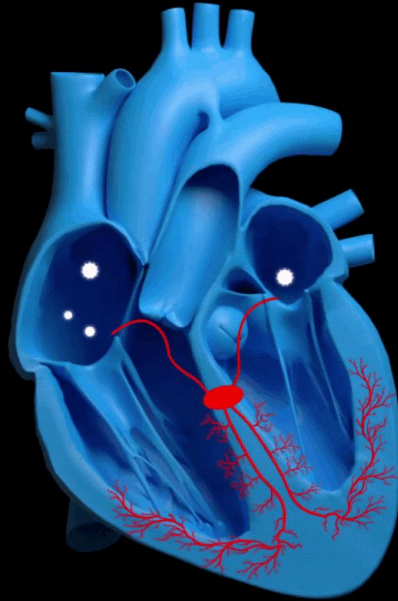


# Atrial Fibrillation detection

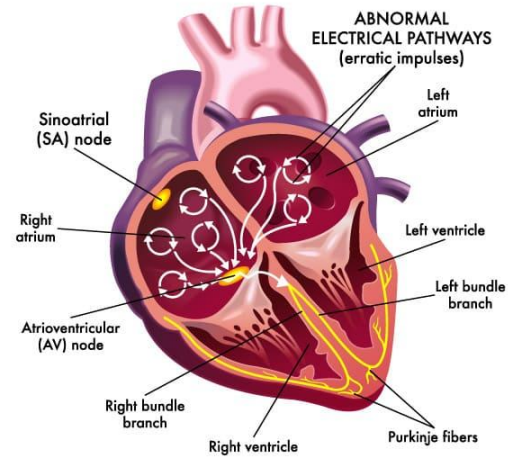
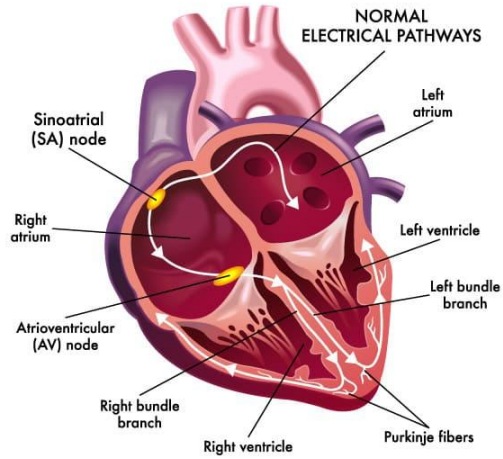
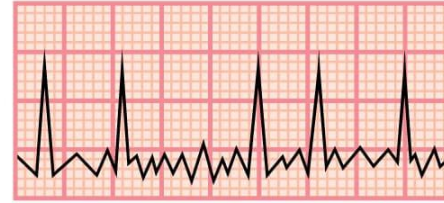


