# Introduction

There is a growing need for assisted-living in elderly care due to increased number of elderly people. Statistics show that falling has been a serious risk to the ageing group, which could lead to bone fracture, coma, and even death. Emergent medical attentions are often required after the fall. Considering that many elderly people live alone, it is not easy for seeking immediate help as severe injury or unconsciousness occurs because of a fall.

In recent years there is a rapid growth of interests on such automatic systems for detecting falls and alarm triggering. Many current methods exploit wearable devices with motion sensors, such as accelerometers and gyroscopes, and reasonable results are obtained in fall detection. However elderly people often feel uncomfortable when wearing such kind of devices for a long time, or forget to wear them at times. Thus, our solution to the aforementioned issues consists of product a simple bracelet to detect falls from any other movements and by that it can minimise the accidents.

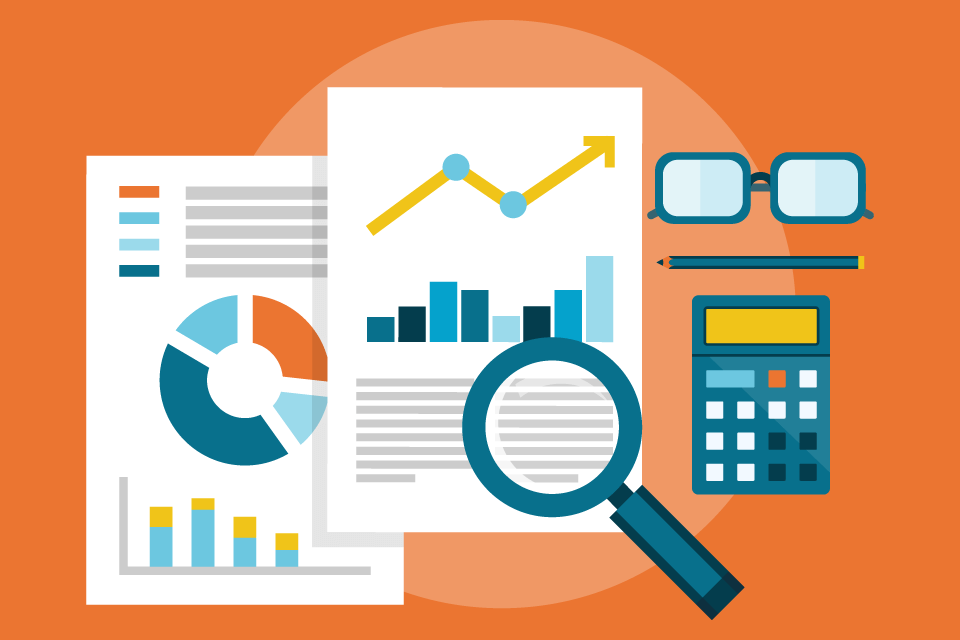
# Context of the Project

## Introduction:

This chapter provides the context and the circumstances in which our project has been carried out. In the first section, we focus on presenting the context of our project. Afterwards, we are going to state the architecture of our project as well as the project tasks and finally the tools which we used during this project.

## General idea of this project:

Thru this project we aim to help elders and young people whom may suffer from some diseases (diabetes, low blood pressure, heart disease …). To better protect themselves, we’re in the process of designing a simple bracelet to detect the falls based on a new concept which is dictionary learning to extract fall characteristics.

Gyroscope

Accelerometer

Sensing Unit Data Pre-processing Fall Detection

Figure 1 : project Architecture

During this project we have two principal tasks:

The first task summarized in Recovery and pre-treatment of pre-recorded signals from the database for the creation of a Dataset: this task involves filtering, segmenting and extracting descriptors. And the second task is to configurate and implement a machine Learning model from the algorithms of Dictionary Learning.

## Development environnement:

For this project, we used two different computer science programming languages, Python for the pre-processing and cleansing phases and for the training phase we used Matlab because the algorithms are available on GitHub with Matlab.

You can install Python thru this website (Matlab is not open source)

1-<https://www.python.org/downloads/>

Conclusion:

Throughout this chapter, we highlight our project’s context. We have presented the general idea of our project and we have clarified the problematic and goals behind this work. In the next chapter, we will explain the pre-processing phase.

