. Figure 26 shows the training and validation losses per epoch.

### 7.2.2 Results

Our most successful model is rather large, at 51.9 million parameters. Multiple attempts were made to reproduce its results with smaller models using different training hyper-parameters; however, these attempts were largely unsuccessful.

**Clean split A.** Training using random augmentations and weighted sampling improved the performance of the model to a ROC-AUC of 0.684 (95% CI: [0.679, 0.690]). Figure 13 shows the label distribution in the FP compared to the distribution of the clean samples for the same threshold. We observe that many of the abnormal classes are excessively flagged as FP, which indicates a bias in the model, since FP are also clean by definition. The distribution remains biased after applying random augmentations and weighted sampling, however the differences are less pronounced in some abnormal labels (see e.g. labels *class_abnorm*, *rhythm_st*, *conduction_rbtb*, etc.).

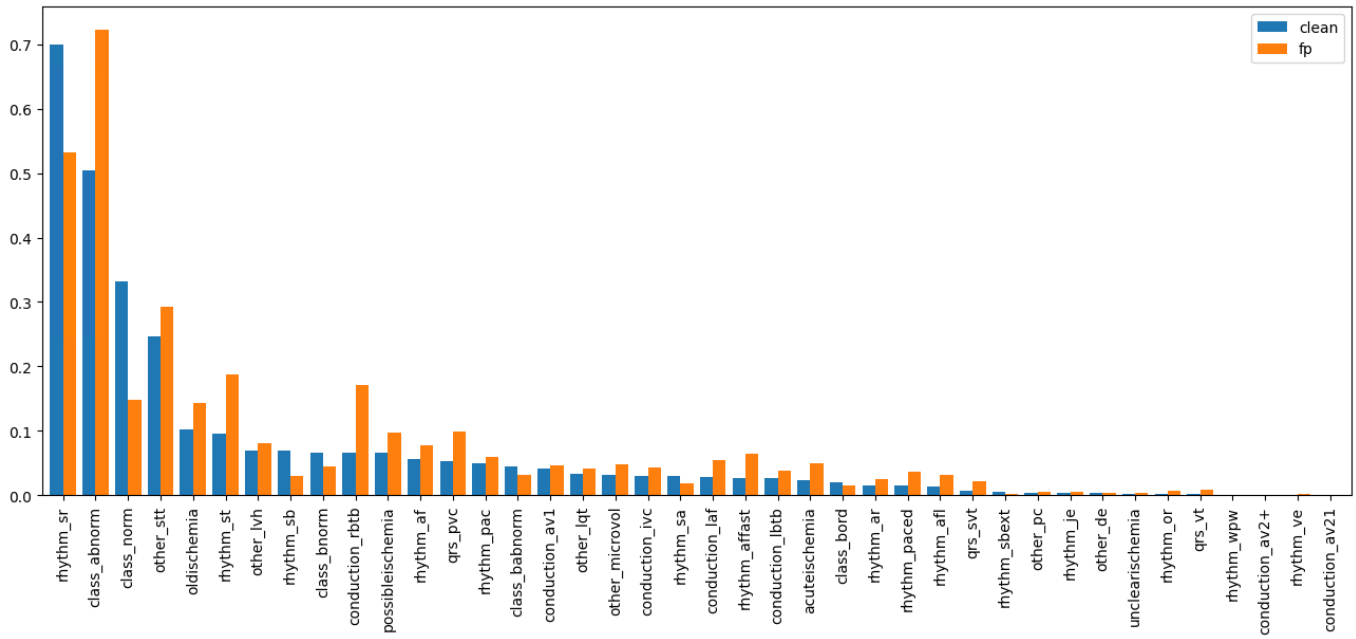
Scaling further increases the ROC-AUC to 0.732 (95% CI: [0.727, 0.737]). Testing on the noisy data after they are filtered with the Butterworth filter however decreases the ROC-AUC to 0.717 (95% CI: [0.711, 0.723]).

We choose a classification threshold of 20 on the reconstruction losses empirically based on the best model performance (scaling, no Butterworth filter). With this threshold we get overall accuracy of 0.70. The ROC curve, the scaled confusion matrix and the classification report are found in Figure 35, Figure 36 and Table 13 respectively.

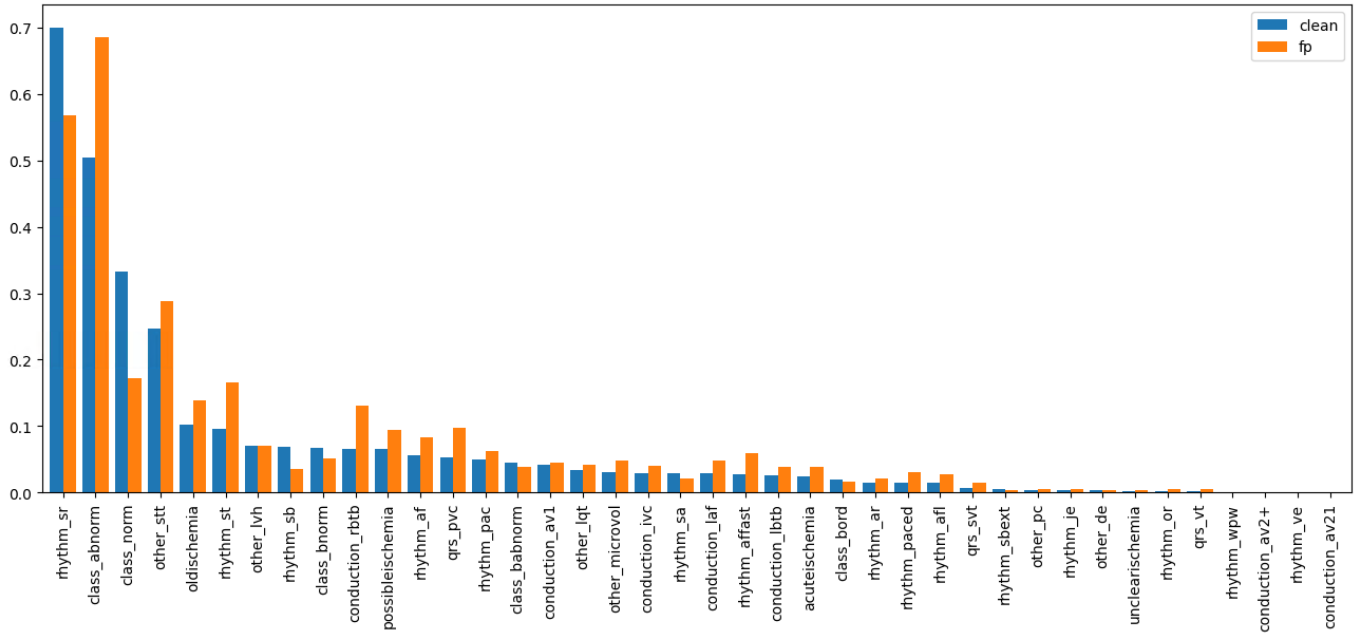
**Clean split B.** The model achieves slightly better results when tested on Clean split B, with ROC-AUC of 0.698 (95% CI: [0.692, 0.703]) without scaling and 0.740 (95% CI: [0.735, 0.745]) with scaling. When the data is reduced by the Butterworth filter the ROC-AUC decreases again to 0.724 (95% CI: [0.719, 0.730]). Using the same threshold as in split A we obtain accuracy of 0.72. The ROC curve, the scaled confusion matrix and the classification report are found in Figure 37, Figure 38 and Table 14 respectively.

**Expert-annotated data.** The model achieves ROC-AUC of 0.723 (95% CI: [0.557, 0.863]) without scaling and 0.760 (95% CI: [0.611, 0.891]) when scaling is applied. Thresholding as in Clean split A, we get an accuracy of 0.71. Figure 39, Figure 40 and Table 15 present the ROC curve, the scaled confusion matrix and the classification report for this classifier on the expert-annotated data.

**PhysioNet 2011 data.** The model achieves ROC-AUC of 0.970 (95% CI: [0.940, 0.991]) without scaling and 0.968 (95% CI: [0.940, 0.989]) with scaling. When thresholding with the same value as before, we get an accuracy of 0.79. We present the ROC curve, the scaled confusion matrix and the classification report in Figure 37, Figure 38 and Table 14 respectively.




(a) Label distribution in false positives before weighting.




(b) Label distribution in false positives after weighting.

Figure 13: Diagnostic label distribution in FP compared with clean samples before and after augmentations and weighted sampling.

## 7.3 Feature space clustering

We now move on to the experimentation process for the feature space clustering method, which is explained in Section 3.2.2, *Feature space clustering for OOD detection*.

### 7.3.1 Experiments

For the feature space clustering experiments we choose a feature space of 128 dimensions. As mentioned, the three architectures that were used for the feature space clustering experiments were AE, DiagCl and SimCLR. Once each architecture was trained for its particular task, the double residual encoder part of each was extracted in order to obtain a feature extractor

$$F : \mathbb{R}^{8\times 5000} \to \mathbb{R}^{128}.$$

For the AE training process refer to Section 7.2.1, *Experiments*. For the DiagCl training process refer to Section 7.4.1, *Experiments*. In this section we only discuss the SimCLR training process.

**SimCLR Loss.** As mentioned in Section 3.2.2, *Feature space clustering for OOD detection*, the loss used for the training of SimCLR is the InfoNCE loss, which is described in detail in [24].

**SimCLR training.** We train the SimCLR architecture on Clean split A over 30 epochs using the AdamW optimiser and the OneCycle learning rate scheduler, with maximum learning rate of $10^{-4}$. The training and validation losses over the training epochs are shown in Figure 27.

**In-distribution embeddings.** The in-distribution embeddings are obtained via the 294,046 ECGs in the training set of Clean split A. The embeddings are then clustered using a GMM. The model was tested using 1-4 clusters. We also tried reducing the number of dimensions using PCA while keeping 99% of the data's variance, and clustering on the reduced data.

### 7.3.2 Results

**Clean split A, Clean split B, expert-annotated set.** This method did not seem to achieve any useful separation between the two classes for any of the UMCU splits, consistently failing to reach ROC-AUC above 0.6 regardless of the number of clusters used for the GMM or whether the embeddings were reduced or not. Table 8 shows the best ROC-AUC scores for each feature extractor (all of which were achieved for a single GMM cluster of the data). We do not report on any further classification metrics for these splits, as they are not of interest.

| | **AE** | **DiagCl** | **SimCLR** |
|---|---|---|---|
| **Clean split A** | 0.512 | 0.514 | 0.523 |
| **Clean split B** | 0.526 | 0.517 | 0.531 |
| **Expert-annotated** | 0.552 | 0.554 | 0.555 |

Table 8: ROC-AUC of each feature space extractor for each UMCU split.

Figure 14 compares the distribution of the Mahalanobis distances of the clean and the OOD data of the Clean split A test set from the training-data distribution, as calculated by a single cluster GMM. Ideally we would expect some separation between the two, with the OOD distances having generally higher values. However, the distributions seem almost identical. Figure 15 shows 2-D PCA plots of the embeddings obtained from the training set, as well as the clean and the noisy parts of the Clean split A test set. Once again, we observe no separation between their distribution in space.

**PhysioNet 2011 data.** The algorithm had surprisingly strong performance on the PhysioNet 2011 data:

- AE achieved a max ROC-AUC of 0.931 (95% CI: [0.896, 0.960]).
- DiagCl achieved a max ROC-AUC of 0.840 (95% CI: [0.725, 0.935]).
- SimCLR achieved a max ROC-AUC of 0.933 (95% CI: [0.865, 0.972]).

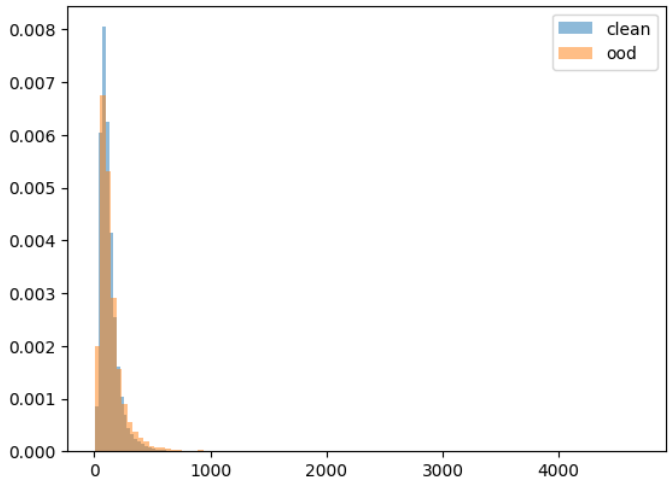



Figure 14: Mahalanobis distance distribution for AE on Clean split A.