Bohdan Hlovatskyi  
Signal Processing Course Project

Normal ECG



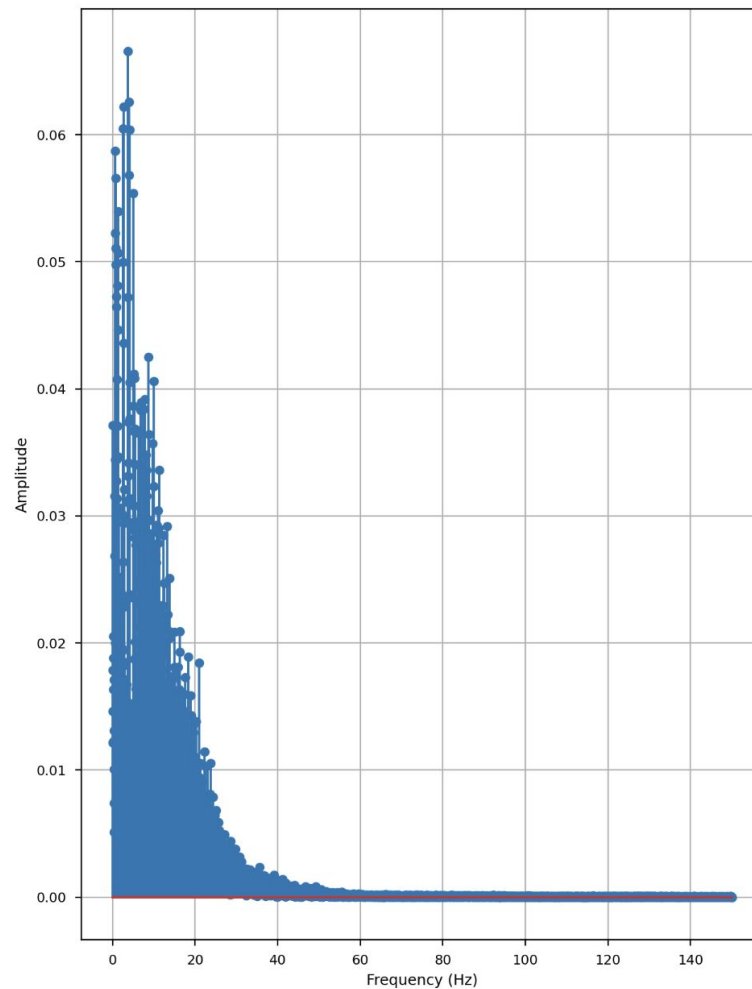
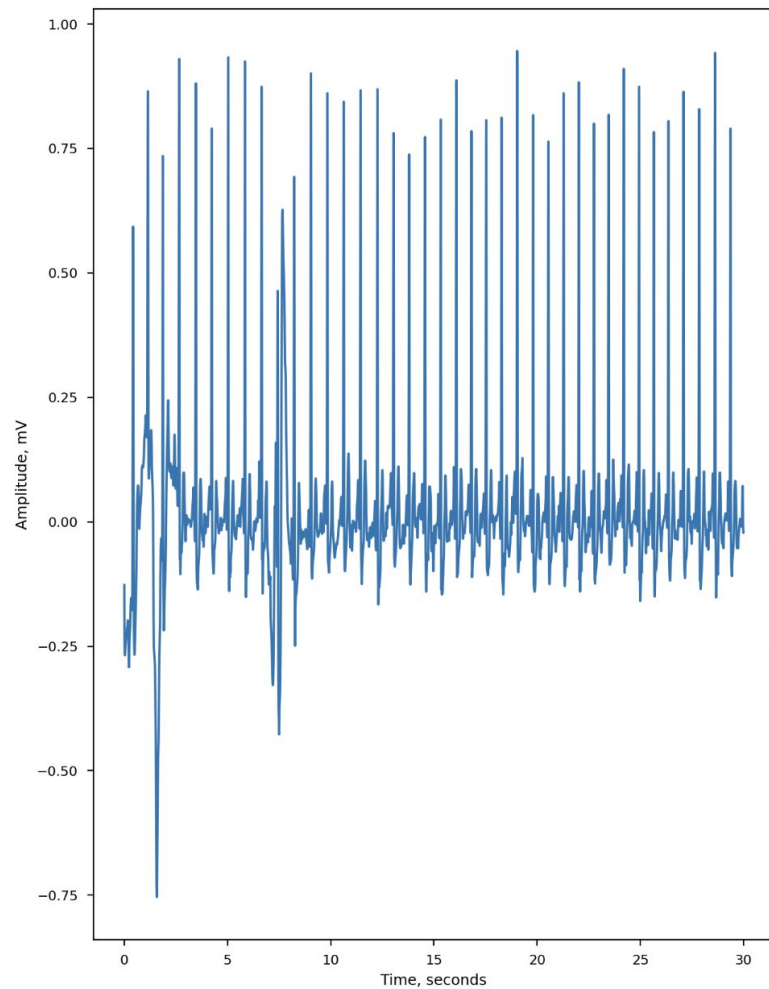
Atrial Fibrillation



