Bed Exit Prediction

BSc Thesis

Alexandra Verhagen & Emilie Jong

Bachelor Graduation Project
December 11th 2020

TU Delft

Bed Exit Prediction

BSc Thesis

by

Alexandra Verhagen & Emilie Jong

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended on Thursday December 17th, 2020 at 10:00 AM.

Student number:	4475593 (Alexandra Verhagen)	
Student number:	4541960 (Emilie Jong)	
Project duration:	October 5, 2020 – December 17, 2020	
Thesis committee:	Prof. dr. ir. G.J.T. Leus,	TU Delft, supervisor
	Dr. ing. I.E. Lager,	TU Delft, BAP coordinator
	M. Zijlmans,	Momo Medical, supervisor

TU Delft

Abstract

This report presents a bed exit prediction model using a decision tree classifier, a machine learning method. This system is designed to alarm a caregiver in a nursing home to prevent bed-falls among institutionalised elderly. For the input of this algorithm, features are derived from four force sensing resistor sensors and six piezoelectric sensors mounted on a device which is placed under the mattress at chest level. The classifier is based on data containing 98 bed exits of two different people on two different mattresses. The best results were obtained for the feature which uses the standard deviation of the force sensing resistor sensors of the last 10 seconds. The classifier was evaluated using a test set which contained a total of 645 time frames of 5 seconds of 23 cases where someone leaves the bed and 622 time frames where there is no bed exit. 10 windows with bed exits were classified correctly out of 23. Moreover, the system has a false alarm rate of 3.9%, meaning that in 24 time windows a bed exit is predicted while there is no actual bed exit. The accuracy with which the model classifies a bed exit or no bed exit correctly 5 seconds beforehand is 94.3%. Thereafter, it was researched whether adding an extra sensor plate would be helpful to improve the bed exit prediction. The extra plate should be placed on the side of the bed if the preferred side for getting up is known, or at the hip level if the side is unknown.This is of added value to define the bed status of the client as this plate, in contradiction to the sensor plate at chest level, still measures pressure when someone sits up straight.

Preface

The aim of this thesis is to explain the design process of creating a decision algorithm to be used in an early warning system to prevent bed falls. The thesis is written for the Bachelor Graduation Project of Electrical Engineering at the Delft University of Technology. With a group of 6 people we worked on a common goal, which was to create algorithms that detect the heart rate & the respiratory rate of a client and predict whether a client intends to leave the bed. These tasks were reached in subgroups of two people and the goal of our subgroup was to predict a bed exit by using machine learning.

Due to the COVID-19 situation, we were not able to work together on campus every day. However, the teamwork between all subgroups still went well and the daily online meetings kept everyone updated on the reached tasks and the struggles that were being faced.

We would like to express our sincere gratitude to our supervisor Prof.dr.ir. Geert Leus for his guidance and continuous support throughout the project. His extensive knowledge and fast correspondence helped us when we encountered problems. We would also like to thank dr. Ioan Lager for making the graduation project possible amid the current situation. Furthermore, we would like to thank our project proposer Momo Medical and especially Thomas Bakker & Maarten Zijlmans for their weekly meetings and cooperation.

Alexandra Verhagen & Emilie Jong
Delft, December 2020

Contents

1 Introduction

The numbers of elderly have been rising rapidly for the past few years. Is is expected that in 2040 there will be 1,600,000 people over 80 years old living in the Netherlands [4]. On January 1st of 2019 over 115,000 people lived in a nursing home [5]. Due to this ageing society it is estimated that more and more people will spend their last years in a nursing home in the coming years. This will inevitably lead to an increase of the workload for caregivers. In specific, the increasing amount of falling incidents are a problem in nursing homes. They can lead to serious injuries and thus have a huge impact on the health of the patient, such as his or her mobility. Especially institutionalised patients who suffer from diseases such as dementia, are more at risk of falls. This is due to the fact that they have problems with the declination of balance and gait, muscle weakness and cognitive impairment which results in poor risk assessment of their surroundings [10] [28].

1.1. Problem Definition

Due to the lack of nursing staff and resources in an ageing society, institutionalised patients are more likely to wander around and get out of bed without any assistance. Most falling incidents occur near the bed or within the bedroom [9]. Research indicates that bed exit alarms could be a solution. These alarms can detect movements of the patient in bed and alarm the nursing staff when someone is about to leave their bed or has already left [1]. An alarm may already be too late for the caregiver to react on the bed exit, thus, detecting the bed exit before the patient is out of bed is of extreme importance in order to prevent a fall. Therefore, there is a need for a system that can predict a bed exit. The goal is to create an algorithm that can accurately predict a bed exit before a patient is about to leave the bed. Different machine learning approaches can be used for detection of human posture behaviour. Firstly, the data needs to be analysed before determining which input features are relevant. Ideally, the algorithm should be able to run realtime in institutions and use signals that are continuously being sent to the server. This report will focus on the development of this algorithm and the design choices made.

1.2. 'State of the art' analysis

To prevent falling incidents and pressure injuries in healthcare, multiple bed exit detection and bed exit prediction systems have already been designed in the past few years. These research papers show that it is possible to detect a bed exit by using for example pressure sensors. Patients who are about to exit their bed, typically sit up and turn on one side before leaving the bed. The pressure levels are indicators of whether the patient is sitting and getting up. By distinguishing extremely low and high pressure levels of the system's sensors, it can determine different states and concludes if a patient is about to leave the bed. Hence, it can send an alarm when it detects a bed exit [7].