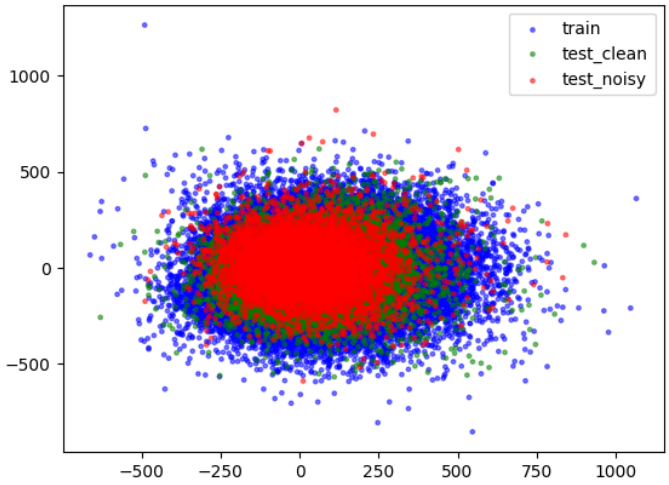



Figure 15: PCA on features extracted from AE on Clean split A.

The above scores were obtained for a single GMM cluster after PCA was applied. In particular for SimCLR, by keeping 99% of the data's variance we go down from 128 to 6-dimensions. The ROC curve for this classifier can be seen in Figure 43. By comparing the Mahalanobis distances of the test samples with the CDF of the chi-squared distribution with 6 degrees of freedom, and rejecting a sample $x$ as noisy if

$$P(X \leq M(x, D_{train})) = 0.99 \iff P(X \geq M(x, D_{train})) = 0.01$$

(where $X$ denotes a random variable that follows the chi-squared distribution with 6 degrees of freedom), we get a classifier that achieves accuracy of 0.82. The confusion matrix and classification report for this classifier show strong performance on the noise detection task, and can be found in Figure 44 and Table 17 respectively.

## 7.4 Uncertainty based methods

Finally, building upon Section 3.3, *Uncertainty based methods*, we discuss our experiments with uncertainty as a measure of noise.

### 7.4.1 Experiments

For this method we mainly use the DiagCl architecture (see Section 6.3, *Diagnostic classifier*).

**Loss.** As explained in the corresponding section, the DiagCl model is simply an encoder with 5 independent classification heads, 4 of which correspond to multi-class problems, while the other is multi-label. As a result, the DiagCl model is optimised with respect to a weighted average of 4 Cross Entropy (CE) losses and 1 Binary Cross Entropy (BCE) Loss.

**Training.** The model is trained on Clean split B over 30 epochs using the AdamW optimizer and the ReduceLROnPlateau learning rate scheduler, starting with a learning rate of $10^{-4}$. Random augmentations as well as weighted sampling are used during the traing process. Weight initialisation by a pre-trained SimCLR network was also attempted, without leading to huge improvements. The above process is repeated 5 times, using different random weight initializations each time. The training metrics can be found in Figure 28.

**Uncertainty estimation.** The dropout layers of each network's classification heads remain active upon inference. Given an input $x$, 4 different predictions are made by each network using randomly generated dropout masks. As a result, for each unique input $x$ we get $5 \times 4 = 20$ independent diagnostic label predictions. The mean of these predictions can be used for inference, while their variance can be used as a measure of epistemic uncertainty, effectively combining the ensemble and Monte Carlo methods for uncertainty estimation.

**Ensemble's predictive capabilities.** The ensemble shows good performance overall on the task at which it was trained for, although it does struggle with a few under-represented classes. Detailed classification metrics can be found in Figures 45 to 48 and table 18.

### 7.4.2 Results

We aim to use the uncertainty estimation as a measure of noise (RQ3), with our hypothesis being that high uncertainty may indicate an OOD ECG. When testing on any of the three test sets however, we observe the opposite effect, obtaining ROC-AUC less than 0.5 on every set. Plotting the uncertainty estimations for the clean and the noisy data of the test set of Clean split B, we see that it is indeed common for the classifier to exhibit higher uncertainty in the clean data rather than the noisy data. Counterintuitively, this effect is even more pronounced with the data that is in Clean split A but not in Clean split B, i.e. those that are excluded purely because of their diagnostic labels; the classifier seems to exhibit even less uncertainty on this data (referred to as "mislabelled" in Figure 16).

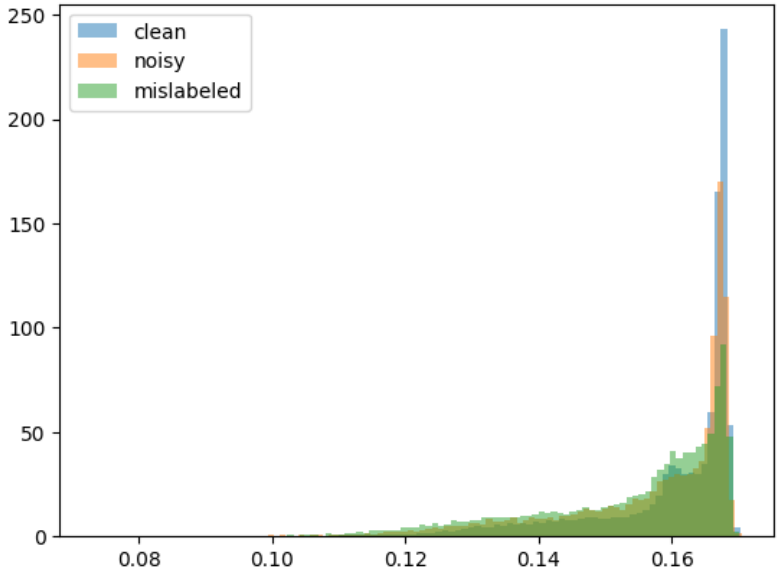



Figure 16: Uncertainty estimations for the Clean split B test data, as well as the "mislabelled" data.

A similar effect is observed in the corresponding plots for the expert-annotated set and the PhysioNet 2011 data (Figure 17 and Figure 18 respectively).

## 7.5 Summary of best ROC-AUC scores

Table 9 summarises the maximum ROC-AUC scores that each method achieved on each dataset that it was tested on.

| | **Binary class.** | **Reconstruction** | **Feature space** | **Uncertainty** |
|---|---|---|---|---|
| **Binary class.** | 0.907 | *n/a* | *n/a* | *n/a* |
| **Clean split A** | *n/a* | 0.732 | 0.523 | *n/a* |
| **Clean split B** | *n/a* | 0.740 | 0.531 | $< 0.5$ |
| **Expert-ann.** | **0.978** | 0.760 | 0.555 | $< 0.5$ |
| **PhysioNet** | 0.909 | **0.970** | **0.933** | $< 0.5$ |

Table 9: Best ROC-AUC scores achieved by each method on each dataset.

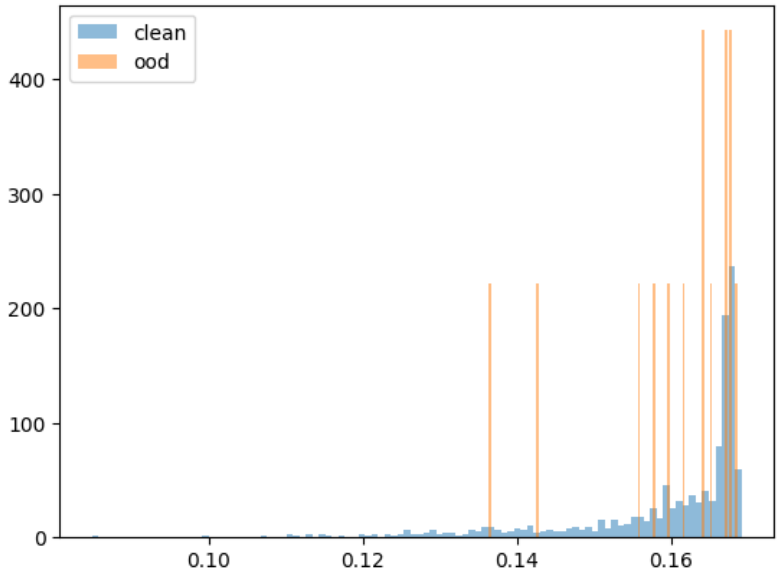



Figure 17: Uncertainty estimations for the expert-annotated data.

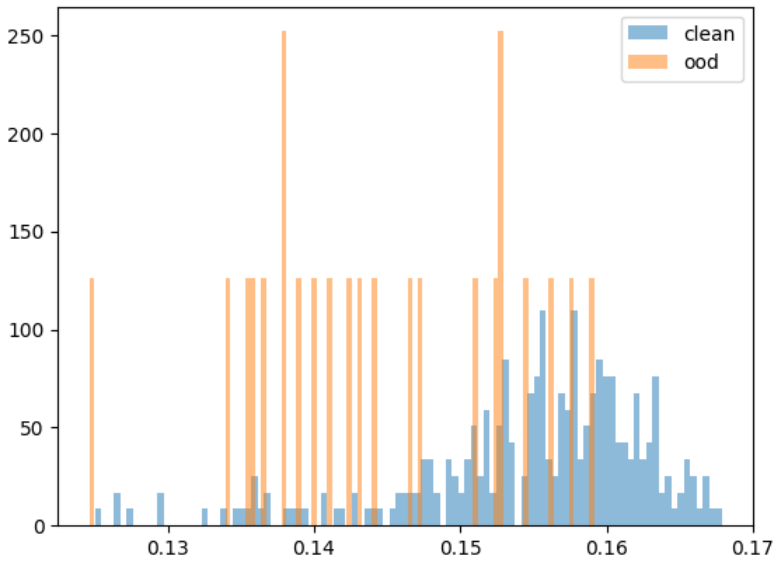



Figure 18: Uncertainty estimations for the PhysioNet 2011 data.

# 8 Discussion

We now discuss our results in relation to the research questions as defined in Section 1.1, *Research questions*.

**RQ1.** Our experiments show that for ECG noise detection, binary classification seems to be by far the preferred method. While supervised learning often provides the best results when labelled data is available, it also has its drawbacks:

- Labelled data is not always available, and creating a sufficiently large labelled dataset from the beginning requires expert knowledge and significant time investment. Moreover, the network's performance relies significantly on the label quality (see Section 8.1, *Limitations*).
- A classifier that has been trained on a single labelled dataset often performs worse for the same task on different datasets.
- The network's decisions are generally hard to explain.

The reconstruction methods show some promise on the noise detection task, although they do not reach the same performance as the binary classifier on the UMCU data. Regardless, the AE can be quite useful in a noise detecting pipeline. For a start, its decisions are easy to interpret, as one can visually inspect the reconstructions. Moreover, not all of the FP generated by this method are necessarily FP, as they may also be mislabelled data. Figure 19 shows one such example, and during error analysis many more were discovered. On the other hand, as shown by the FN examples in which the AE produced excellent reconstructions without the high-frequency noise, the AE can also be used for noise removal instead of noise detection.

The feature space methods did not produce good results for the noise detection task. While SimCLR is one of the easiest contrastive learning techniques in terms of implementation, it is probably not suitable for this task, as it is trained to produce similar embeddings for similar signals. It is therefore likely that a normal ECG will have a closer embedding to a noisy version of itself rather than to an ECG with severe pathology. Thus, the method is probably more suited for diagnostic anomaly detection.

**RQ2.** The two main techniques that were found to be useful for the AE method in particular were the introduction of augmentations and weighted sampling during training. These techniques were also used during the training of the other methods, and improved the results empirically. Other ideas that were attempted, as mentioned in Section 7.2.1, *Experiments*, were less successful. More sophisticated methods may be required to achieve better results on this topic.

**RQ3.** So far in our experiments, uncertainty has shown to be unsuccessful in terms of predicting the noise level or the uninterpretability of an ECG. However, our experimentation on this topic is not broad enough to give a conclusive answer.

## 8.1 Limitations

While this study aims to cover a broad selection of techniques, it comes with certain limitations.

**Unknown quality of available labels.** Noise generally appears to an extent in almost every ECG signal, so it is often complicated even for an expert to judge the level and type of noise required for an ECG to be considered unsuitable for diagnostic purposes. Moreover, it might be the case that even if some leads (or part of leads) of the ECG are very noisy, those that are left are still sufficient for a diagnosis, at which point it is up to the expert's judgement whether they would label the ECG as noisy or not.

**Restriction to 8-lead ECGs.** This work focuses exclusively on classifying 8-lead ECGs as clean or noisy. However, strong arguments can be made for the case of detecting noise in specific leads, or specific parts of each lead instead:

- Many, but not all machines produce 8+ lead ECGs. Machines with fewer leads are sometimes used in ambulances or during routine check-ups.