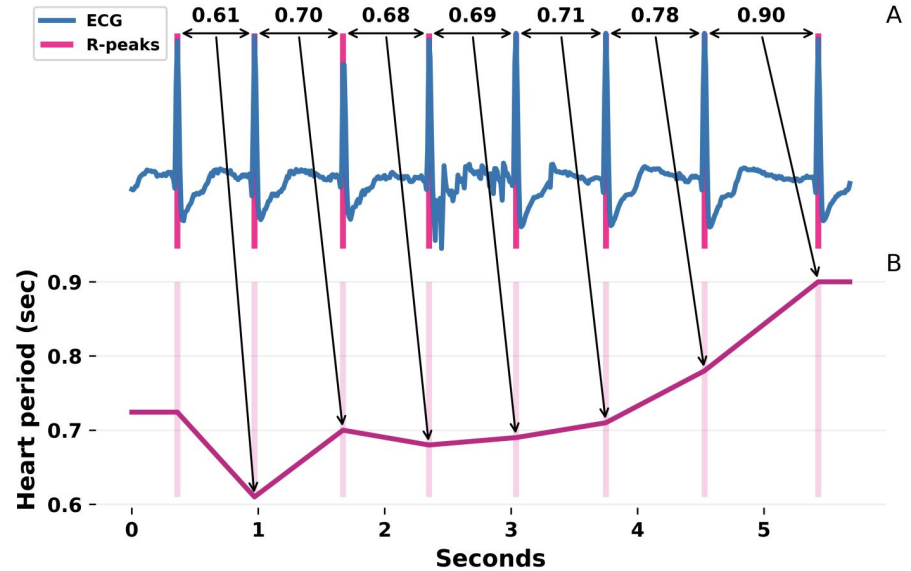
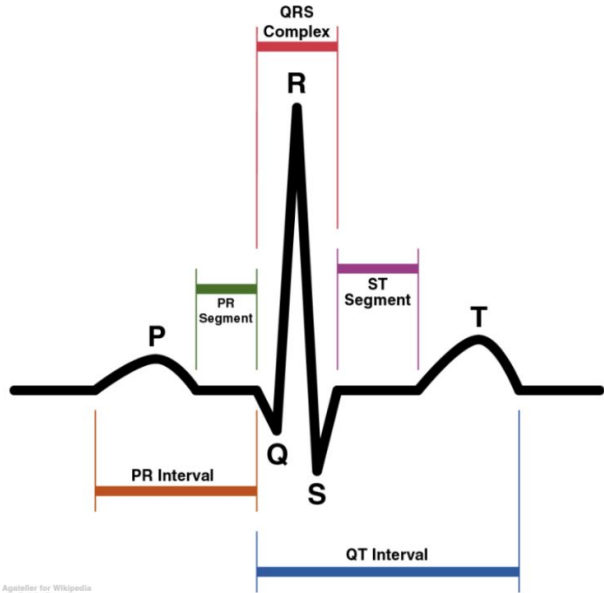
# Data

- AF Classification from a Short Single Lead ECG Recording: The PhysioNet/Computing in Cardiology Challenge 2017
- The training set contains 8,528 single lead ECG recordings lasting from 9 s to just over 60 s
- ECG recordings were sampled as 300 Hz and they have been bandpass filtered by the AliveCor device