Research also reveals that a bed exit can be predicted using accelerometer data [13]. An accelerometer can measure the angle of the patient bed and thus indicate the position of a patient. In specific, such a system has been applied to hospital beds [24]. Another system that has been designed, detects the behaviour before getting out of bed via an air pressure- and an ultrasonic sensor. The ultrasonic oscillosensor was developed to obtain low-frequency vibrations of the patient and the air pressure sensor detects movements by changes in pressure [27].

A bed exit could also be detected by using piezoelectric sensors [19]. Piezoelectrics are preferable when a fast response is needed. Those are extremely sensitive and however, more fragile as well. A piezoelectric sensor uses the piezoelectric effect of piezoelectric materials to measure vibrations and can be used to measure acceleration and force [17]. Through the piezoelectric sensor heart and respiration rates can be obtained contact-free, for example by placing the sensor under a mattress. In another study more research is conducted on how vital signs can be obtained from piezoelectric sensors. In that study it is also explained how these vital signs could also be incorporated in the bed exit detection [21]. It is suggested to implement a vital signs monitoring function into the bed-leaving detection system to improve the bed exit detection accuracy. Piezoceramic bed sensing devices could be used for establishing the body movement, position change and scratching motion of a patient. This extra information provides insights in the behaviour of someone who is about to leave the bed [22].

Several researches have also been done for bed exit systems using cameras. One of these systems uses image processing algorithms to detect when the patient sits on the edge of the bed, by analysing camera images. This requires the consent of the client, otherwise the system will not be installed [15]. Another model trained with convolutional neural networks, analyses movements of the upper body and captures images with depth cameras for its feature extraction. Because the system uses low-resolution depth images, there were no privacy concerns [6].

In another study [14] different bed exit detection and prediction systems are reviewed as well. This review also described other options for detection and prediction systems such as infrared sensors, temperature sensors and humidity sensors. It is concluded that a bed exit is difficult to detect from the change of temperature and amount of water vapor directly, because the temperature and amount of water vapor change due to the change of environmental conditions [25]. An infrared sensor on the center of headboard could be used to detect the elderly when he or she is sitting around the center of the bed. This method is used in combination with pressure sensors [18].

Three categories are distinguished between bed-fall alarm systems: wearable systems sensors, non-wearable systems sensors and fusion systems sensors. Non-wearable systems, in particular the vision-based bed-fall alarm systems, are more sensitive to privacy concerns. On the other hand, wearable systems tend to be intrusive, which is less desirable as the monitored person is not always cooperative [2]. Fusion systems sensors are a fusion of wearable and non-wearable systems. The cost of a bed-fall alarm system is related to the type of sensors that are used. Therefore, for this research non-intrusive techniques with low-cost sensors are preferable, which take the privacy of the user into account [14]. Furthermore, for the design of the bed exit prediction system, only simple machine learning methods, such as a Decision Tree and a Support Vector Machine, are considered because they have limited training time and are more comprehensible.

1.3. The BedSense system

The device which is used for this research is the sensor plate developed by the start-up Momo Medical: the 'BedSense' system. The BedSense system monitors the activity of patients through eight force sensing resistors (FSRs), six piezoelectric (PE) sensors and an accelerometer. The device is placed below the mattress, around the chest area. It provides information about the movements of the patient in bed without having to check on him directly. Via the device the nurse can obtain the needed information, which is shown in an app. For instance, nurses are alerted when a patient leaves the bed. Momo Medical has already developed an algorithm using Python to detect whether a patient is in bed. The algorithm for detecting when a patient leaves the bed is based on when a patient sits in bed. In 76,6% of the detected cases when a patient sits on the bed, the patient gets out of bed. Thus, when a bed exit alarm is sent when a patient sits on the bed, it is in 76,6% of the cases true. In 23,4% of the cases, the patient does not leave the bed and just sits on the bed then lies down again. Only when the caregiver is alerted on time, a falling incident can be prevented. In Figure 1.1 the BedSense device is shown. The total composition of the plate with sensors can be seen in Figure 1.2. In the currently used model, the sensor plate is connected to a cube on the wall. This is the control unit (CU) that sends data to the server. In the newest model that Momo Medical has developed, which is used in this project, this cube has been integrated within the sensor plate so the device only consists of the plate which is also the CU. Moreover, the plate has four FSR sensors instead of eight.

Figure 1.1: The BedSense device [20]. Provided by Momo Medical.

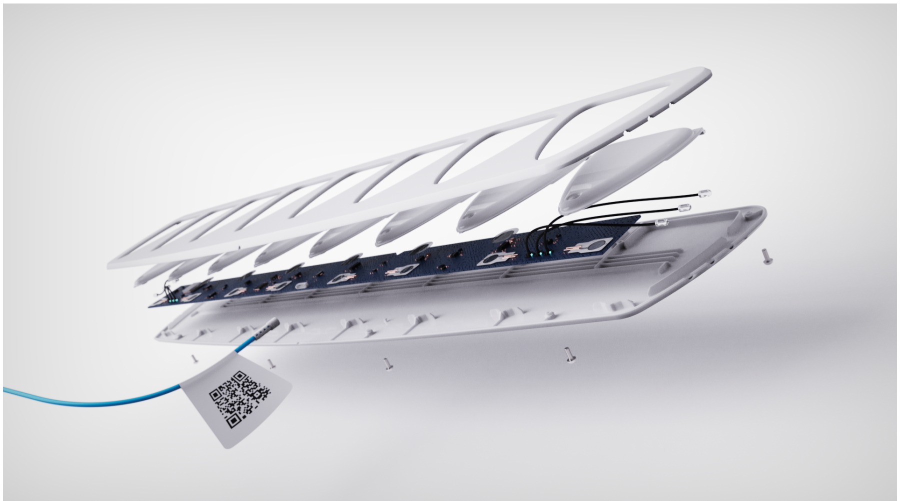


Figure 1.2: A thorough view of the sensor plate of the BedSense (Model 9) [20]. Provided by Momo Medical.

