# split.py

The purpose of this class is to provide a convenient object that can store all the relevant information contained in the data frame, such as subject name, time stamps, label, etc. It was designed specifically to deal with time series data.

#### How to Declare:

np\_example = TimeSeriesNP(time\_steps, steps)

* time\_steps will determine how many rows of the data frame will stored in each numpy-array element
* steps will determine how steps forward are taken after each element is generated. This parameter controls the degree of overlap between each element.
* *steps = time\_steps* there is no overlap.
* *steps > time\_steps* there is no overlap and there is a small gap of data skipped between each array element.
* *steps < time\_steps there is overlapping data included in each sample.*

#### Functions:

**timeSlice(dataframe)**

* Intake a dataframe that is **date-time indexed** and steps through the data and converts it into separate numpy arrays for storing the sensor data, labels, subjects, and time stamps. Samples that do not have the same labels or subjects are thrown out.

**setArrays(dateframe, encode = True, one\_hot\_encode = True)**

* Primary function for setting the arrays equal to the contents of a dataframe. The encode variable will convert the labels from text values to numbers. Setting encode = True will also create an object variable called **mapping** which is a dictionary that can be used to convert the encoded labels back to their text form. One hot encode just converts the labels to a one hot encoded format.

**updateArrays(dataframe)**

* Intake another dataframe and stack the contents on top of the existing arrays. *This function needs to be tested out more as it wasn’t used in the development process. This kind of function could be useful for processing data sets that have gaps in them, or at least a similar style of function could be useful.*

**trainTestSplit\_subj(split\_subj)**

* Intake a dictionary of the following form: *split\_subj = dict(train\_subj = [1,2,3], validation\_subj = [4,5,6], test\_subj = [7,8,9].* This will split the numpy arrays into train, test, and validation data. This function has not been tested enough, but it is based on previously written code from the IMICS Lab repo where a deep learning model is made from E4 data.