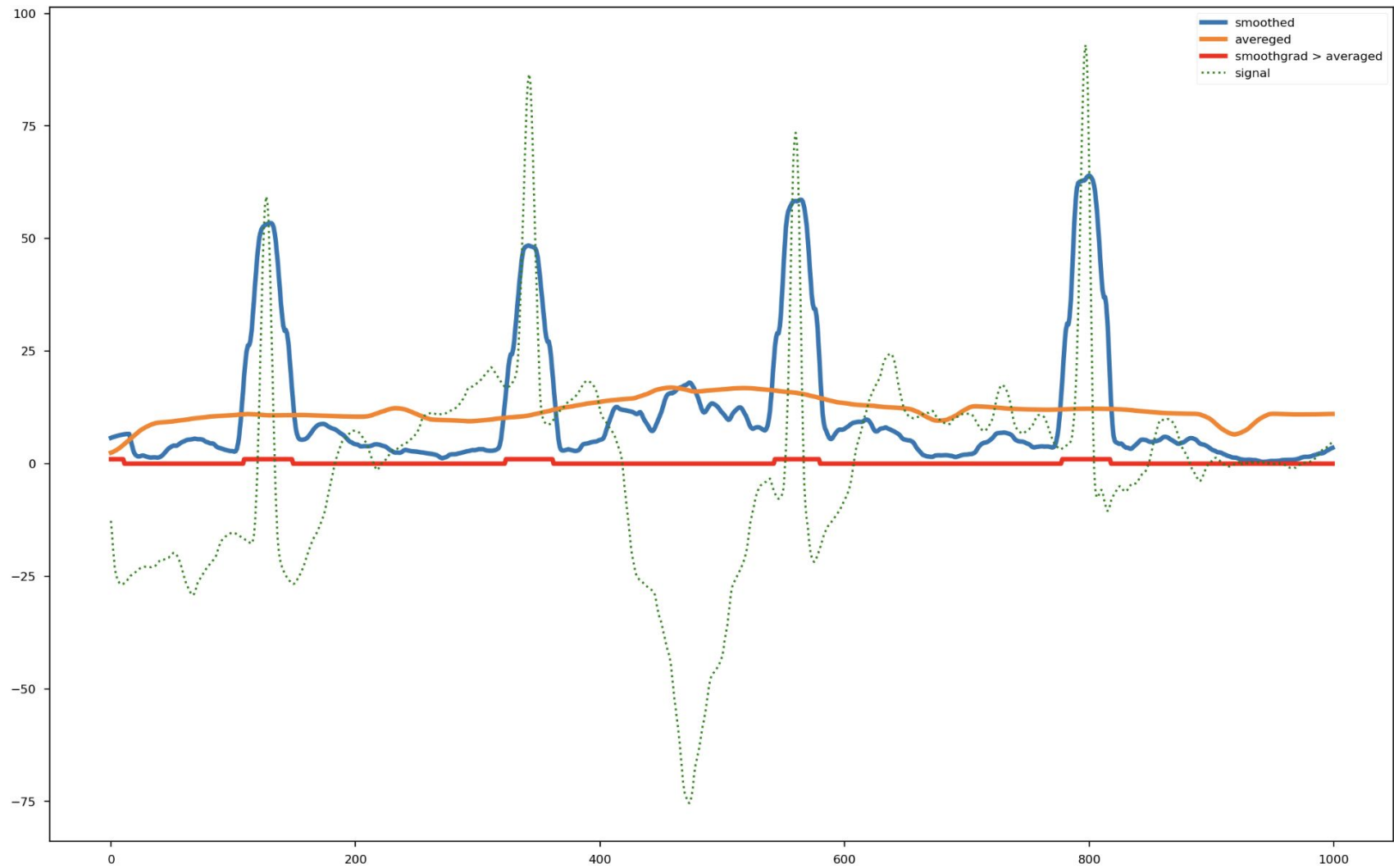
| Type         | # recording | Time length (s) |             |             |           |            |
|--------------|-------------|-----------------|-------------|-------------|-----------|------------|
|              |             | Mean            | SD          | Max         | Median    | Min        |
| Normal       | 5154        | 31.9            | 10.0        | 61.0        | 30        | 9.0        |
| AF           | 771         | 31.6            | 12.5        | 60          | 30        | 10.0       |
| Other rhythm | 2557        | 34.1            | 11.8        | 60.9        | 30        | 9.1        |
| Noisy        | 46          | 27.1            | 9.0         | 60          | 30        | 10.2       |
| <b>Total</b> | <b>8528</b> | <b>32.5</b>     | <b>10.9</b> | <b>61.0</b> | <b>30</b> | <b>9.0</b> |