# Experimental configuration

## Introduction:

In this section, we’re going to give a clarification about the most important phase which is Data Pre-processing and also, we will discuss Slicing Signals and how far the choice of window size can help us to ameliorate the accuracy during the modelling phase.

## Data pre-processing:

### 2.1 Introduction:

Data pre-processing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we pre-process our data before feeding it into our model.

### 2.2 Slicing Signals:

Machine-learning-based approaches normally use two types of sliding window: a fixed-length non-overlapping sliding window (FNSW) or a fixed-length overlapping sliding window (FOSW) in order to split the data stream into several segments. Then features are extracted from these segments and used to train and test the classifier.

Selecting a correct size and type of the sliding window is critical to produce differentiable feature values, where those feature values can optimize the detection rate by increasing the classifier’s accuracy.

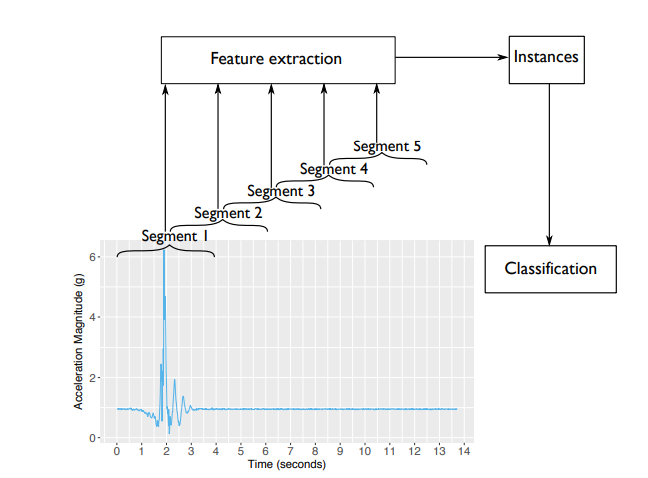


Figure 2 : FOSW-based machine-learning approach for fall detection.

In our case we used the (FOSW) fixed-length overlapping sliding window with window overlap of 50%

And we varied the window size multiple times and we found that the optimal one to our data is 4s.

The first figure shows an example of Non-Fall slicing data of <TA: total acceleration > with window size=4s and the second one shows a Fall slicing window for <TA> also with window size=4s.

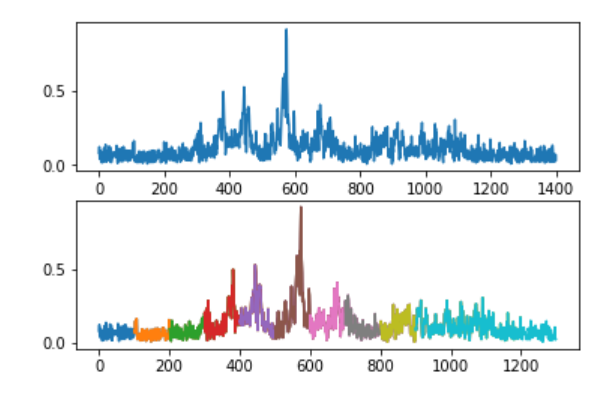


Figure 3 : Before and after slicing Non fall data (window size=4s)

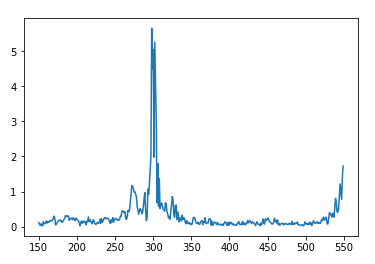
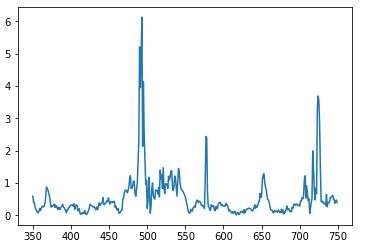


Figure 4 : After Fall slicing data (with window size=4s)

Conclusion:

This chapter was devoted, in the first part, for the description of the Data pre-processing phase which is used to achieve the goals of our application. In the second part, we detailed the effect of the window size and we gave some examples.