CDT-HDS Biomedical Time Series Analysis

Wearable Laboratory

Organiser(s): .... Dr Xiao Gu

Department of Engineering Science

University of Oxford

Nov 2024

Table of contents

1 Lab overview 1
2 Data Loading and Understanding 2
  2.1 Data Description 2
  2.2 Task 1: Visualize One ECG Recording 2
3 Classification Model Development 3
  3.1 Option A – Pipeline of Conventional Machine Learning 3
  3.2 Option B – Pipeline of Deep Learning 4
  3.3 Task 2: Complete Inference/Test Function 4
  3.4 Instructions 4
4 Test on your own recordings 5
  4.1 Digitize ECG recordings 5
  4.2 Task 3 Apply the function on your own recording 5
  4.3 Instructions 5

1 Lab overview

The aim of this laboratory is to gain hands-on experience with developing machine/deep learning models for electrocardiogram (ECG) data collected from wearable sensors. You will learn how to preprocess and analyze ECG signals and perform inference on your own ECG recordings collected from consumer wearable devices.

The lab materials are available online at `https://github.com/AnshThakur/CDT-TimeSeries/tree/main/labs/lab_3/wearable`. You don't need to download the materials to your laptop/PC. Instead, please click the icon Open in Colab to run the whole notebook step by step online.

2 Data Loading and Understanding

2.1 Data Description

The provided ECG data is derived from the PhysioNet 2017 AF Classification Challenge, available at https://physionet.org/content/challenge-2017/1.0.0/. This data is saved in a dictionary format using the `pickle` module. Each key in the dictionary corresponds to a unique recording identifier, and the associated value is another dictionary containing the following:

- `'data'`: The ECG signal data as a flattened array.
- `'label'`: The label for the ECG signal.

The labels in the ECG data indicate the type of rhythm or signal quality:

- `N`: Normal sinus rhythm.
- `A`: Atrial fibrillation.
- `O`: Other rhythm.
- ∼: Noise.

2.2 Task 1: Visualize One ECG Recording

Visualize one sample ECG recording annotated with `A` (arrhythmia). To complete these tasks, refer to Task 1 in the notebook.

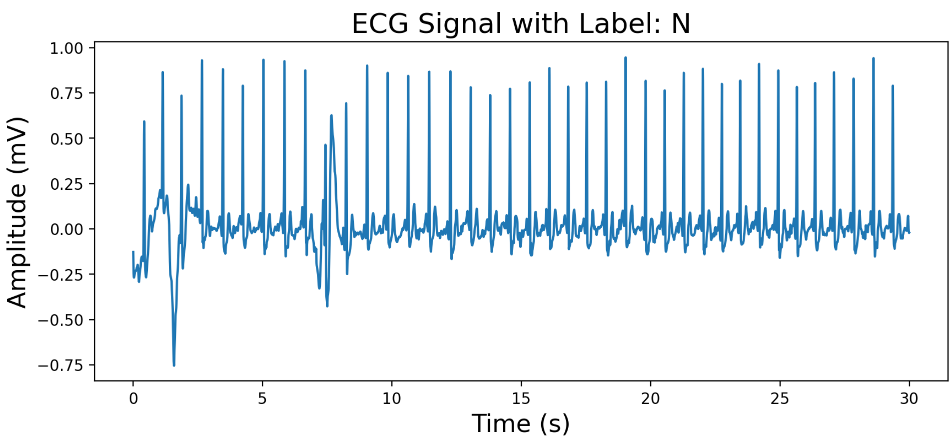

Figure 1: Example of visualization of one ECG recording annotated with `N` (normal)

3 Classification Model Development

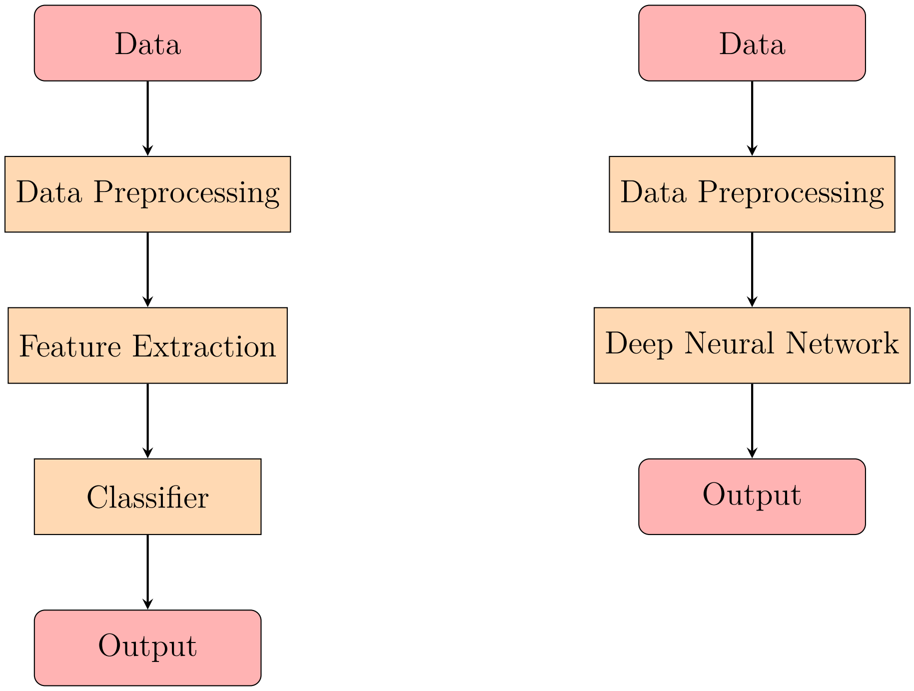

Figure 2: Comparison of Conventional Machine Learning Pipeline and Deep Learning Pipeline for ECG Signals.