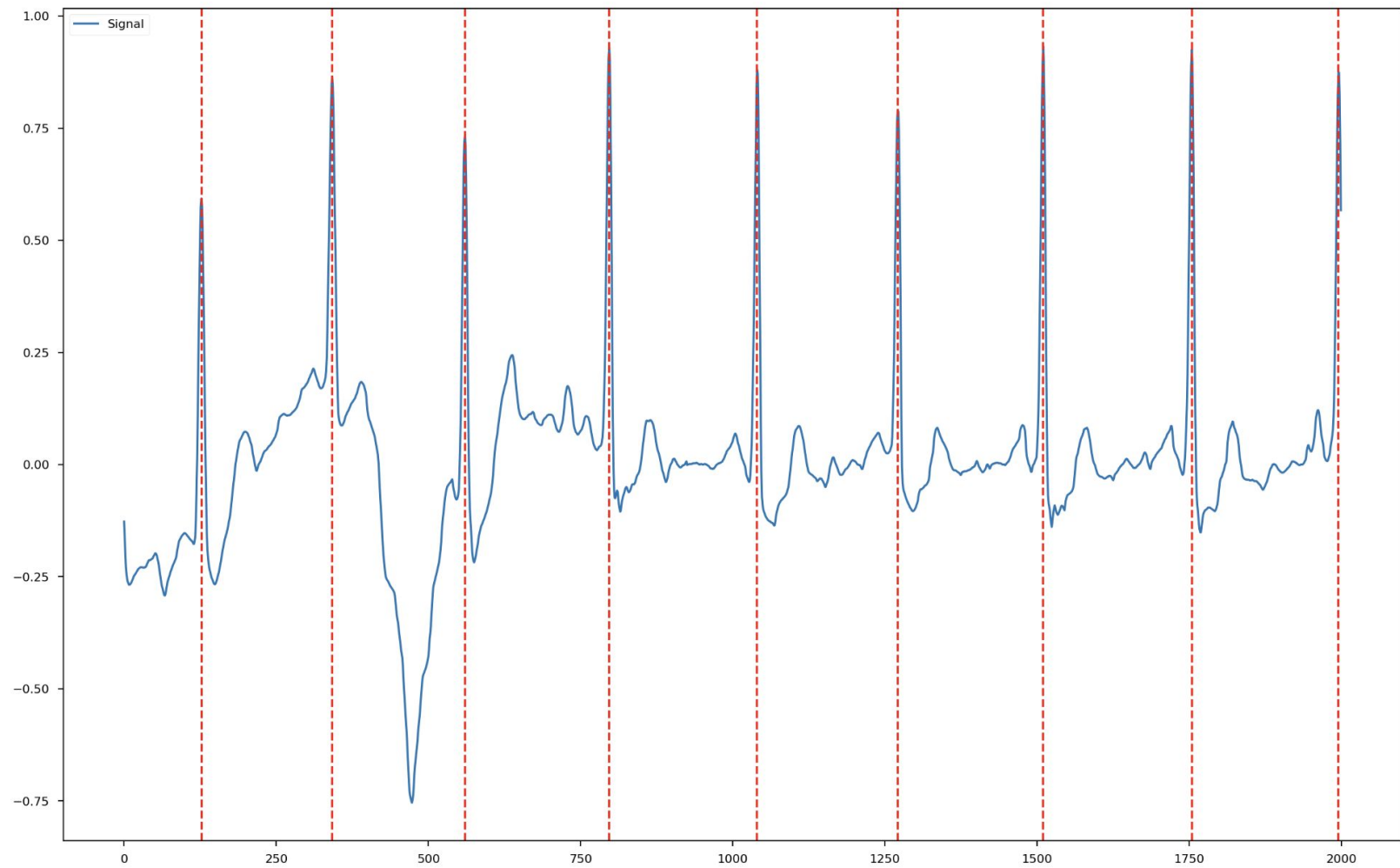


# Preliminaries

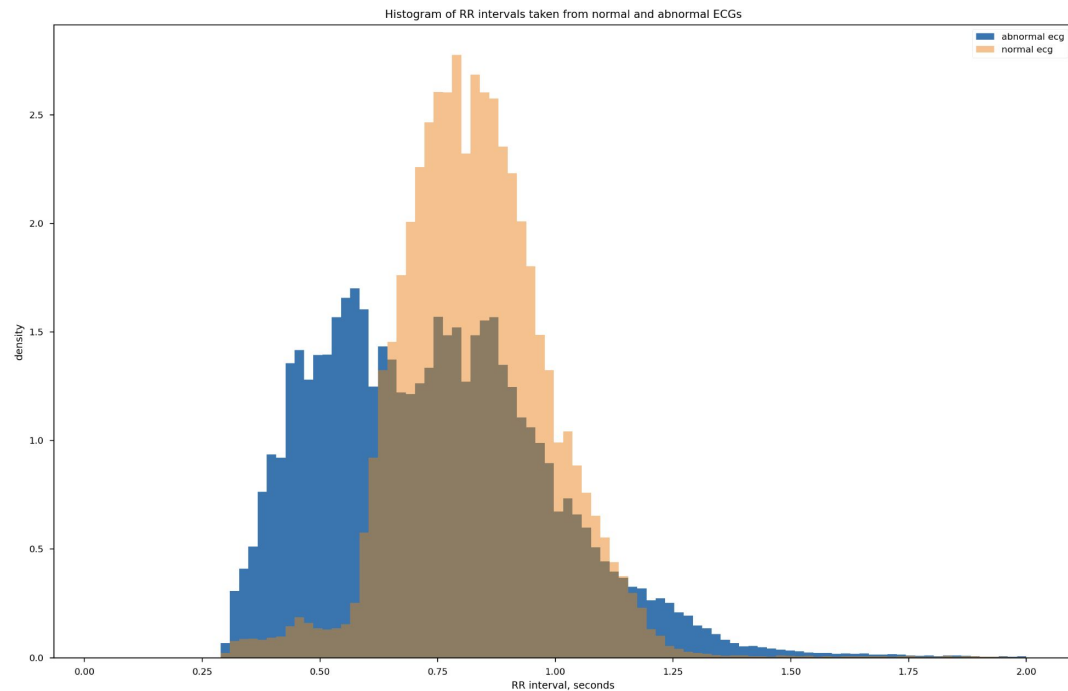


**Figure 1:** Extraction of heart period (panel B) based on R-peaks in an ECG (panel A). Note that this is conceptually identical to the extraction of heart period based on systolic peaks in PPG.