1.4. Thesis synopsis

The process of the design is described in the following chapters. Chapter 2 describes the design requirements to meet the goal. Chapter 3 and Chapter 4 provide an explanation about the analysed data and the feature extraction. These features are used as input for the prediction models, which are described in Chapter 5. In this chapter a machine learning model and a non machine learning model are compared. In addition, Chapter 6 covers the testing that has been performed in order to conduct more research. Thereafter, the discussion of the development process is covered in Chapter 7 and finally Chapter 8 contains the conclusion, the recommendations and the opportunities for future development.

2

The Design Requirements

This chapter discusses the mandatory design requirements and additional design requirements. As stated in the Introduction, the goal is to design an algorithm that predicts a bed exit. Below the design requirements are more specified in detail and prioritised in three categories.

Must-have requirements

Firstly, must-have or the essential requirements are described. These are necessary to provide a reliable and useful algorithm for predicting a bed exit. The following requirements must be satisfied in order to meet the goal:

- The algorithm should predict a bed exit before someone leaves the bed.
- The bed exit prediction should work for labeled data.
- The algorithm should be based on features that are obtained from signals of the device.

Should-have requirements

Secondly, some should-have requirements are summed up which are not crucial for achieving the goal, however, they would be of added value if possible. The should-have requirements are:

- The algorithm should predict a bed exit at least 5 seconds before someone leaves the bed.
- The algorithm should correctly predict at least 35% of the windows where a bed exit occurs in the test set, as a bed exit.
- The algorithm predicts a bed exit in a window where no actual bed exit occurs, for at most 5% of the total windows in the test set where there is no bed exit.
- The algorithm needs to be created in Python.
- The bed exit prediction should work when placed under different mattresses.
- The bed exit prediction should work for people with different body types.

Could-have requirements

Furthermore, the following requirements would be nice to have in case there is enough time available:

- The bed exit prediction should work for unlabeled client data.
- The improved heart rate detection and respiratory rate detection algorithms should be used as input for the algorithm.
- The algorithm sends a signal to alert the caregiver.

3

Data description

Before an algorithm is written, the data set that is obtained from Momo Medical is visualised in order to get insight in the sensors in relation to a bed exit. Furthermore, for the algorithm it is necessary to modify the data so that a bed exit is interpretable for a computer. Finally, the self-made reference data is discussed.

3.1. Data set

Different data sets were made accessible by Momo Medical to comprehend the data from the sensors. The first data set was collected from various patients in healthcare institutions. The second available data set only contained sensor data from patients that are at higher risk of falling. However, these two data sets were unlabeled due to privacy reasons thus did not contain much information on the exact movements of the patient. The third data set, or the reference data, has been created by Momo Medical by using a known protocol. Therefore, further analysis with the reference data is done to recognise certain patterns and understand the working of the sensors in certain situations.

For the reference data, the raw data is visualised and the output signals of the three sensors are observed. Figure 3.1 shows the four FSR and six PE sensor signals during the first part of the reference data. The bed angle which is constructed by the signal of the accelerometer did not show a consistent correlation with a bed exit and is therefore not taken into further consideration for predicting a bed exit. Pd_reference is a manually generated signal by pressing a button when someone is in or out of bed. The signal depicts whether someone is in bed (1) or out of bed (0).

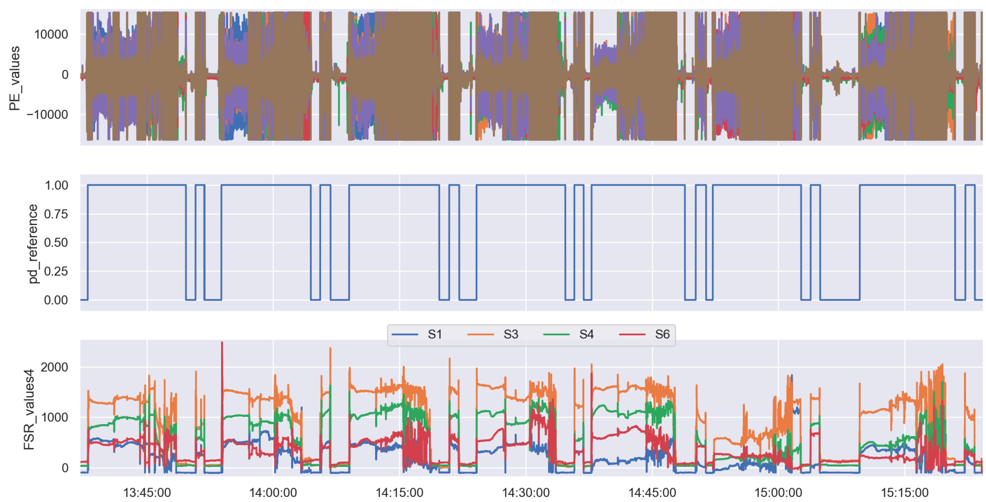


Figure 3.1: A visualisation of the first half of the reference data.