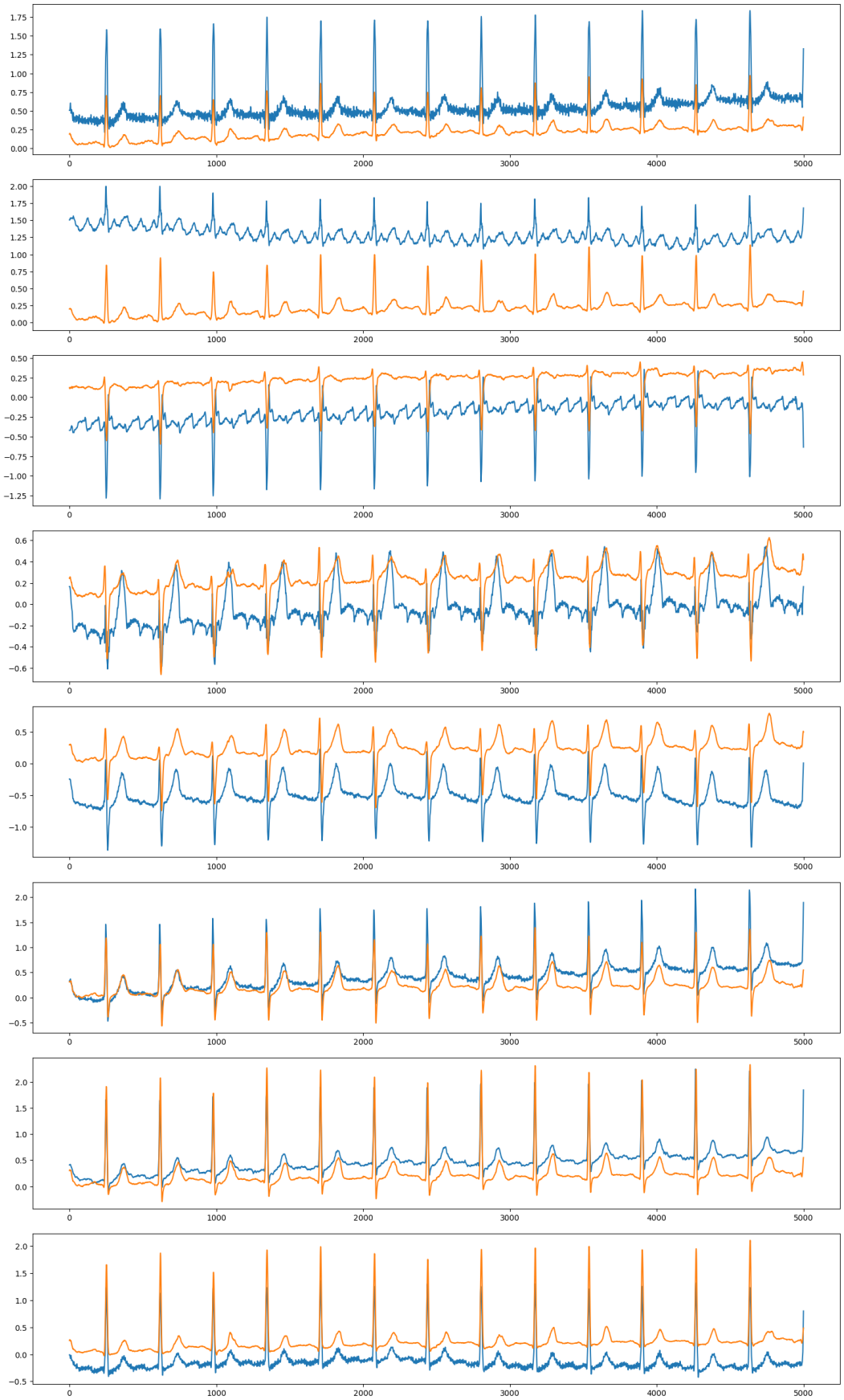


Figure 19: Mislabelled FP sample with high reconstruction loss.

- Even if one or more of the leads is noisy, the ECG might still be diagnostically interpretable.
- An algorithm trained to recognise noise within a lead might have better performance, since its predictions are not influenced by the other leads.

The labels given to the original UMCU dataset however do not specify the location of the noise, as such detailed labelling is significantly more time consuming. While the expert-annotated segmentation dataset was eventually available during the internship, this dataset is much smaller, and by that time the main algorithms and pipelines used in this work had already been developed.

# 9 Future Work

Based on our experimentation and results, we can suggest relevant research directions such as the following.

- Regarding RQ1, experiments with different number of leads, or including segmentation information may improve the performance of the noise-detection algorithms.
- More experimentation is needed with respect to RQ2. The development of dedicated methods to tackle this topic specifically might lead to better results.
- Regarding RQ3, the role of uncertainty, how it relates to ECGs that are hard to interpret, and how it influences the predictions of an automated diagnostic system is a very interesting and potentially rich research topic.
- While noise detection itself is important, the task may be modified to better suit the needs of an automated diagnostic system. For example, it might be the case that an ECG with minor or localised noise may still be interpetable by an automated system. Noise removal techniques may also be employed. Whether the exclusion of ECGs that are flagged by an OOD detector leads to more accurate predictions for an automated system is also something that should be tested.
- Apart from off-line noise detection, an interesting topic is whether the signal can be detected as noisy at the time of acquisition, using time-series anomaly detection techniques.

# References


[1] Selcan Kaplan Berkaya et al. "A survey on ECG analysis". In: *Biomedical Signal Processing and Control* 43 (2018), pp. 216–235. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2018.03.003. URL: https://www.sciencedirect.com/science/article/pii/S1746809418300636.

[2] Chandrasekhar Nataraj, Ali Jalali, and Parham Ghorbanian. "Application of Computational Intelligence Techniques for Cardiovascular Diagnostics". In: Apr. 2012. ISBN: 978-953-51-0534-3. DOI: 10.5772/38032.

[3] Stefan P Nelwan et al. "Reconstruction of the 12-lead electrocardiogram from reduced lead sets". In: *Journal of Electrocardiology* 37.1 (2004), pp. 11–18. ISSN: 0022-0736. DOI: https://doi.org/10.1016/j.jelectrocard.2003.10.004. URL: https://www.sciencedirect.com/science/article/pii/S0022073603001729.

[4] David A Cook, So-Young Oh, and Martin V Pusic. "Accuracy of physicians' electrocardiogram interpretations: a systematic review and meta-analysis". In: *JAMA internal medicine* 180.11 (2020), pp. 1461–1471.

[5] Shubhojeet Chatterjee et al. "Review of noise removal techniques in ECG signals". In: *IET Signal Processing* 14.9 (2020), pp. 569–590. DOI: https://doi.org/10.1049/iet-spr.2020.0104. eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-spr.2020.0104. URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-spr.2020.0104.

[6] Jeroen F Vranken et al. "Uncertainty estimation for deep learning-based automated analysis of 12-lead electrocardiograms". In: *European Heart Journal - Digital Health* 2.3 (May 2021), pp. 401–415. ISSN: 2634-3916. DOI: 10.1093/ehjdh/ztab045. eprint: https://academic.oup.com/ehjdh/article-pdf/2/3/401/47116803/ztab045.pdf. URL: https://doi.org/10.1093/ehjdh/ztab045.

[7] Luca Neri et al. "Validation of a New and Straightforward Algorithm to Evaluate Signal Quality during ECG Monitoring with Wearable Devices Used in a Clinical Setting". In: *Bioengineering* 11.3 (2024). ISSN: 2306-5354. DOI: 10.3390/bioengineering11030222. URL: https://www.mdpi.com/2306-5354/11/3/222.

[8] Reza Sameni et al. "A Nonlinear Bayesian Filtering Framework for ECG Denoising". In: *Biomedical Engineering, IEEE Transactions on* 54 (Jan. 2007), pp. 2172–2185.

[9] Guohua Lu et al. "Removing ECG noise from surface EMG signals using adaptive filtering". In: *Neuroscience Letters* 462.1 (2009), pp. 14–19. ISSN: 0304-3940. DOI: https://doi.org/10.1016/j.neulet.2009.06.063. URL: https://www.sciencedirect.com/science/article/pii/S0304394009008593.

[10] Jakub Kuzilek et al. "Independent Component Analysis and Decision Trees for ECG Holter Recording De-Noising". In: *PloS one* 9 (June 2014), e98450. DOI: 10.1371/journal.pone.0098450.

[11] Peng Cui and Jinjia Wang. "Out-of-Distribution (OOD) Detection Based on Deep Learning: A Review". In: *Electronics* 11.21 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11213500. URL: https://www.mdpi.com/2079-9292/11/21/3500.

[12] Ugo Lomoio et al. *AUTAN-ECG: An AUToencoder bAsed system for anomaly detectioN in ECG signals*. 2023. DOI: 10.36227/techrxiv.24638856.

[13] Sahar Soltanieh, Javad Hashemi, and Ali Etemad. "In-Distribution and Out-of-Distribution Self-Supervised ECG Representation Learning for Arrhythmia Detection". In: *IEEE Journal of Biomedical and Health Informatics* 28.2 (Feb. 2024), pp. 789–800. ISSN: 2168-2208. DOI: 10.1109/jbhi.2023.3331626. URL: http://dx.doi.org/10.1109/JBHI.2023.3331626.

[14] Bayu Wijaya Putra et al. "Abnormality Heartbeat Classification of ECG Signal Using Deep Neural Network and Autoencoder". In: *2019 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*. 2019, pp. 213–218. DOI: 10.1109/ICIMCIS48181.2019.8985206.

[15] Yoon Dukyong et al. "Deep Learning-Based Electrocardiogram Signal Noise Detection and Screening Model". In: *hir* 25.3 (2019), pp. 201–211. DOI: 10.4258/hir.2019.25.3.201. eprint: http://www.e-sciencecentral.org/articles/?scid=1130331. URL: http://www.e-sciencecentral.org/articles/?scid=1130331.

[16] Sardar Ansari, Jonathan Gryak, and Kayvan Najarian. "Noise Detection in Electrocardiography Signal for Robust Heart Rate Variability Analysis: A Deep Learning Approach". In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2018, pp. 5632–5635. DOI: 10.1109/EMBC.2018.8513537.

[17] Radhika Dua et al. *Automatic Detection of Noisy Electrocardiogram Signals without Explicit Noise Labels*. 2022. arXiv: 2208.08853 [eess.SP]. URL: https://arxiv.org/abs/2208.08853.

[18] Corneliu T.C. Arsene, Richard Hankins, and Hujun Yin. "Deep Learning Models for Denoising ECG Signals". In: *2019 27th European Signal Processing Conference (EUSIPCO)*. 2019, pp. 1–5. DOI: 10.23919/EUSIPCO.2019.8902833.

[19] Karol Antczak. *Deep Recurrent Neural Networks for ECG Signal Denoising*. 2019. arXiv: 1807.11551 [cs.NE]. URL: https://arxiv.org/abs/1807.11551.

[20] Siti Nurmaini et al. "Deep Learning-Based Stacked Denoising and Autoencoder for ECG Heartbeat Classification". In: *Electronics* 9.1 (2020). ISSN: 2079-9292. DOI: 10.3390/electronics9010135. URL: https://www.mdpi.com/2079-9292/9/1/135.

[21] scikit-learn. *Gaussian mixture models*. https://scikit-learn.org/stable/modules/mixture.html. Accessed: 2024-08-07.

[22] P.C. Mahalanobis. "On the generalized distance in statistics". In: *Proceedings of the National Institute of Science of India* 2 (1936), pp. 49–55.

[23] Ting Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *CoRR* abs/2002.05709 (2020). arXiv: 2002.05709. URL: https://arxiv.org/abs/2002.05709.

[24] Aäron van den Oord and Yazhe Li and Oriol Vinyals. "Representation Learning with Contrastive Predictive Coding". In: *CoRR* abs/1807.03748 (2018). arXiv: 1807.03748. URL: http://arxiv.org/abs/1807.03748.

[25] Chuan Guo et al. *On Calibration of Modern Neural Networks*. 2017. arXiv: 1706.04599 [cs.LG]. URL: https://arxiv.org/abs/1706.04599.

[26] Ramneet Kaur et al. *Detecting OODs as datapoints with High Uncertainty*. 2021. arXiv: 2108.06380 [cs.LG]. URL: https://arxiv.org/abs/2108.06380.

[27] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006). ROC Analysis in Pattern Recognition, pp. 861–874. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2005.10.010. URL: https://www.sciencedirect.com/science/article/pii/S016786550500303X.

