Radar-Based Vital Sensing [Team #5]

Using Machine-Learning

About this document

Scope and purpose

In this document, we provide a brief explanation of our approach for the problem statement presented at the EESTech-2022-Hackathon.

Table of contents

1 Introduction

In the proposed approach, we implement a machine-learning based classifier to distinguish between non-static and quasi-static movements of a subject based on acquired sensor-data. Further, for the quasi-static case, we estimate the vital information like heart rate (HR) and breathing-rate (BR) from the statistical analysis on chunks of the recorded data.

2 Methodology

In our implementation, we performed the following steps.

2.1 Data Acquisition

1. The following values are extracted using the provided pre-processing functions from the raw radar output:
    a. 3 abses
    b. phase
2. Above data is stored as a dictionary in a **.npz** file.
3. The readings are recorded at 3 different distances to get varied data.
4. For efficient gathering and labeling, long sequence of readings was recorded for each class.

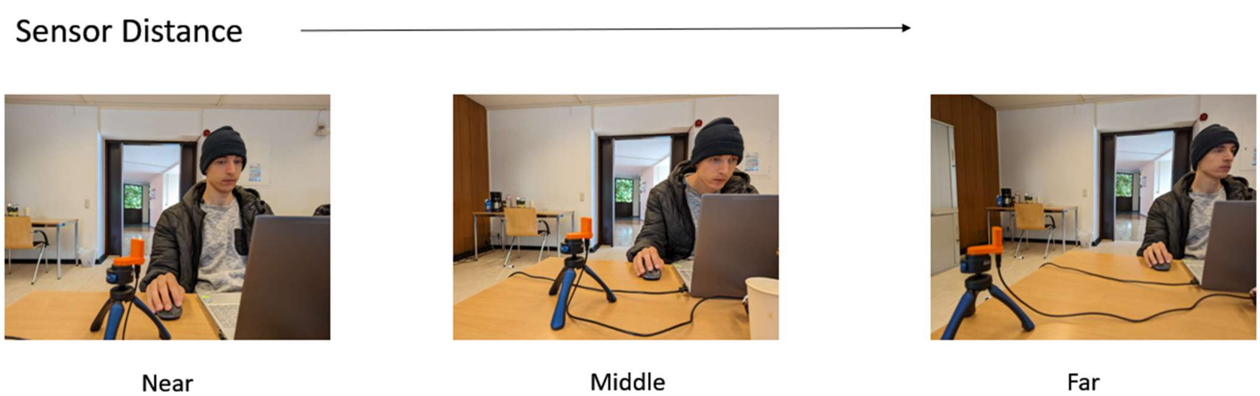

Fig. 1 Recording from various distances

2.2 Data Pre-processing

1. Acquired data was split into manageable chunks of duration 3000 samples.
2. The chunks were labeled before being split.
3. For every chunk, mean was calculated and subtracted from the data-points in the chunk.
4. Each chunk (consisting of 4 time-series (abses[0,1,2] and phases)) is described by a feature vector consisting of 7 attributes: mean, std, min, max, 25% quantile, 50% quantile, 75% quantile.

   (The mean is redundant since we have removed it previously.)
5. This gives us a total of 28 features. These features were used by both movement classifier and activity state classifier.
6. The labelled data (28 features) was then shuffled together to form the dataset.
7. The dataset was then split into training_dataset and testing_dataset (80:20).
8. The features are then fed to the classification model.

2.3 Classification Model

Our model consists of two classifiers:

1. Movement Detection Classifier (Moving v/s Static)
2. Activity State Classifier (Relaxed and Anxious/After-Activity)

Both models are Support Vector Machines (SVM). We have previously trained an XGBoost model which manifested over-fitting, so to get around that problem we chose SVMs for their generalization capabilities. The default hyperparameters in the sklearn library were used to create both classifiers. The linear kernel in SVM gave the best accuracy.