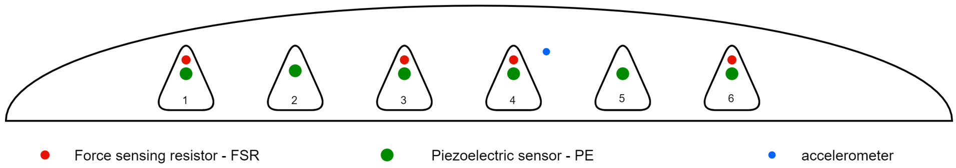


Figure 3.2: The placement of the sensors on the BedSense

The latest version of the BedSense device (model 10), which is used in this report, is depicted in Figure 3.2. The order of the FSR sensors on the device from left to right is 1, 3, 4 and 6.

3.2. Bed exit signal

The total data set is split into windows of 5 seconds, because for the bed exit prediction the model should look at patterns during 5 seconds, as the prediction should be done 5 seconds before the bed exit happens as stated in Chapter 2. The values are analysed in shorter time fragments which is useful for feature extraction in the following chapter. Moreover, using these windows a bed exit signal is generated to determine whether a bed exit happens in each window. Knowing the status of the person in or out of bed, the code checks at the beginning and at the end of the window what the status is. Only when the signal goes from 1 to 0 during a window a bed exit needs to be detected. The working of this code is visualised in Figure 3.3. After determining which windows contain a bed exit, this bed exit signal needs to correspond with the previous window as the model needs to predict a bed exit beforehand. The bed exit signal is thus shifted with one window, as shown in Figure 3.4.

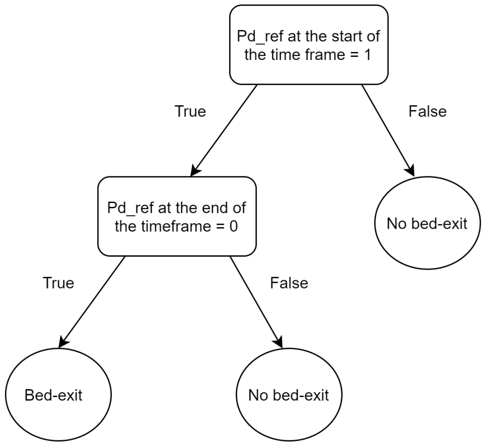


Figure 3.3: The detection of a bed exit

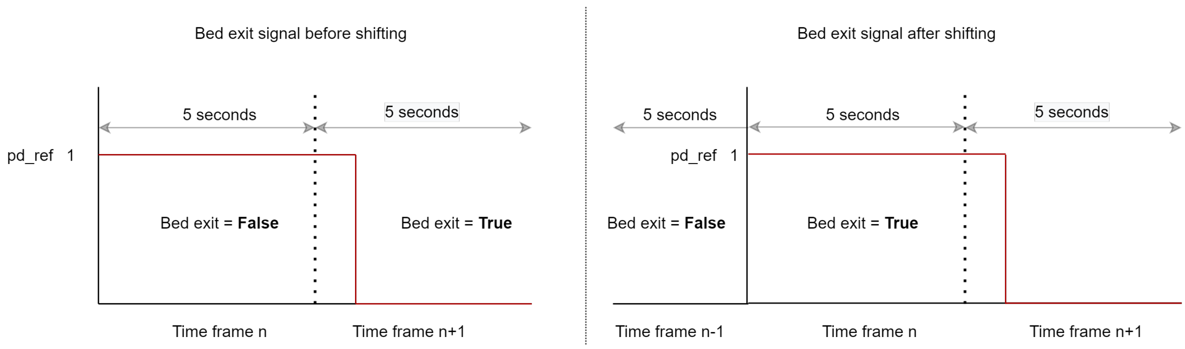


Figure 3.4: Shifting of the bed exit signal

3.3. New created data set

The reference data however showed some inconsistencies and did not contain enough samples. These inconsistencies were situations in the sensor data that could not be explained from the testing protocol. Therefore it was decided to create a new data set. The setup for testing can be seen in Figure 3.5. The BedSense was placed around the chest area as the device was designed for. To get enough samples, 98 bed exits were performed. When the person entered the bed, a button was pressed. Right after this person exited the bed, the button was pressed again. To avoid creating a certain pattern in when a bed exit would occur, a minimal break of 30 seconds was taken between each bed exit. The movements done when lying in bed included rolling over, sitting up straight, moving the head, sitting on the side of the bed and talking. These actions were done at random to again avoid a pattern in movements. The test data was created using two different mattresses: 49 bed exits with an anti-decubitus mattress and 49 with a regular mattress. The first mattress showed a higher sensitivity when it comes to sensor data than the second mattress. Figure 3.6 shows the FSR values that were obtained. The first mattress was used from 10:10 to 11:42 and the other mattress from 11:50 on. The peaks of the FSR values in the first part deviate around a much higher value than the peaks of the FSR sensors in the second part.