3.1 Option A – Pipeline of Conventional Machine Learning

In conventional machine learning, the analysis of biomedical signals would normally involve three stages: data preprocessing, feature extraction, and classifier training. You will go through these three steps in the notebook.

3.1.1 Instructions

- Preprocess the ECG data: Clean the data and extract characteristic waveforms with `neruokit2`.
- Extract features: Use domain-specific methods to extract relevant features (morphological, heart rate variability, signal quality index) from the ECG signals.
- Train a classifier: Train a conventional machine learning classifier, such as RandomForest, using the extracted features.
- Evaluate the model: Assess the performance of the model on validation and test datasets.

Refer to Section 2.2 in the notebook for detailed code and explanations on how to implement these steps.

3.2 Option B – Pipeline of Deep Learning

In this option, you will apply deep learning methods directly to the ECG data for model development.

3.2.1 Instructions

- Preprocess the ECG data: Normalize the data and prepare it for input into a deep learning model.
- Define the deep learning model: Use a convolutional neural network (CNN) or recurrent neural network (RNN) to model the ECG data.
- Train the model: Use appropriate loss functions and optimizers to train the deep learning model.
- Evaluate the model: Assess the performance of the model on validation and test datasets.

3.3 Task 2: Complete Inference/Test Function

Based on the option you choose (Option A or Option B), complete the test function to read an ECG recording and generate the probability of it being a particular class (e.g., normal, atrial fibrillation, etc.).

3.4 Instructions

1. Implement a function to preprocess a new ECG signal (crop or pad to a fixed length for deep learning method).
2. Extract features or preprocess the signal as required by the model (feature extraction for conventional ML, normalization for deep learning).
3. Use the trained model to predict the class of the new ECG signal.
4. Print and interpret the output probabilities.

Refer to Task 2A or Task 2B in the notebook for detailed code and explanations on how to implement these steps.

4 Test on your own recordings

4.1 Digitize ECG recordings

In this section, you will be given one ECG recording collected from an Apple Watch. Use your developed machine learning model to infer whether this belongs to normal or atrial fibrillation (AF).

If you have an Apple Watch and want to use your own recording, please follow the instructions to generate your report from the Health app on your iPhone. More details can be found at https://support.apple.com/en-gb/120278.

4.2 Task 3 Apply the function on your own recording

Your task is to test your developed method on digitized ECG recordings and to generate the probability of whether the recording is normal or indicates AF.

4.3 Instructions

- Preprocess your own ECG recording: Ensure it is in the correct format and sampling rate.
- Apply the inference function you developed in Task 2 to your own ECG recording.
- Interpret the output probabilities to determine the likely class of your ECG recording.

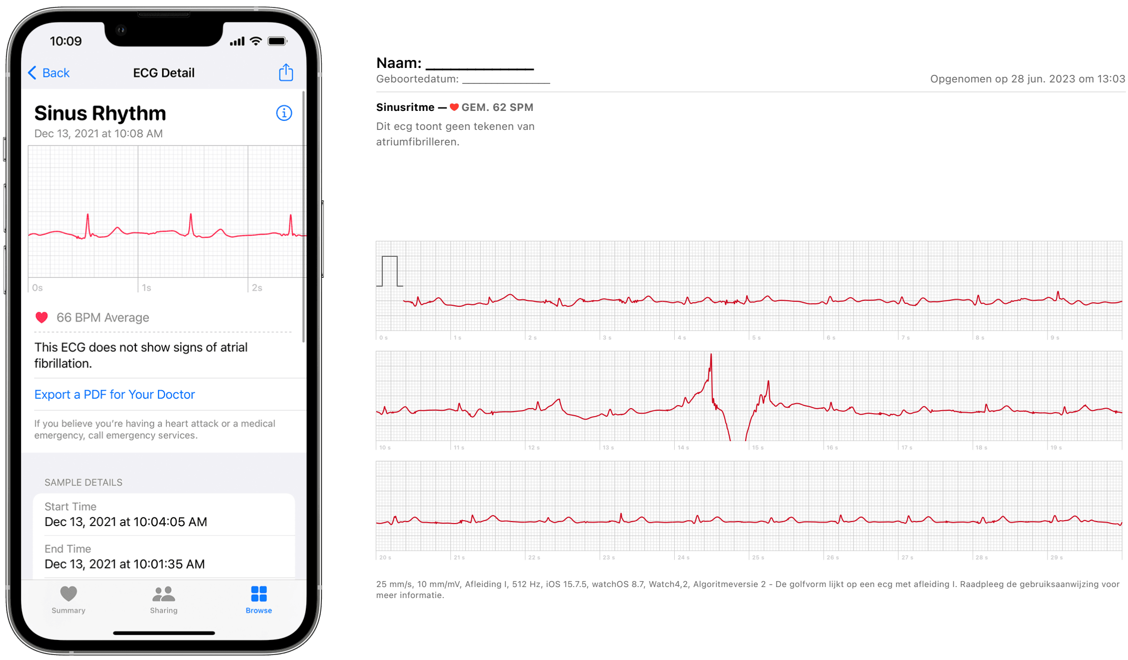

Figure 3: Example of ECG report generated from Apple Watch. Image Source:https://support.apple.com/en-gb/120278