# KS Test



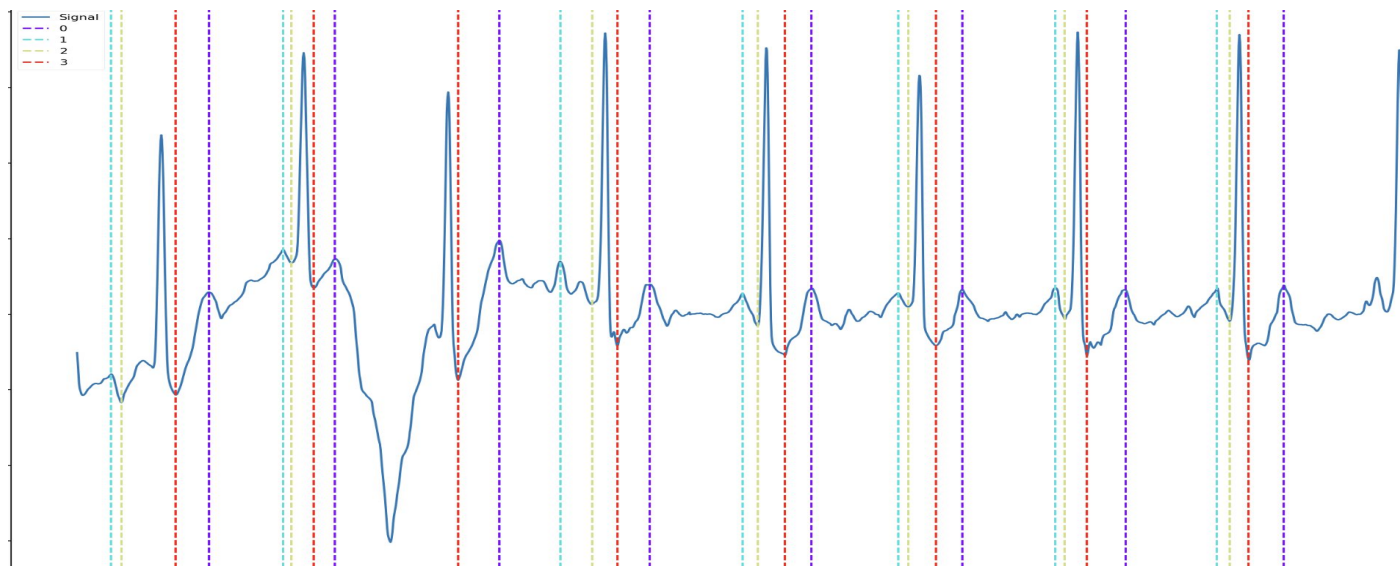
$$D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)|,$$

where  $F_{1,n}$ ,  $F_{2,m}$  are the empirical distribution functions of the first and the second sample respectively