[28] Karimollah Hajian-Tilaki. "Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation". In: *Caspian journal of internal medicine* 4 (Sept. 2013), pp. 627–635.

[29] B. Efron. "Bootstrap Methods: Another Look at the Jackknife". In: *The Annals of Statistics* 7.1 (1979), pp. 1–26. DOI: 10.1214/aos/1176344552. URL: https://doi.org/10.1214/aos/1176344552.

[30] Ikaro Silva, George B Moody, and Leo Celi. "Improving the Quality of ECGs Collected Using Mobile Phones: The PhysioNet/Computing in Cardiology Challenge 2011". In: *Computers in Cardiology* 38 (2011), pp. 273–276.

[31] Ary L Goldberger et al. "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals". In: *Circulation [Online]* 101.23 (2000), e215–e220.

[32] Ikaro Silva, George B. Moody, and Leo Anthony Celi. *Improving the Quality of ECGs Collected using Mobile Phones: The PhysioNet/Computing in Cardiology Challenge 2011*. Accessed: 2024-06-13. 2011. URL: https://moody-challenge.physionet.org/2011/.

[33] Liang Xie et al. "Computational Diagnostic Techniques for Electrocardiogram Signal Analysis". In: *Sensors (Basel)* 20.21 (Nov. 2020), p. 6318. DOI: 10.3390/s20216318.

[34] Awni Y Hannun et al. "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network". In: *Nature medicine* 25.1 (2019), pp. 65–69.

[35] Kartik Gupta et al. *Understanding and Improving the Role of Projection Head in Self-Supervised Learning*. 2022. arXiv: 2212.11491 [cs.LG]. URL: https://arxiv.org/abs/2212.11491.

[36] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection". In: *CoRR* abs/1708.02002 (2017). arXiv: 1708.02002. URL: http://arxiv.org/abs/1708.02002.

[37] Ilya Loshchilov and Frank Hutter. "Fixing Weight Decay Regularization in Adam". In: *CoRR* abs/1711.05101 (2017). arXiv: 1711.05101. URL: http://arxiv.org/abs/1711.05101.

[38] Yibo Zhou. *Rethinking Reconstruction Autoencoder-Based Out-of-Distribution Detection*. 2023. arXiv: 2203.02194 [cs.CV]. URL: https://arxiv.org/abs/2203.02194.

[39] Sahar Soltanieh, Ali Etemad, and Javad Hashemi. *Analysis of Augmentations for Contrastive ECG Representation Learning*. 2022. arXiv: 2206.07656 [eess.SP]. URL: https://arxiv.org/abs/2206.07656.

[40] S. Butterworth. "On the Theory of Filter Amplifiers". In: *Experimental Wireless and the Wireless Engineer* 7 (1930), pp. 536–541.

[41] Leslie N. Smith and Nicholay Topin. "Super-Convergence: Very Fast Training of Residual Networks Using Large Learning Rates". In: *CoRR* abs/1708.07120 (2017). arXiv: 1708.07120. URL: http://arxiv.org/abs/1708.07120.

# 10 Appendix

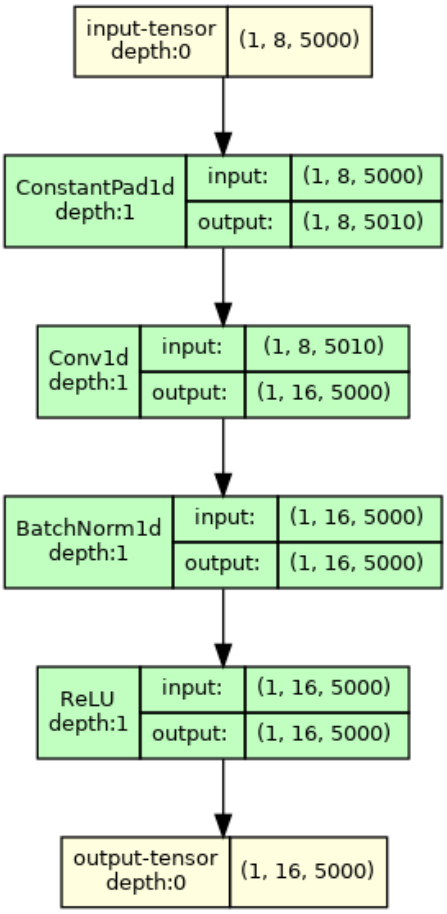



Figure 20: ConvBlock architecture.

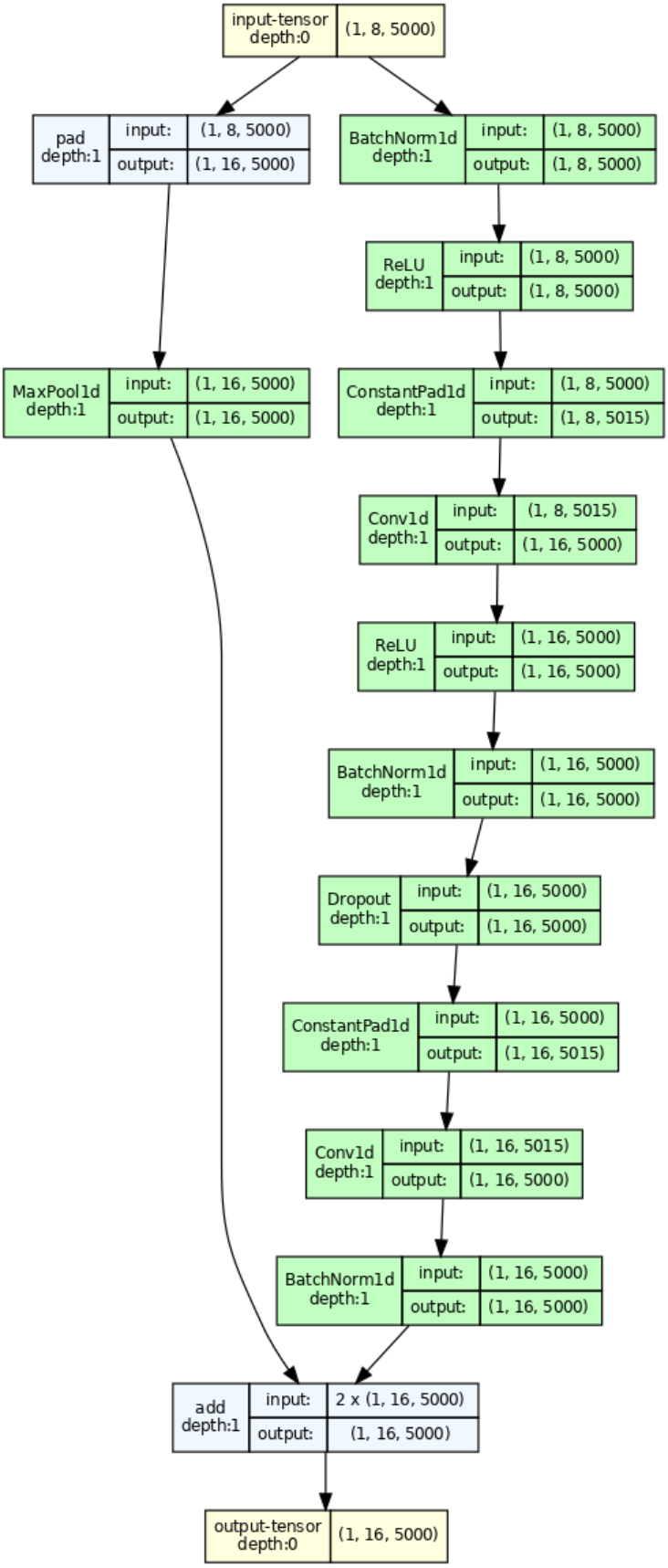



Figure 21: ResidualMaxPoolDoubleConvBlockForward architecture.

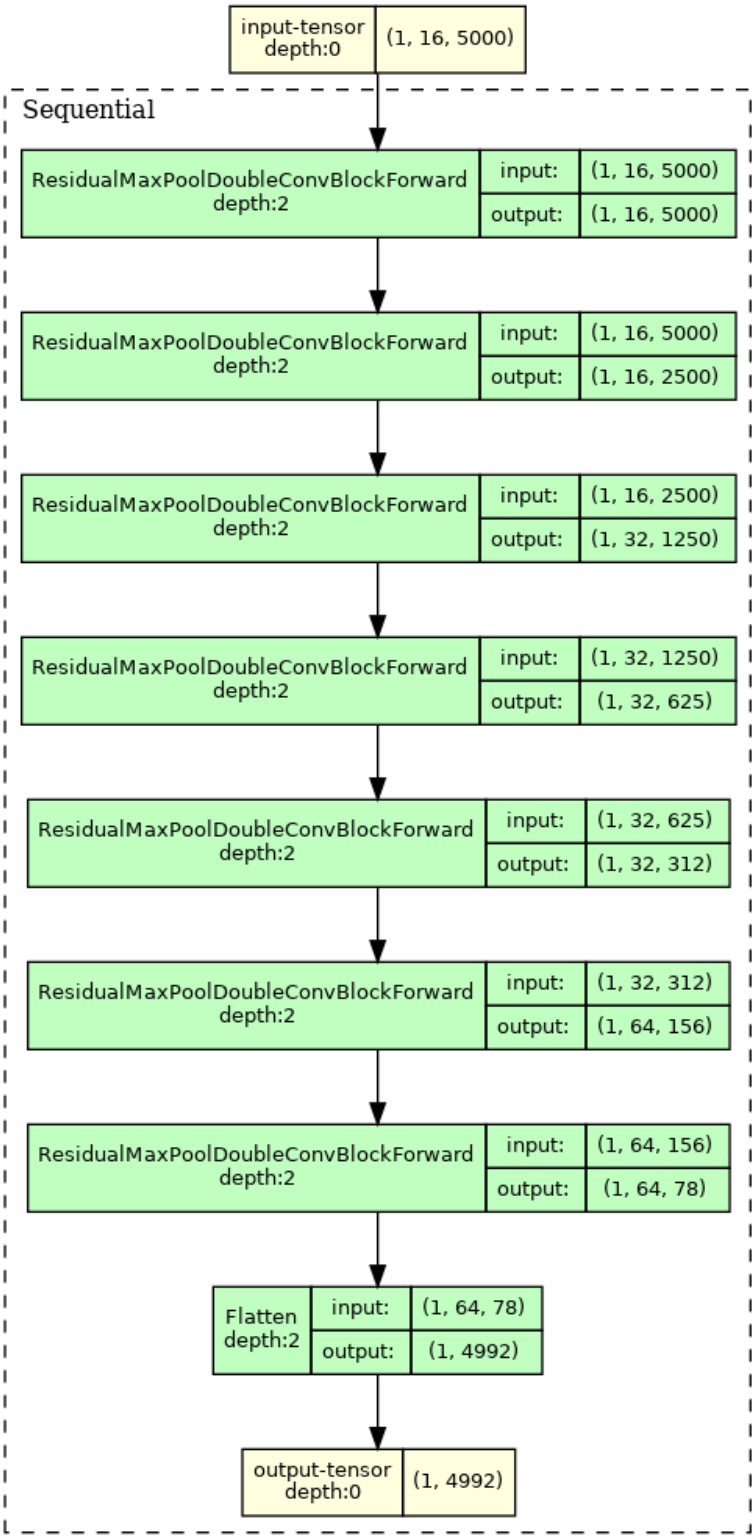



Figure 22: CNNDoubleResidual architecture.