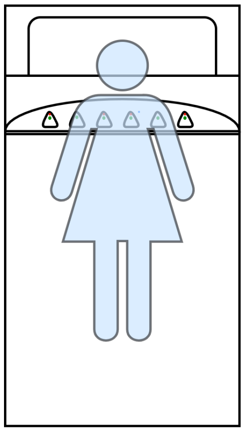


Figure 3.5: Placement of the BedSense during testing

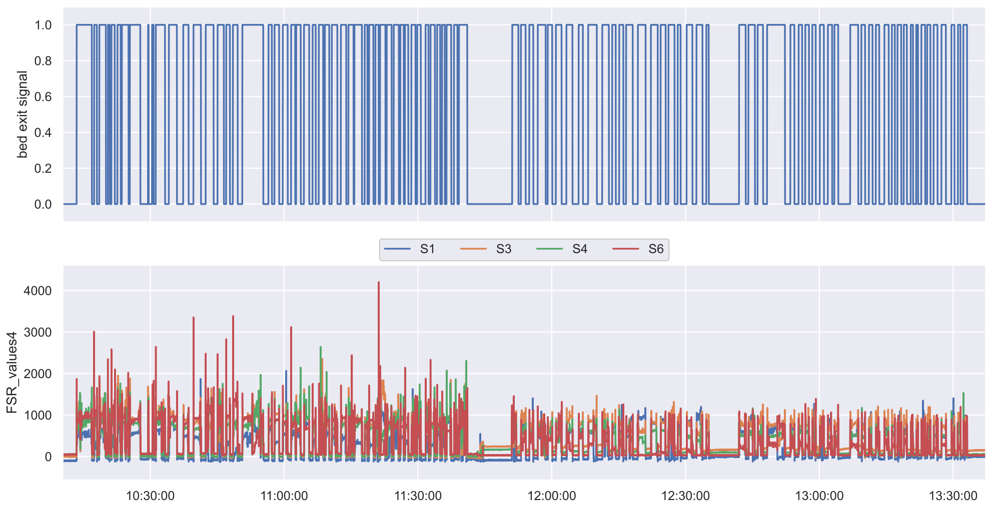


Figure 3.6: Data set containing the FSR sensors & the bed exit signal for the two different mattresses

4 Feature Extraction and Selection

After performing the data description a set of features is extracted from the raw data. Features are individual measurable properties to detect patterns in the data set. They are suitable for training models. Feature extraction means reducing the dimensions of the data set by creating new, smaller sets of features to make the data better manageable. The useful features are then selected by filtering out irrelevant features and this will be the basis of the input signals for the designed systems. This chapter will provide an elaboration on extracting and selecting features as input for our algorithm.

4.1. Feature Extraction

The features presented in Table 4.1 are derived from the raw data of the sensors on the BedSense system. The features are constructed for each time frame of 5 seconds. The FSR sum is calculated by adding the value of each sensor at each time sample and then by taking the mean of the FSR sum of all time samples within a window. The sum of the FSR sensors could detect whether someone is in or out of bed: all sensors give a higher value when someone is in bed and a lower value when someone is out of bed. Thus, this feature might as well be relevant for predicting when someone is about to leave the bed. The PE sum is determined in the same way as the FSR sum.

Furthermore, the means of the FSR and PE of the separate sensors are calculated for each window. Namely, a difference in the FSR or PE mean between time frames could indicate that someone is awake and rolling over in bed. When someone is in unrest, the values of each sensor show more and higher variations from the mean than when someone is lying still. Therefore, the standard deviation of the FSR and PE sensors within a window was created as feature. This also holds for the peak to peak value, which is the difference between the maximum and minimum reached value within that frame. A high peak most of the time occurs when a patient is about to leave the bed, so this feature was also created.

Moreover, when someone is rolling over to one side, the sensors on that side measure more force while the sensors on the other side sense less force. Thus, it seemed useful to look at the difference between the values of two sensors. For example, a small Δ_{13} or Δ_{46} means that someone has more weight on one side, thus is leaning over. A high difference means that the patient is lying in the middle. The difference is calculated by subtracting the FSR sensor values in a window from each other and by then taking the mean of the differences, which results in a difference for each window.

$$FSR_{pkpk} = FSR_{max} - FSR_{min} \tag{4.1}$$

The peak to peak values of each FSR sensor for each window are defined as in the equation above. The maximum value of the FSR sensor in a window is taken and then the minimum value of the FSR sensor in a window is subtracted. This gives the peak to peak value of the FSR sensor in a window.

4.2. Feature Selection

In order to compare the features, they are standardised. Standardisation is a scaling technique to center the values around the mean with a unit standard deviation. The mean of the feature becomes

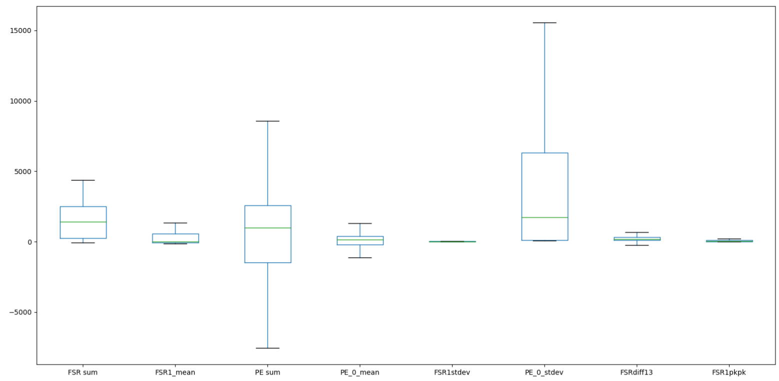
Feature	Description
FSR sum	Total sum of FSR sensors
FSR mean	Mean of the FSR sensors
PE sum	Total sum of PE sensors
PE mean	Mean of the PE sensors
σ_{FSR}	Standard deviation of each FSR sensor
σ_{PE}	Standard deviation of each PE sensor
Δ_{13}	Difference between FSR sensor 1 & 3
Δ_{46}	Difference between FSR sensor 4 & 6
Δ_{14}	Difference between FSR sensor 1 & 4
Δ_{36}	Difference between FSR sensor 3 & 6
FSR peak to peak	Difference between the minimum and maximum value of each FSR sensor