# Machine Learning

- Logistic regression on raw ECG
- Logistic regression with hand-crafted features
- XGBoost with hand-crafted features



## Logistic regression (baseline)

| Classification Report: |           |        |          |         |  |
|------------------------|-----------|--------|----------|---------|--|
|                        | precision | recall | f1-score | support |  |
| 0                      | 0.60      | 0.63   | 0.61     | 1049    |  |
| 1                      | 0.36      | 0.33   | 0.35     | 657     |  |
| accuracy               |           |        | 0.52     | 1706    |  |
| macro avg              | 0.48      | 0.48   | 0.48     | 1706    |  |
| weighted avg           | 0.51      | 0.52   | 0.51     | 1706    |  |

## Logistic regression (add. feat.)

| Classification Report: |              |        |          |         |      |
|------------------------|--------------|--------|----------|---------|------|
|                        | precision    | recall | f1-score | support |      |
|                        | 0            | 0.63   | 0.76     | 0.69    | 1011 |
|                        | 1            | 0.51   | 0.35     | 0.42    | 693  |
|                        | accuracy     |        |          | 0.60    | 1704 |
|                        | macro avg    | 0.57   | 0.56     | 0.56    | 1704 |
|                        | weighted avg | 0.58   | 0.60     | 0.58    | 1704 |

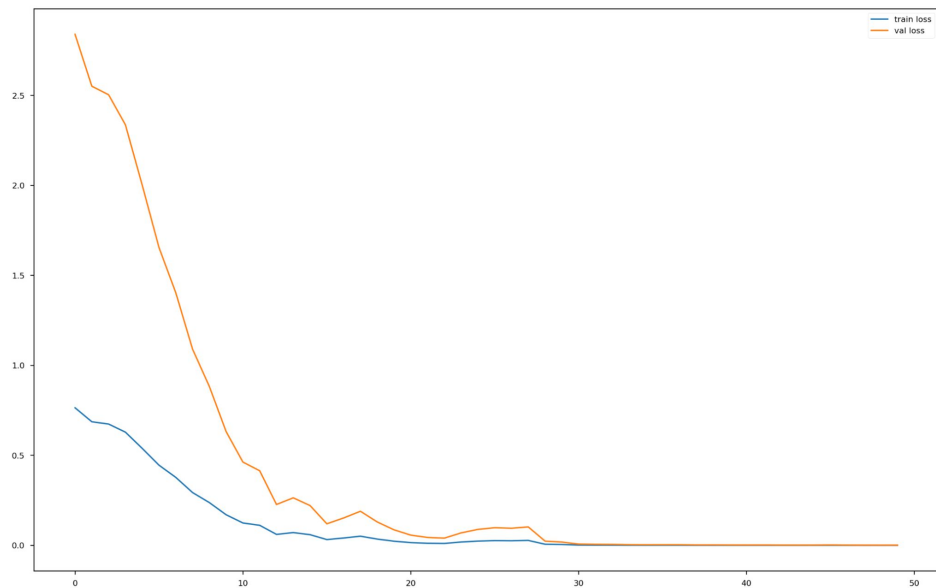
## XGBoost (add. feat.)

| Classification Report: |           |        |          |         |      |
|------------------------|-----------|--------|----------|---------|------|
|                        | precision | recall | f1-score | support |      |
| 0                      | 0.71      | 0.92   | 0.80     | 1011    |      |
| 1                      | 0.80      | 0.45   | 0.58     | 693     |      |
| accuracy               |           |        | 0.73     | 1704    |      |
| macro avg              |           | 0.76   | 0.69     | 0.69    | 1704 |
| weighted avg           |           | 0.75   | 0.73     | 0.71    | 1704 |

# CNNLSTM

---

```
ECGLSTM(  
  (conv1): Conv1d(1, 3, kernel_size=(3,), stride=(3,))  
  (conv2): Conv1d(3, 9, kernel_size=(3,), stride=(2,))  
  (bn1): BatchNorm1d(9, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
  (conv3): Conv1d(9, 9, kernel_size=(3,), stride=(2,))  
  (conv4): Conv1d(9, 27, kernel_size=(3,), stride=(2,))  
  (conv5): Conv1d(27, 27, kernel_size=(3,), stride=(2,))  
  (conv6): Conv1d(27, 81, kernel_size=(3,), stride=(2,))  
  (bn2): BatchNorm1d(81, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
  (lstm): LSTM(81, 256, num_layers=2, batch_first=True, bidirectional=True)  
  (fc1): Linear(in_features=1024, out_features=256, bias=True)  
  (bn3): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
  (fc2): Linear(in_features=256, out_features=2, bias=True)  
  (dropout): Dropout(p=0.3, inplace=False)  
)
```