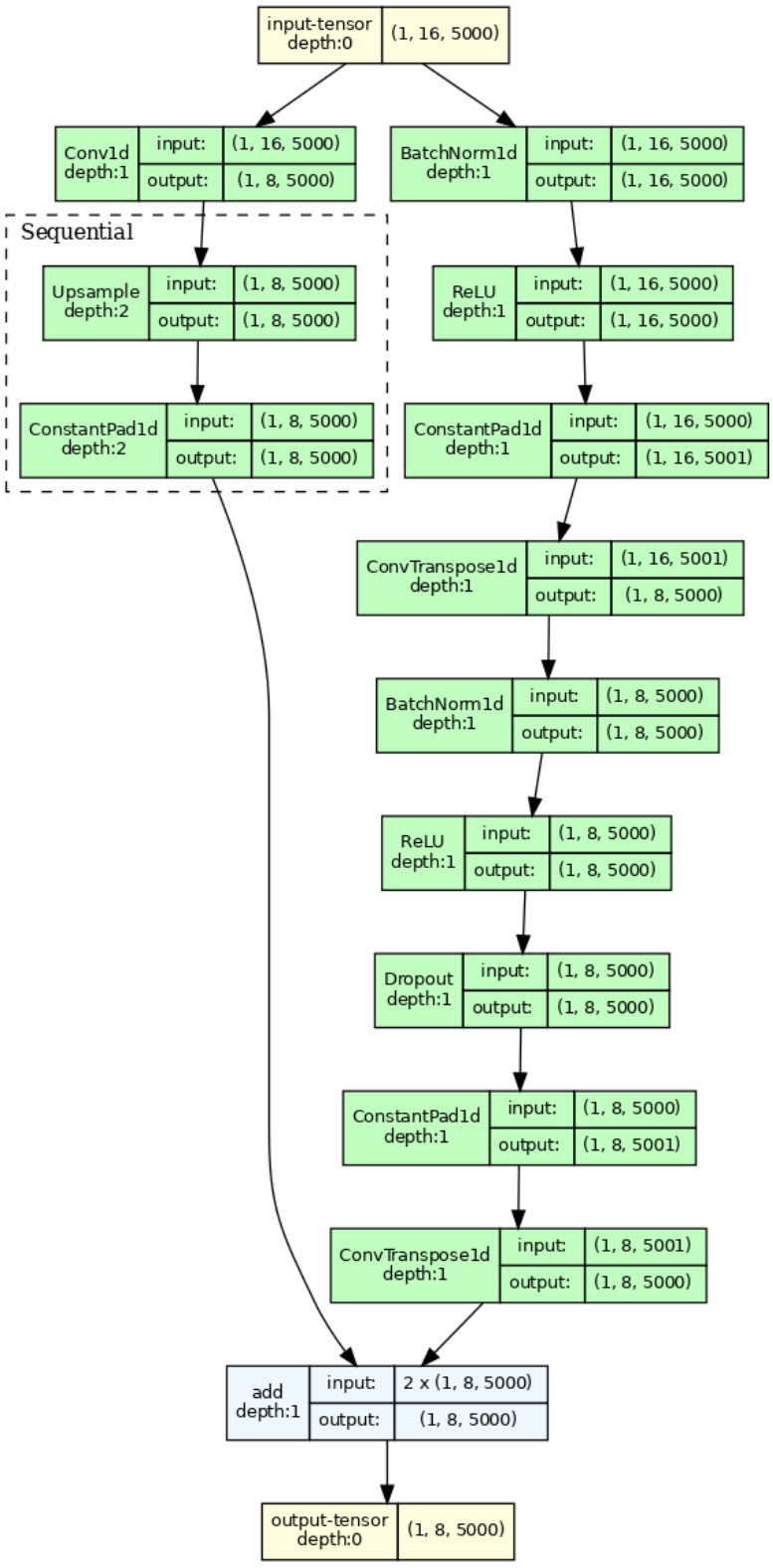



Figure 23: ResidualMaxPoolDoubleConvBlockBackward architecture.

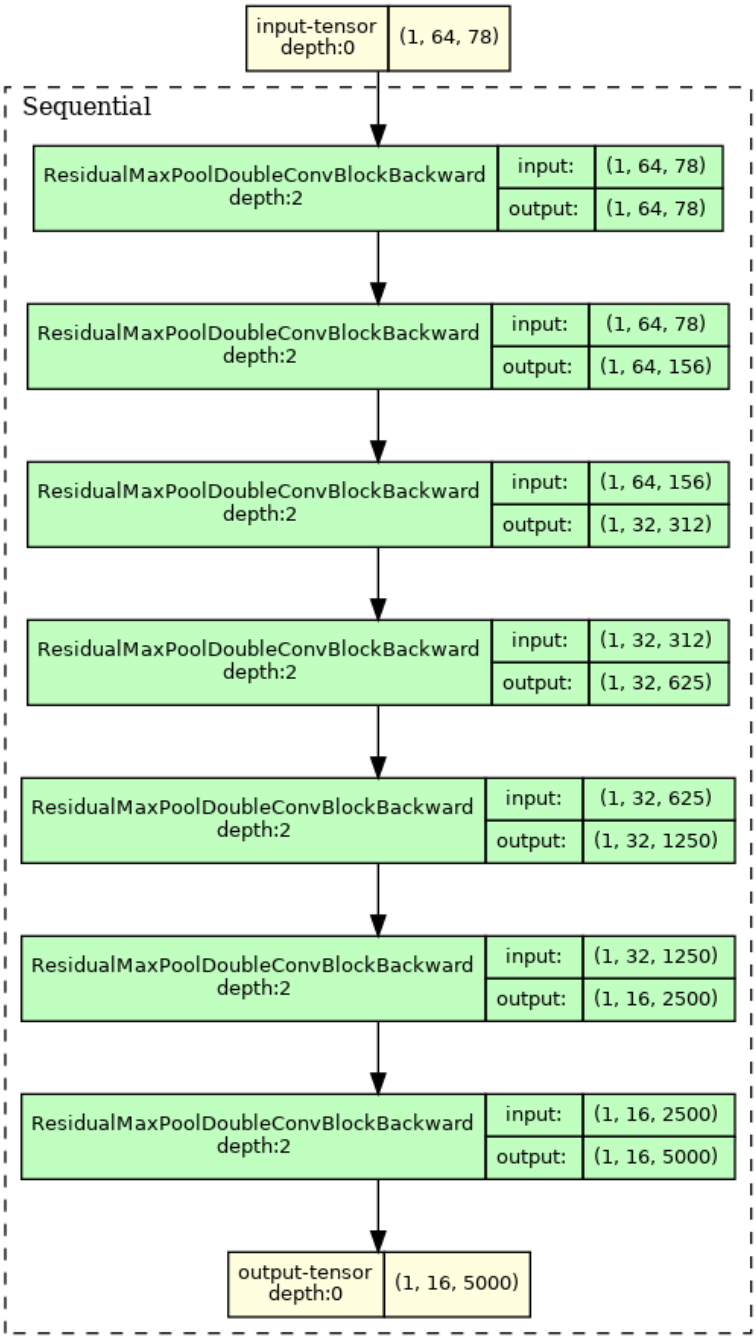
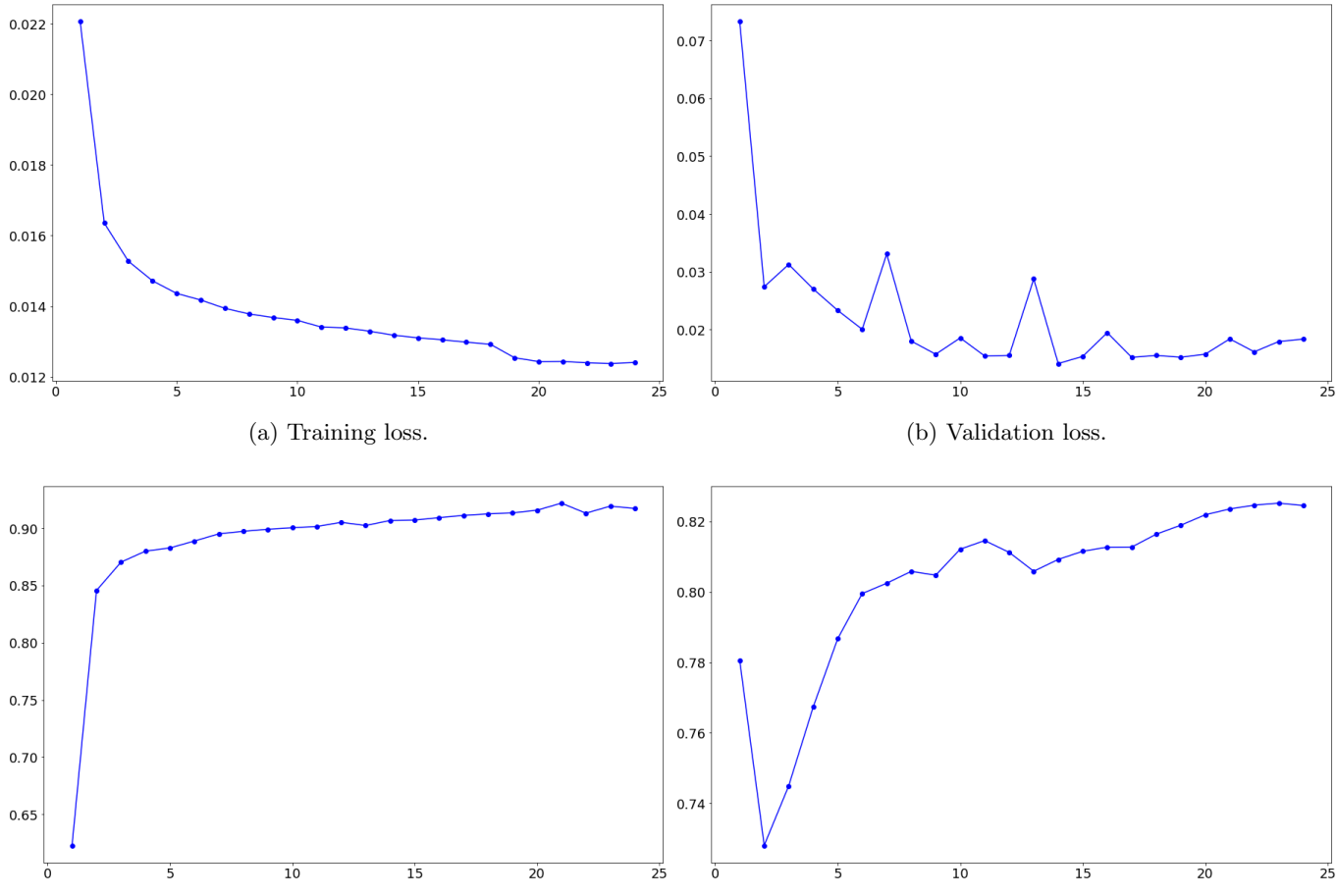



Figure 24: CNNDoubleResidualBackward architecture.



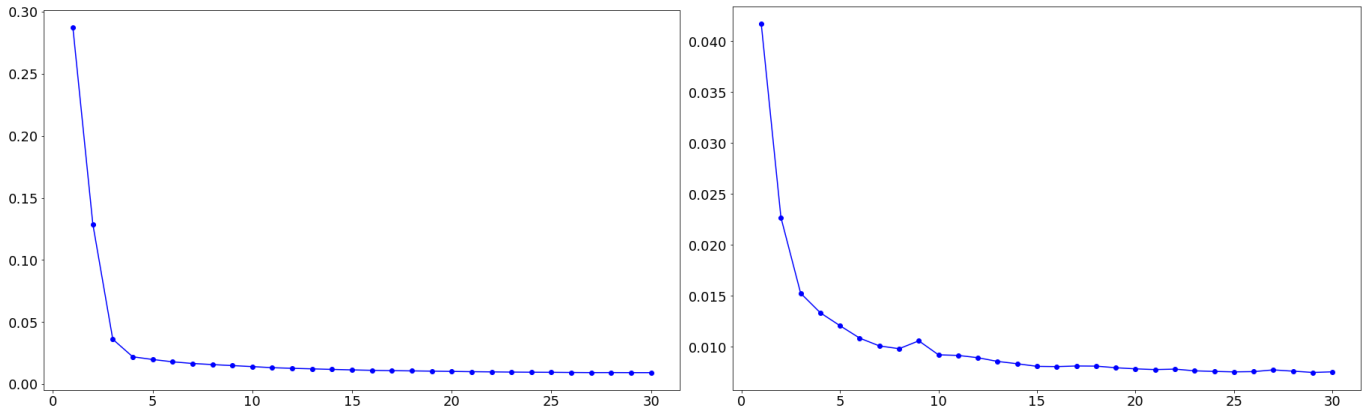

(a) Training loss.

(b) Validation loss.

(c) Training ROC-AUC.