2.4 Model Pipeline

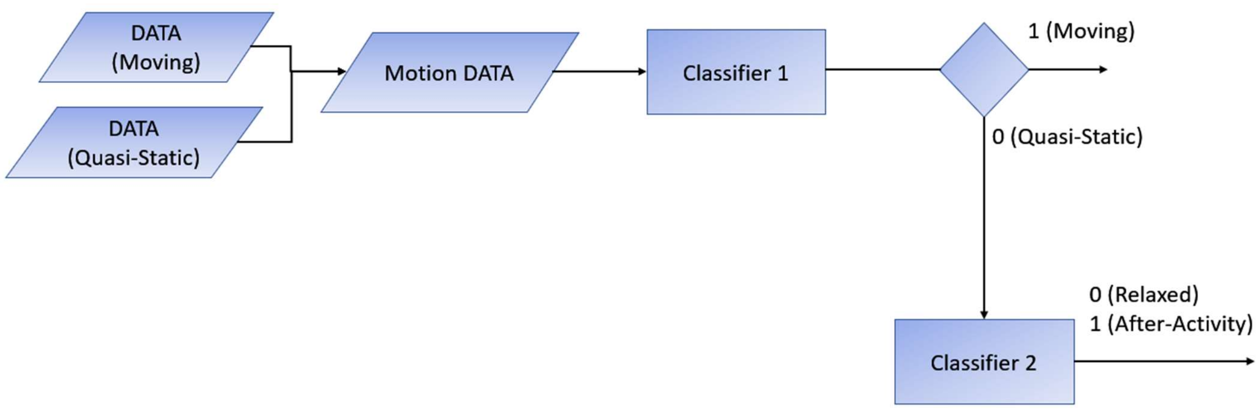

Fig. 2 Model Pipeline

The way we envision the implementation of our algorithm is the following:

1. On an edge device, data from the sensor is continuously streamed into a buffer of length 'N=3000' (chunk length).
2. Every 'K' sampling steps, the 'N' samples in the buffer go through the following (like a sliding window):
   a. Data pre-processing
   b. Classification

3 Results & Discussion

We train our classifiers on the gathered data and achieve 95% validation accuracy on the movement detection task and 89% on the activity detection task. Looking at the results on previously unseen test data we can say that the movement prediction is almost perfect and the activity classification is satisfactory.

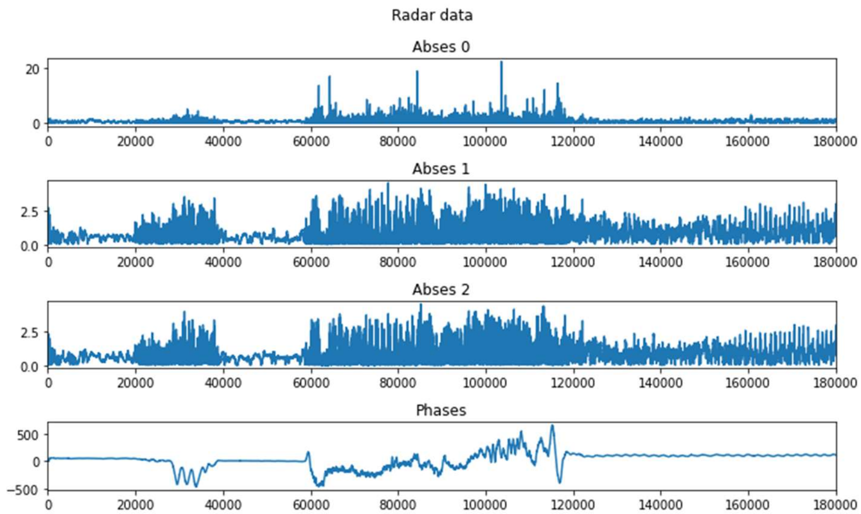

Fig 3.1 Visualization of radar-acquired data

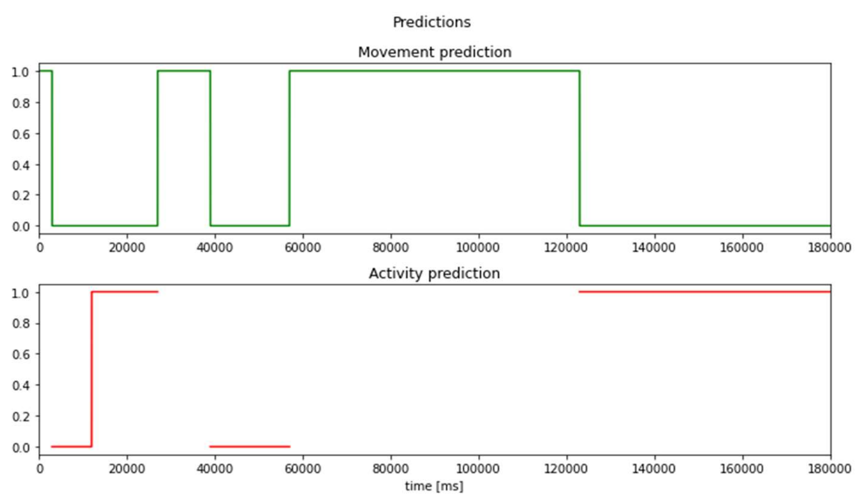

Fig 3.2 Model output for above input data

4 Solution Feasibility

- Both the classification models are small in size.
- The Motion Classifier and Activity-state Classifier are of size 55 KB and 99KB respectively.
- The prediction is really quick and feasible for application on edge devices as our model uses simple linear classifiers which require just a few multiplications and additions, and our features are simple time series stats stats.
- Since SVM is an industry standard, the solution has a lot of community support.