Table 4.1: Created features per time frame of 5 seconds

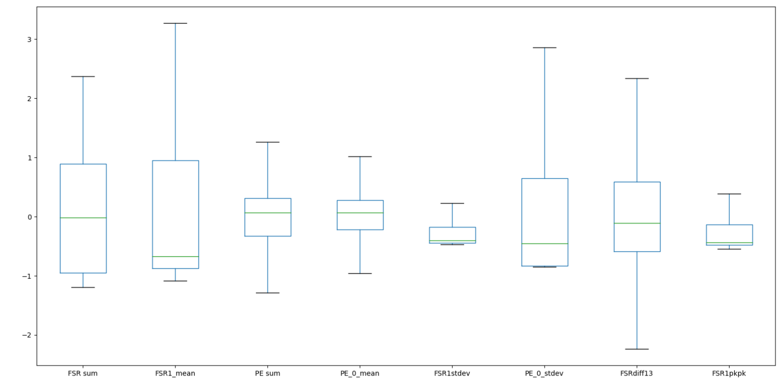
zero and its distribution has a unit standard deviation. Standardisation is done by applying the following equation.

$$X' = \frac{X - \mu}{\sigma} \tag{4.2}$$

The result is shown in the boxplots in Figure 4.1.



(a) Boxplot of the features



(b) Boxplot of the standardised features

Figure 4.1: Boxplots of the features

After standardising the features, a correlation matrix is plotted where the correlation coefficient between each feature and the bed exit is calculated using:

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \tag{4.3}$$

The correlation coefficient indicates a relationship between the two variables. When the coefficient is +1, the values have an ideal increasing linear relationship. The coefficient is -1 in case of an ideal decreasing linear relationship. A coefficient closer to -1 or +1 means that the variables have a stronger relationship. In this case the last column and last row are the most relevant. These show the correlation between the features and the bed exit signal.