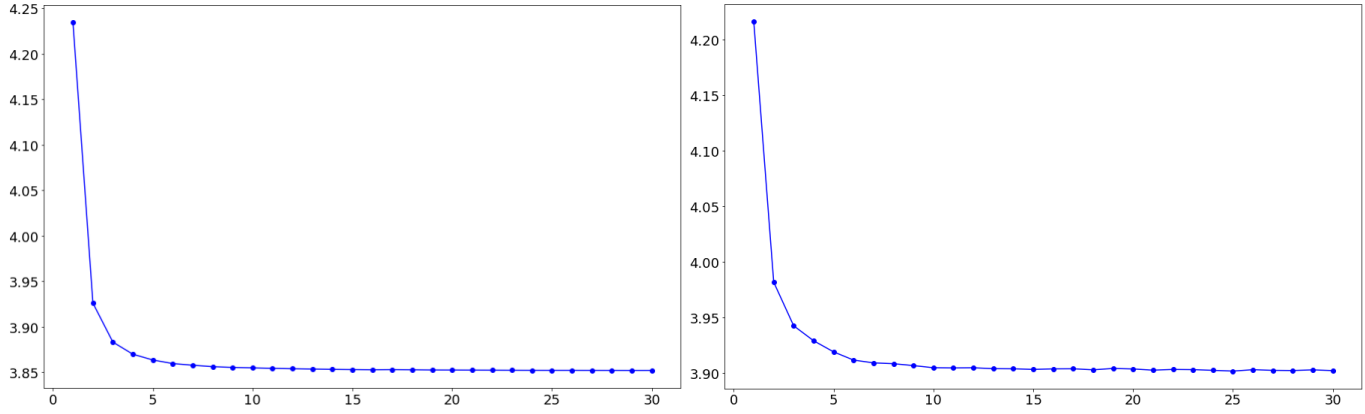
(d) Validation ROC-AUC.

Figure 25: Metrics during the training of the BinCl.



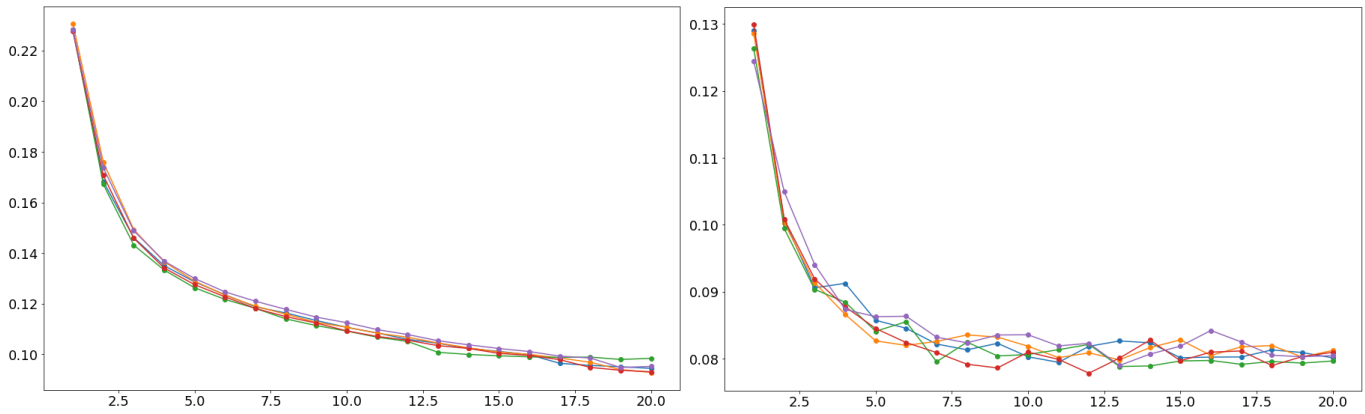

(a) Training loss. (b) Validation loss.

Figure 26: Metrics during the training of the AE.




(a) Training loss. (b) Validation loss.

Figure 27: Metrics during the training of SimCLR.




(a) Training loss. (b) Validation loss.

Figure 28: Metrics during the training of the DiagCl ensemble.

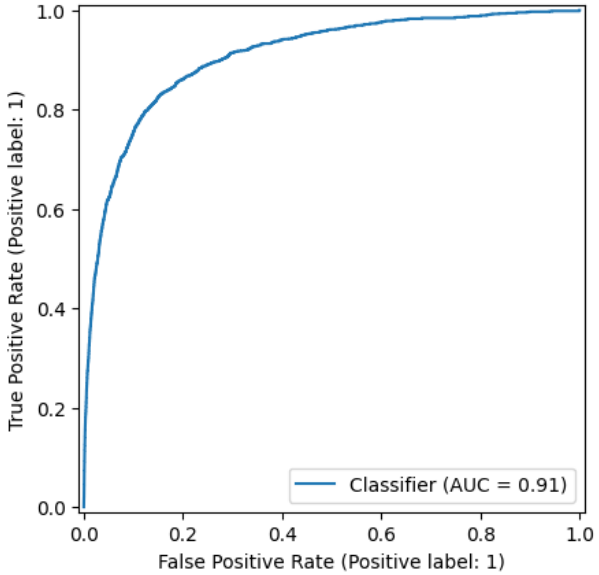



Figure 29: BinCl on binary classification data: ROC-AUC curve.

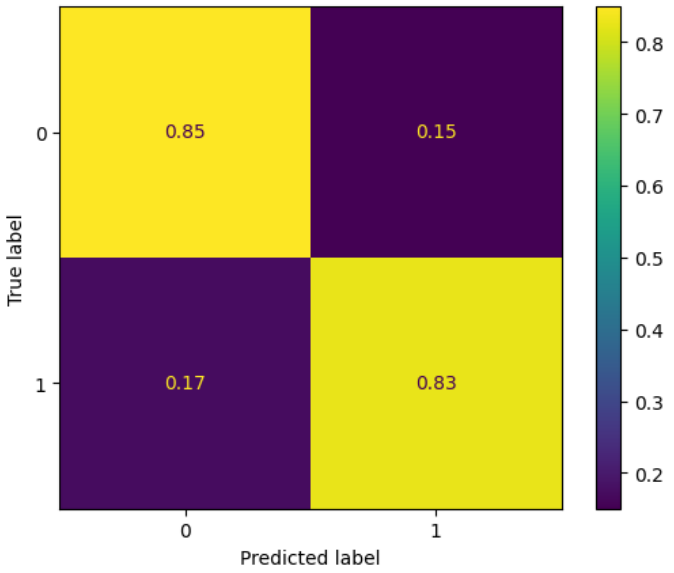



Figure 30: BinCl on binary classification data: normalised confusion matrix.

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| **Clean** | 0.99 | 0.85 | 0.92 | 37,069 |
| **Noisy** | 0.17 | 0.83 | 0.28 | 1,340 |
| **Average** | 0.58 | 0.84 | 0.60 | 38,409 |
| **Weighted average** | 0.96 | 0.85 | 0.89 | 38,409 |

Table 10: BinCl on binary classification data: classification report.

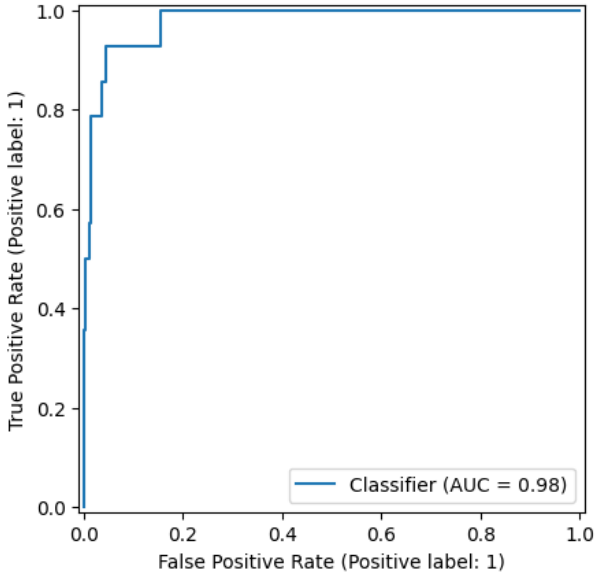



Figure 31: BinCl on Expert-annotated set: ROC-AUC curve.

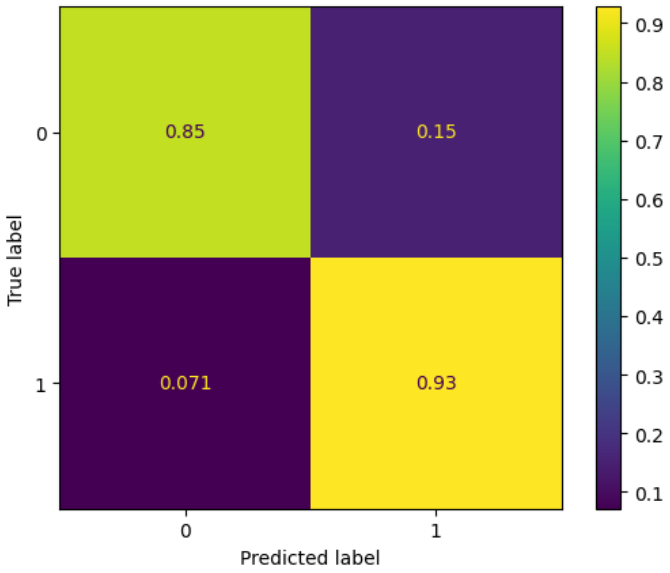



Figure 32: BinCl on Expert-annotated set: normalised confusion matrix.

| | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| **Clean** | 1.00 | 0.74 | 0.85 | 276 |
| **Noisy** | 0.23 | 0.96 | 0.38 | 23 |
| **Average** | 0.61 | 0.85 | 0.61 | 299 |
| **Weighted average** | 0.94 | 0.76 | 0.81 | 299 |

Table 11: BinCl on PhysioNet 2011 data: classification report.

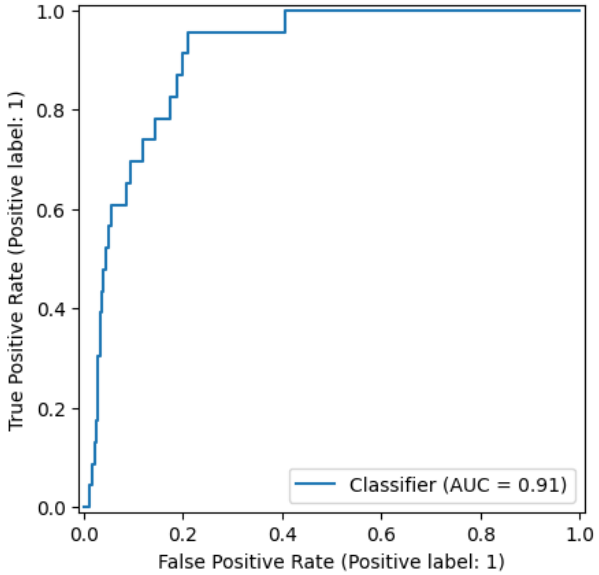



Figure 33: BinCl on PhysioNet 2011 data: ROC-AUC curve.

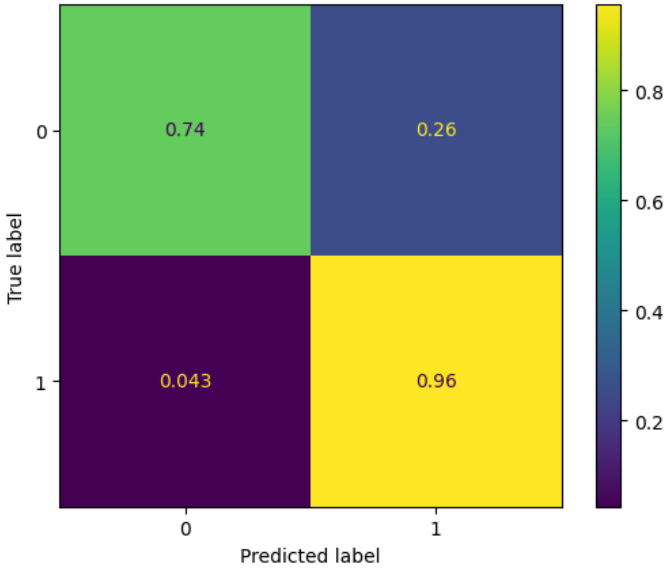



Figure 34: BinCl on PhysioNet 2011 data: normalised confusion matrix.

| | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| **Clean** | 1.00 | 0.74 | 0.85 | 276 |
| **Noisy** | 0.23 | 0.96 | 0.38 | 23 |
| **Average** | 0.61 | 0.85 | 0.61 | 299 |
| **Weighted average** | 0.94 | 0.76 | 0.81 | 299 |

Table 12: BinCl on PhysioNet 2011 data: classification report.

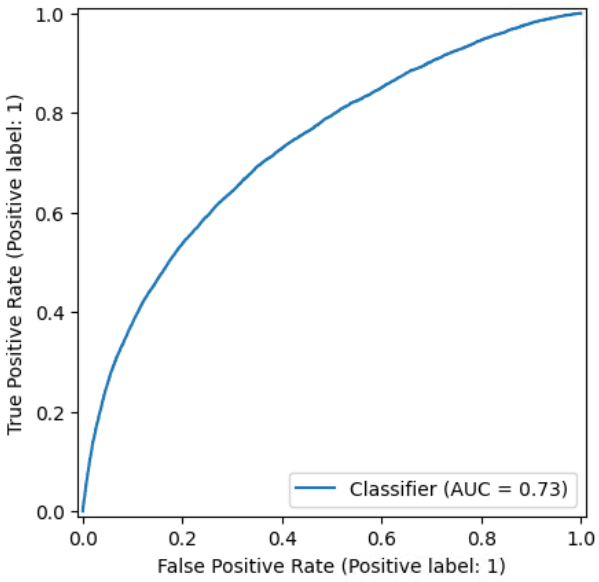



Figure 35: AE on Clean split A: ROC-AUC curve.