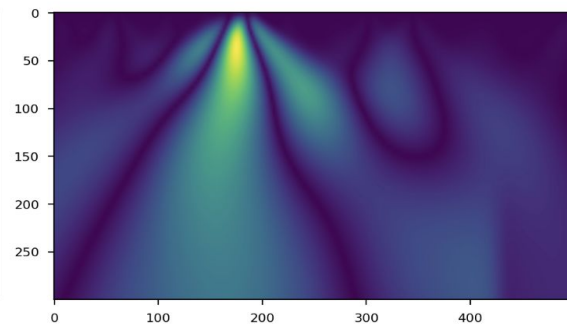
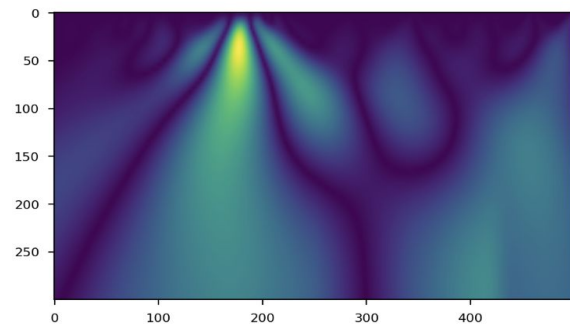
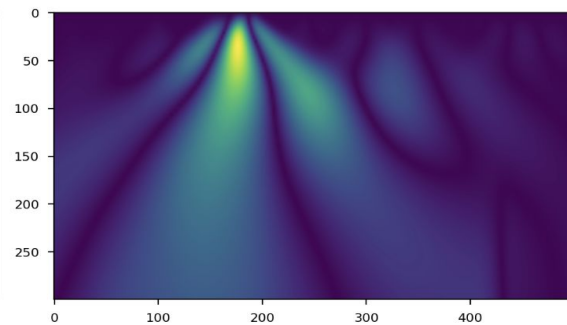
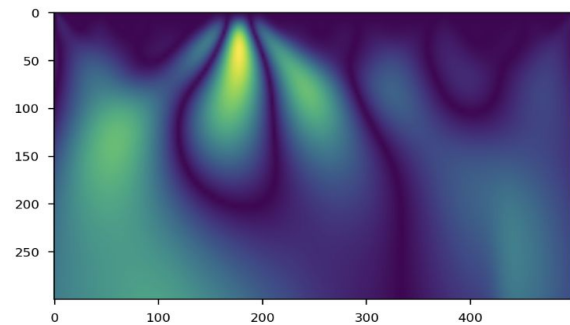


### Classification Report:

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.80      | 0.84   | 0.82     | 1049    |
| 1            | 0.72      | 0.67   | 0.69     | 657     |
| accuracy     |           |        | 0.77     | 1706    |
| macro avg    | 0.76      | 0.75   | 0.75     | 1706    |
| weighted avg | 0.77      | 0.77   | 0.77     | 1706    |

| Method                              | F1 Result   |
|-------------------------------------|-------------|
| KS Test                             | <b>0.6</b>  |
| Logistic Regression                 | 0.48        |
| Logistic Regression with add. feat. | 0.56        |
| XGBoost                             | <b>0.69</b> |
| CNNLSTM                             | <b>0.75</b> |

# DWT



# Conclusion

**Atrial fibrillation** is one of the most common types of heart arrhythmia. It is dangerous as it **increases the risk of a stroke**. Approximately five percent of the population is diagnosed with heart arrhythmia. Several approaches to Atrial Fibrillation detection were implemented and discussed. They were compared on a dataset provided by **PhysioNet in Cardiology Challenge 2017**. While methods with hand-crafted features give good performance, they require a lot of domain knowledge. Deep learning approaches showed to be superior in terms of accuracy

.