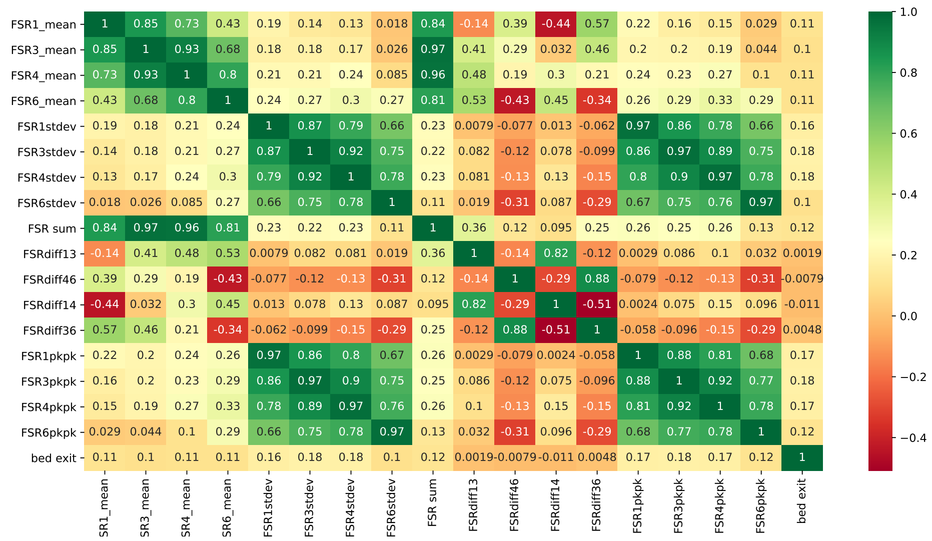


Figure 4.2: The correlation matrix of the FSR features and the bed exit signal

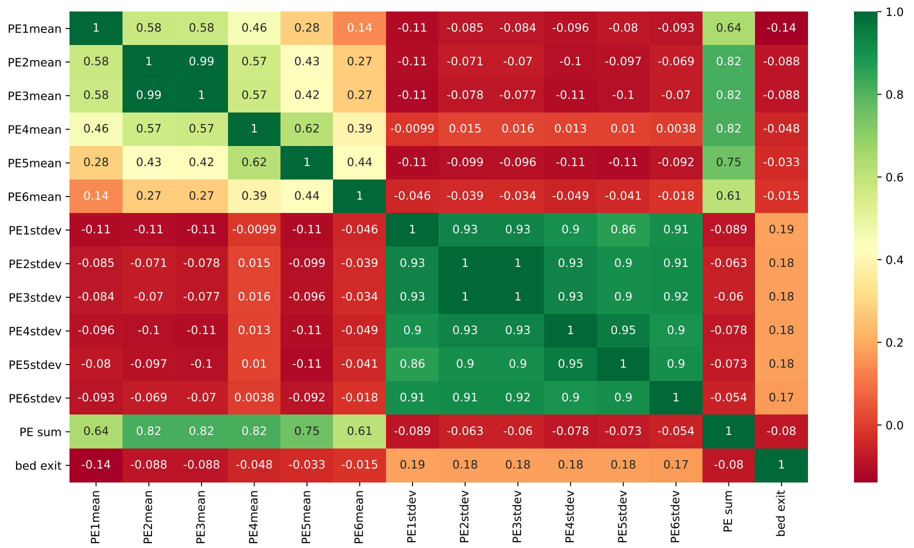


Figure 4.3: The correlation matrix of the PE features and the bed exit signal

From the correlation matrix of the FSR features depicted in Figure 4.2 it can be concluded that the peak to peak values of the FSR sensors and the standard deviation of the FSR sensors are the most useful for predicting a bed exit, because those features have the highest correlation with the bed exit signal. The correlation matrix of the PE features as shown in Figure 4.3 indicates that the standard deviation of the PE sensors gives the highest correlation value. Therefore, the standard deviation of the PE and FSR sensors and the peak to peak value of the FSR sensors will be used as input for the algorithm, which will be developed in the next chapter.

4.2.1. Weighted factors

Previous time frames may contain information about a possible bed exit. In order to take into account the past values of the sensors in previous time frames, the sum of weighted features is used. The features that are described above are accumulated in the following equation.

$$\sum_{k=0}^{n-1} f(t-k)w_k \tag{4.4}$$

In this equation $f(t)$ is the feature itself, as a function of the t^{th} time frame, and w_k is the allocated weight. The weight depends on the amount of past time frames n that need to be considered.
The weights are determined by:

$$w_k = \frac{n-k}{n} \tag{4.5}$$

For this weight function it was observed for which n the features correlate best with the bed exit signal. As the standard deviation showed the highest correlation, this feature is used to compare the amount of frames.

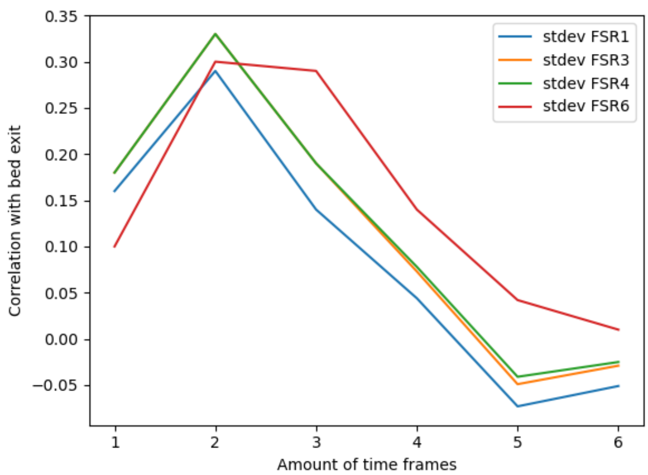


Figure 4.4: The correlation of the standard deviation of the FSR sensors weighted with n time frames

In Figure 4.4 the correlation of the standard deviation of the FSR sensors with the bed exit signal is plotted when weighted with n time frames in order to see whether it is useful to make use of past time frames. From this graph it can be concluded that the maximum correlation is achieved when n is 2. This implies that the information of the current time frame should be combined with information from the previous time frame, so the features during 10 seconds are used. Thus, for predicting a bed exit Equation 4.4 with w_k as in Equation 4.5 with $n = 2$ is used for each feature. The correlation matrix of the features that will be used for the input while weighting the past time frame is given in Figure 4.5.
Moreover, a feature $X_{combined}$ was created which includes the three features that are most relevant for the algorithm. This feature was computed by weighting the features according to their correlation with the bed exit signal as:

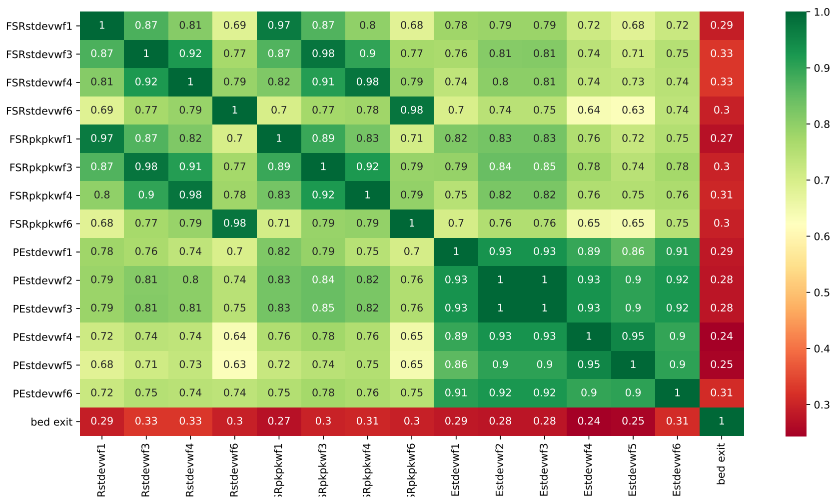


Figure 4.5: The correlation matrix of the relevant features with weighting the past time frame and the bed exit signal

$$X_{combined} = i \cdot \sigma_{FSR} + j \cdot FSR_{pkpk} + k \cdot \sigma_{PE} \tag{4.6}$$

where i, j and k are the weights that are assigned to a specific feature based on their correlation with the bed exit signal. In the next chapter they are varied to find the optimal result for predicting a bed exit.