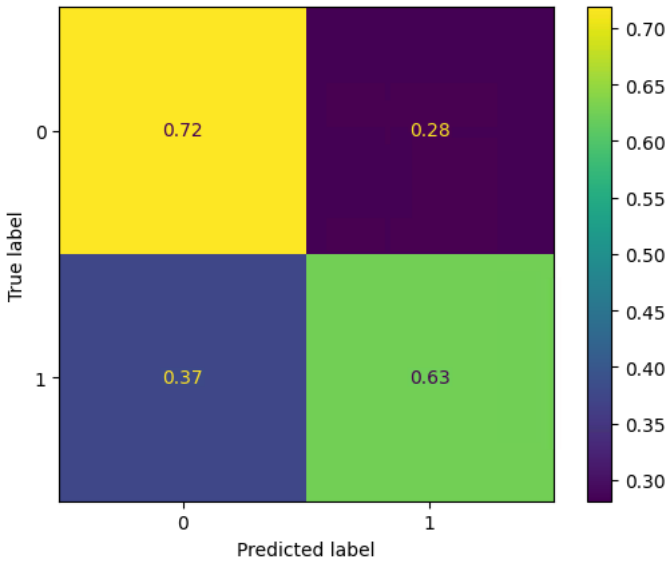



Figure 36: AE on Clean split A: normalised confusion matrix.

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| **Clean** | 0.85 | 0.72 | 0.78 | 36,593 |
| **Noisy** | 0.43 | 0.63 | 0.51 | 12,612 |
| **Average** | 0.64 | 0.67 | 0.65 | 49,205 |
| **Weighted average** | 0.74 | 0.70 | 0.71 | 49,205 |

Table 13: AE on Clean split A: classification report.

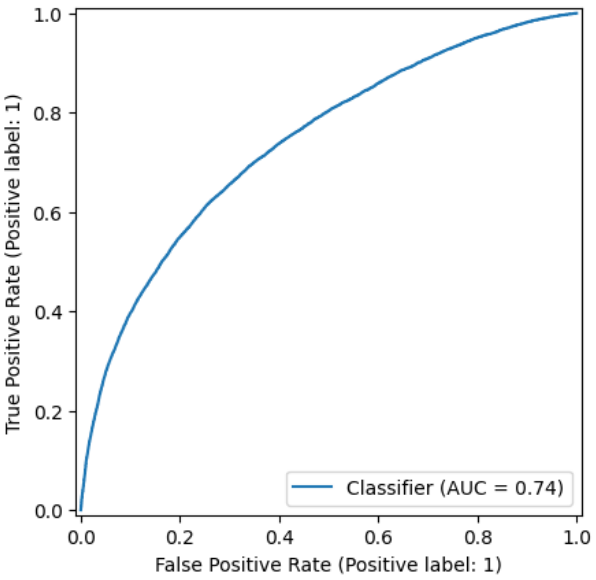



Figure 37: AE on Clean split B: ROC-AUC curve.

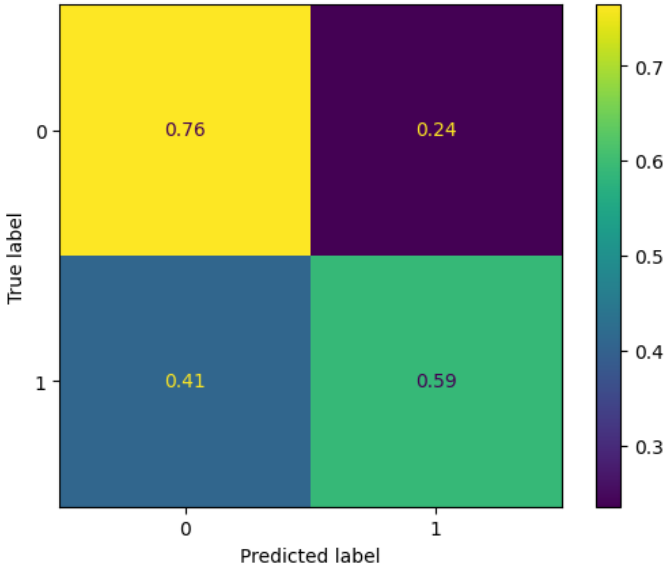



Figure 38: AE on Clean split B: normalised confusion matrix.

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| **Clean** | 0.84 | 0.76 | 0.80 | 35,097 |
| **Noisy** | 0.47 | 0.59 | 0.53 | 12,612 |
| **Average** | 0.66 | 0.68 | 0.66 | 47,709 |
| **Weighted average** | 0.74 | 0.72 | 0.73 | 47,709 |

Table 14: AE on Clean split B: classification report.

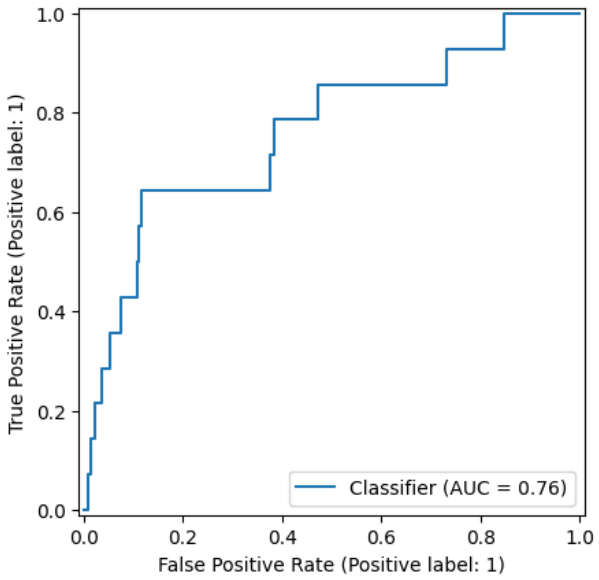



Figure 39: AE on expert-annotated data: ROC-AUC curve.

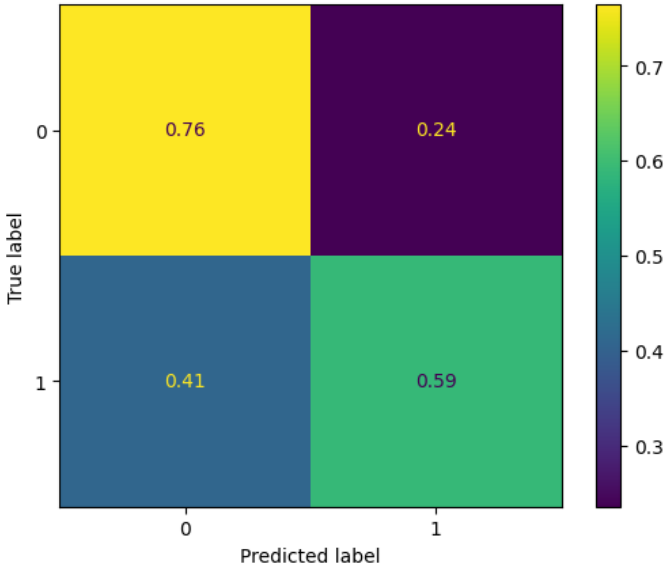



Figure 40: AE on expert-annotated data: normalised confusion matrix.

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| **Clean** | 0.99 | 0.71 | 0.83 | 962 |
| **Noisy** | 0.03 | 0.64 | 0.06 | 14 |
| **Average** | 0.51 | 0.68 | 0.44 | 976 |
| **Weighted average** | 0.98 | 0.71 | 0.82 | 976 |

Table 15: AE on expert-annotated data: classification report.

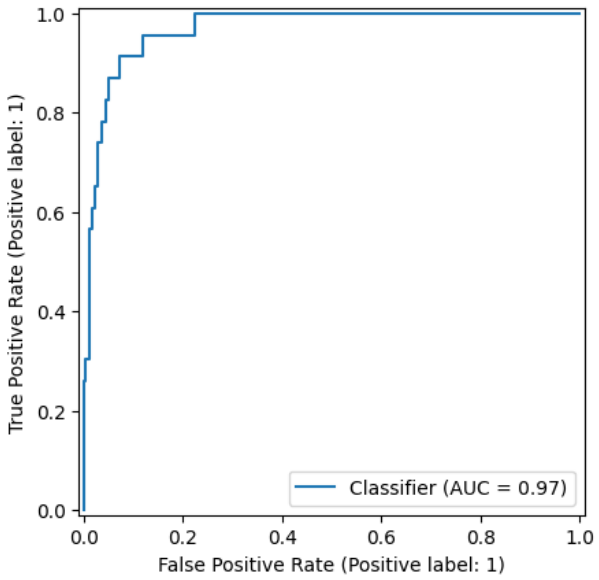



Figure 41: AE on PhysioNet 2011: ROC-AUC curve.

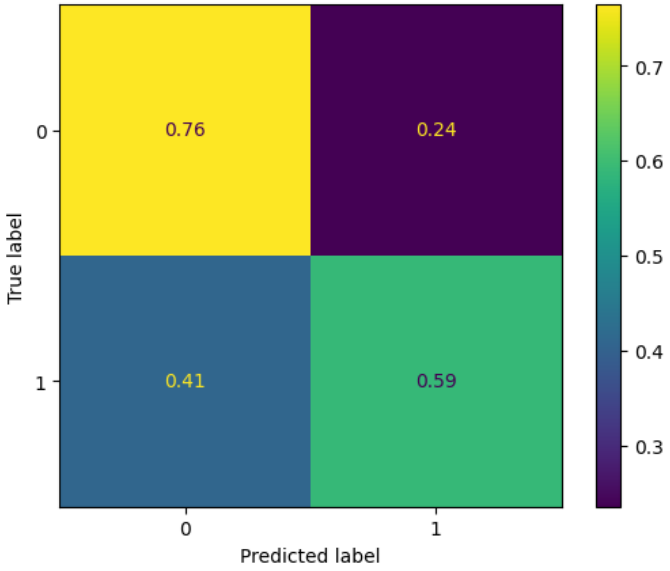



Figure 42: AE on PhysioNet 2011: normalised confusion matrix.

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| **Clean** | 1.00 | 0.78 | 0.87 | 276 |
| **Noisy** | 0.26 | 0.96 | 0.41 | 23 |
| **Average** | 0.63 | 0.87 | 0.64 | 299 |
| **Weighted average** | 0.94 | 0.79 | 0.84 | 299 |

Table 16: AE on PhysioNet 2011: classification report.

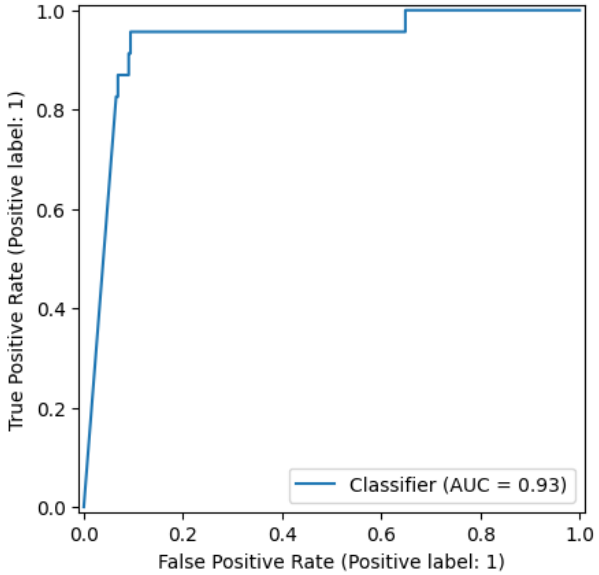



Figure 43: SimCLR for feature clustering on PhysioNet 2011: ROC-AUC curve.

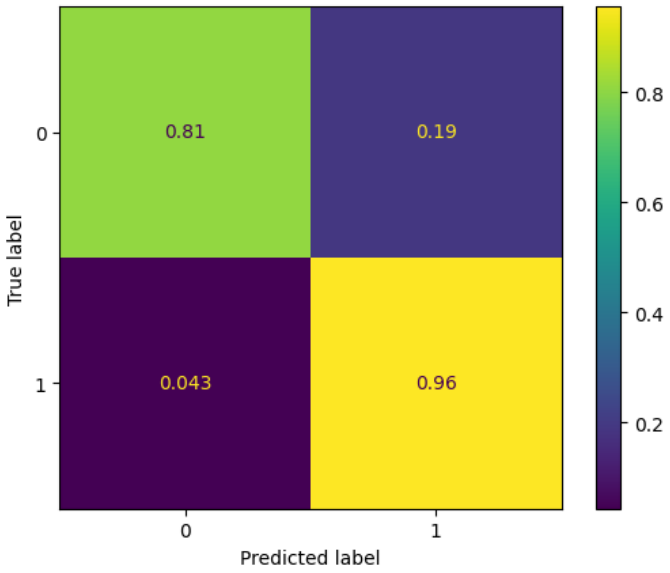



Figure 44: AE on PhysioNet 2011: normalised confusion matrix.

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| **Clean** | 1.00 | 0.81 | 0.89 | 276 |
| **Noisy** | 0.29 | 0.96 | 0.45 | 23 |
| **Average** | 0.64 | 0.88 | 0.67 | 299 |
| **Weighted average** | 0.94 | 0.82 | 0.86 | 299 |

Table 17: SimCLR for feature clustering on PhysioNet 2011: classification report.

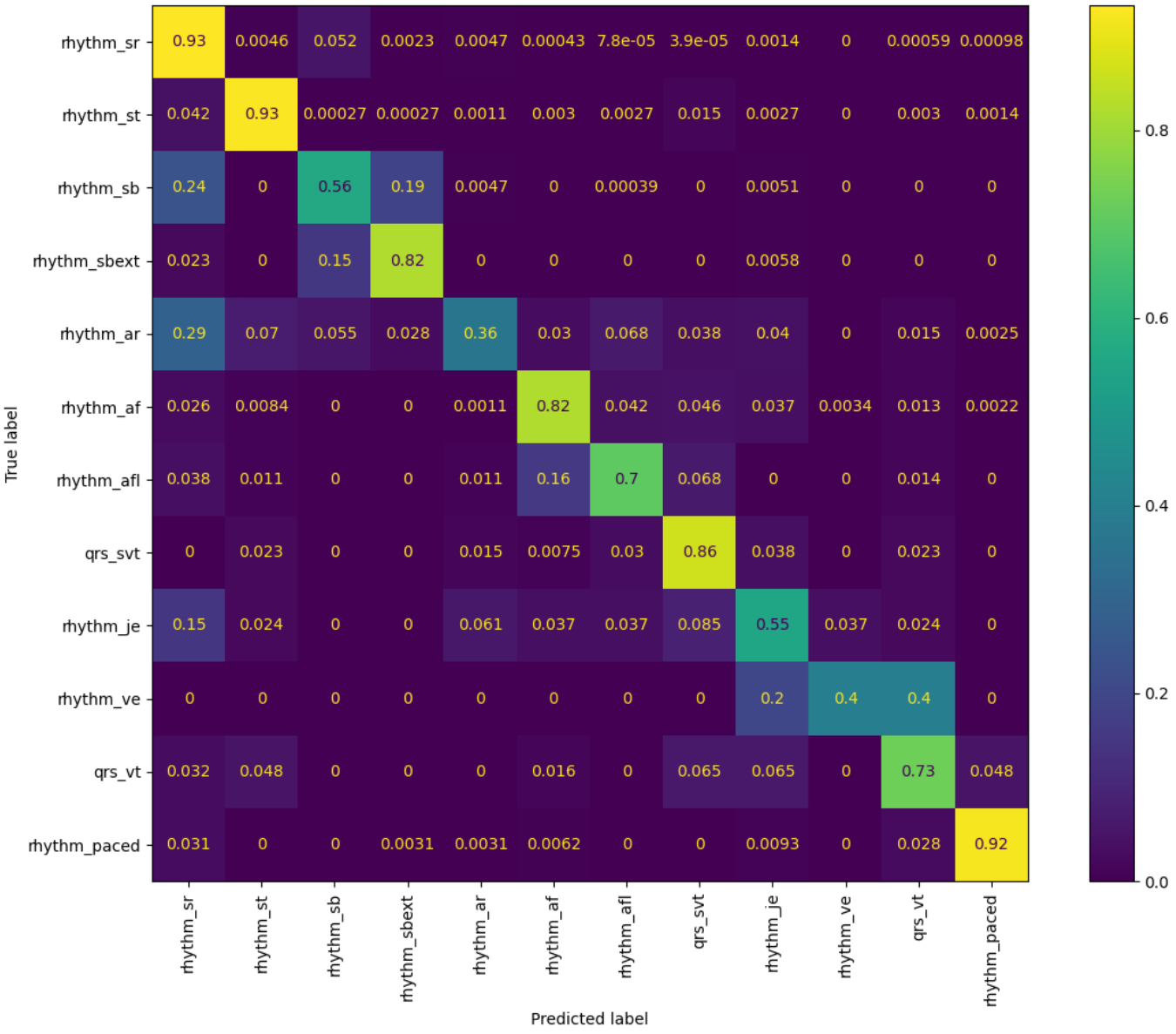



Figure 45: DiagCl ensemble, Rhythm Type labels: confusion matrix.

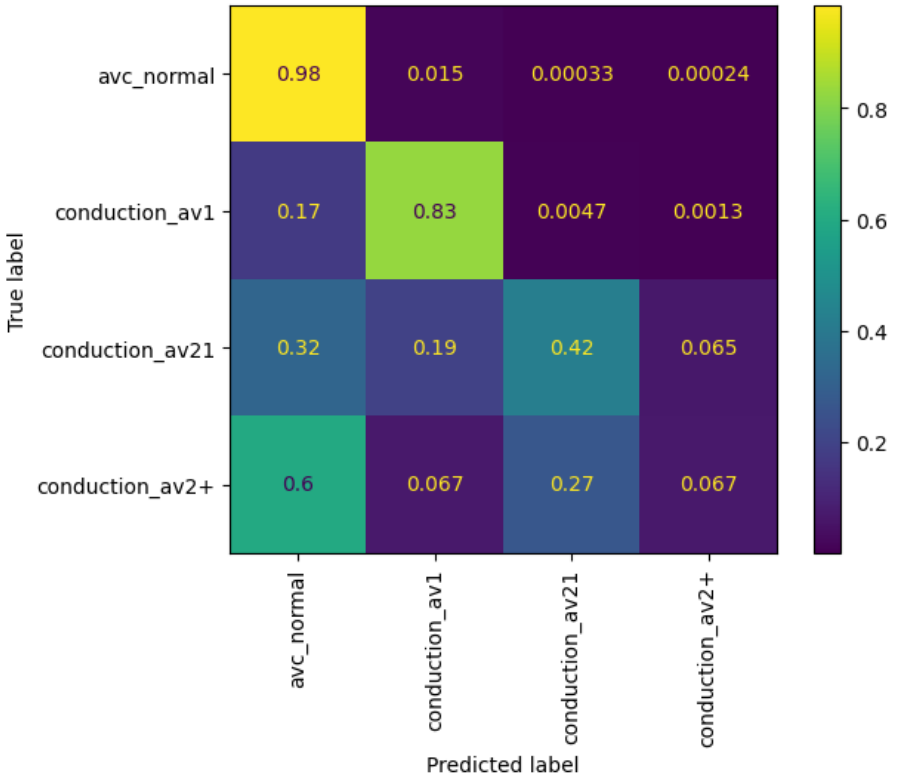



Figure 46: DiagCl ensemble, AVC labels: confusion matrix.

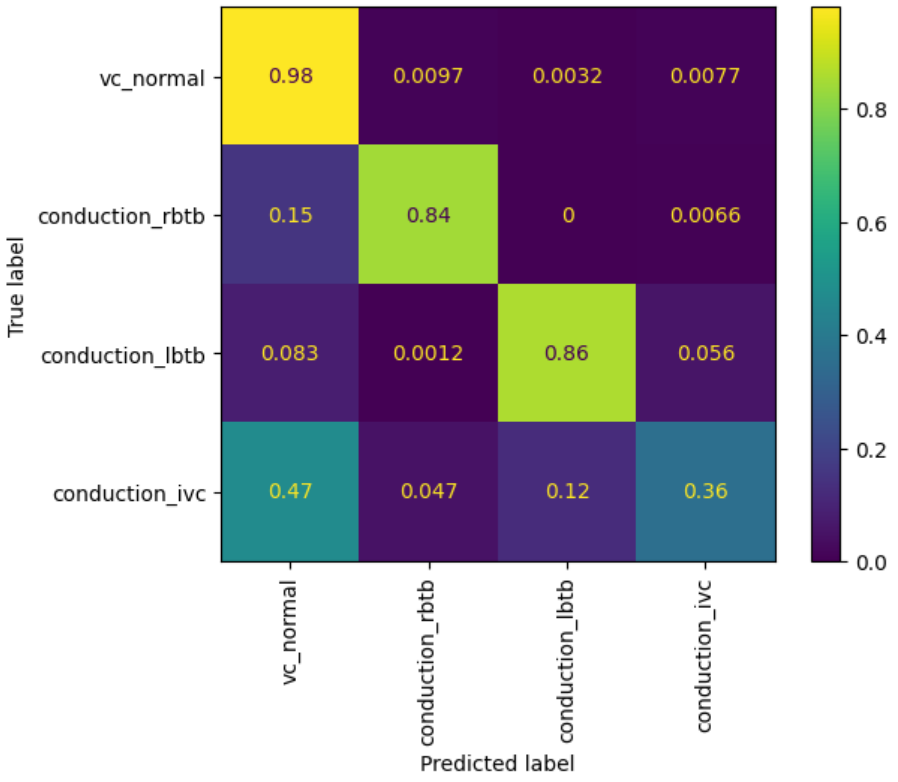



Figure 47: DiagCl ensemble, VC labels: confusion matrix.

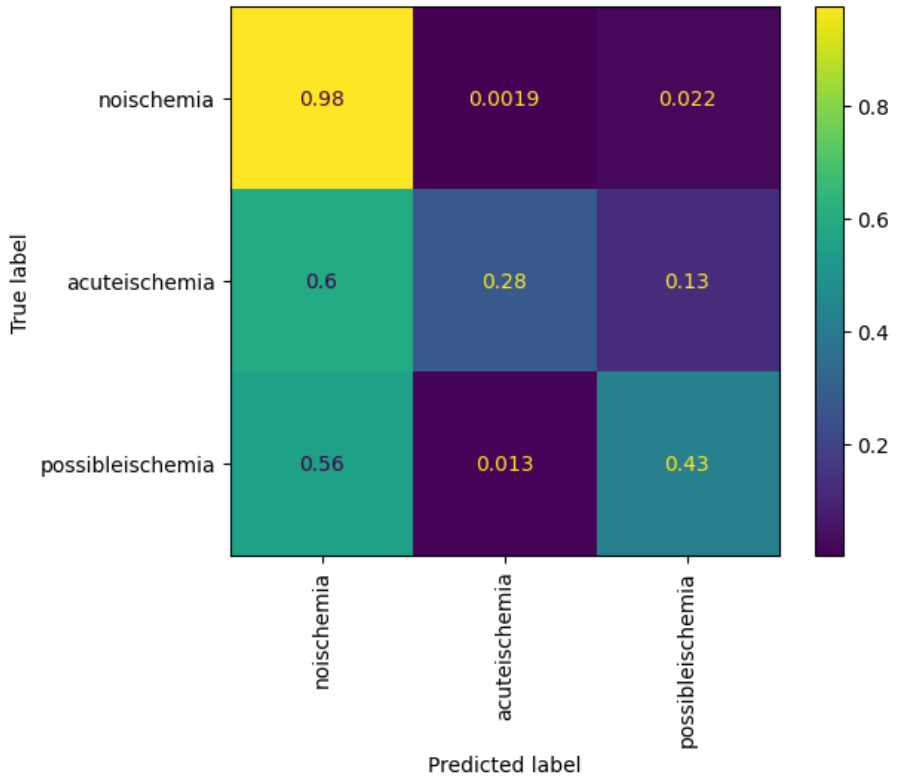



Figure 48: DiagCl ensemble, Ischemia labels: confusion matrix.

| Condition | ROC-AUC |
|---|---|
| oldischemia | 0.8713090894371586 |
| other_lvh | 0.9444848639303637 |
| other_pc | 0.9456560317017403 |
| rhythm_wpw | 0.980714054299229 |
| qrs_pvc | 0.9618043914938221 |
| rhythm_pac | 0.8929069326422576 |

Table 18: DiagCl ensemble, Other labels: ROC-AUC.