5 Prediction Models & Results

Now that the features have been selected for the algorithm, this chapter elaborates on the design of the algorithm. Two machine learning approaches are compared to deal with the classification problem. Thereafter, another algorithm to tackle the problem without machine learning is explained.

5.1. Machine Learning Approaches

Machine learning is a form of artificial intelligence in which training data sets are used to make predictions for the future. A lot of research has already been conducted in the field of machine learning. These researches are helpful to decide what method could work for predicting a bed exit. Machine learning is classified in two categories: supervised and unsupervised learning. In case of unsupervised learning the algorithm tries to find patterns between the input data and finds features itself. Supervised learning is used when prior knowledge about the data is available in order to train the algorithm. As it is easier to understand the working of the machine learning model in case of supervised learning, only supervised learning models are considered for this project.
Furthermore, predicting a bed exit is a binary classification problem: The patient either exits the bed or not. For the created data these two target classes are already known as the data is labeled. Supervised learning is covered in many machine learning models. Due to a limitation in the training time and because of the comprehensibility of the algorithm, only simple machine learning methods are taken into consideration. In this report a Decision Tree and a Support Vector Machine (SVM) are used and evaluated respectively. SVMs are based on the principle of finding a margin which separates two classes. The margin, or hyperplane, is maximised to create the largest distance between the two classes. SVMs are suitable when the number of features is large compared to the training data. Decision Trees split the data set at the root node into subsets for successor nodes based on certain rules. They are preferable as they can be visualised [16] [3]. Both models are simulated using the Scikit-Learn library in Python [23]. This library includes several machine learning algorithms and supports Python numerical and scientific libraries.

The performance of a classifier is evaluated by using several statistical measures: the true positive rate (TPR), the true negative rate (TNR), the false discovery rate (FDR), the false positive rate (FPR), the false negative rate (FNR) and the accuracy (ACC). These measures can be calculated as:

$$TPR = \frac{TP}{P} \cdot 100 \tag{5.1}$$

$$TNR = \frac{TN}{N} \cdot 100 \tag{5.2}$$

$$FDR = \frac{FP}{FP + TP} \cdot 100 \tag{5.3}$$

$$FPR = \frac{FP}{N} \cdot 100 \tag{5.4}$$

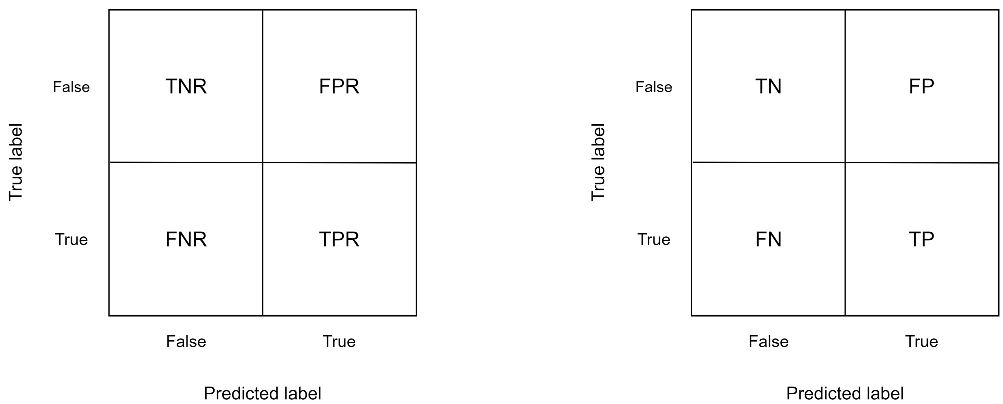
$$FNR = \frac{FN}{P} \cdot 100 \tag{5.5}$$

$$ACC = \frac{TP + TN}{P + N} \cdot 100 \tag{5.6}$$

Variable	Description
Positives (P)	Number of real positive cases in the data
Negatives (N)	Number of real negative cases in the data
True positives (TP)	Number of correctly predicted positives
True negatives (TN)	Number of correctly predicted negatives
False positives (FN)	Number of falsely predicted positives
False negatives (FN)	Number of falsely predicted negatives

Table 5.1: Terminology of confusion matrix

Table 5.1 explains the variables that are used for computing the statistical measures. The sensitivity or true positive rate gives the proportion of positives that is correctly classified. The sensitivity thus gives insight in how many bed exits are correctly predicted with respect to the total real bed exits. Furthermore, the specificity is defined as the true negative rate. This is the proportion of time frames that are correctly classified as a non-bed exit in relation to the total amount of real non-bed exits. The false discovery rate is the amount of false positives in comparison with the false and true positives. This rate shows the proportion of false bed exit alarms in comparison to the total amount of predicted bed exits. Additionally, the false positive rate or false alarm rate is the ratio between the false positives and the number of real negatives. The false alarm rate in case of the bed exit prediction is the rate with which the prediction model predicts a bed exit while there is no real bed exit. The ratio between the false negatives and the positives is defined as the false negative rate. These are the cases in which a bed exit occurs that are classified as no bed exit. Finally, the accuracy of the classifiers is compared. The accuracy is the amount of true predictions versus the amount of total cases. The higher the amount of correctly predicted time frames, the higher the accuracy of the classifier is. A confusion matrix visualises the performance of a classifier in a matrix. In Figure 5.1 the confusion matrix is explained in terms of the variables in Table 5.1 and the statistical measures given above.



(a) Normalized confusion matrix (b) Confusion matrix

Figure 5.1: The confusion matrix explained

5.1.1. Decision Tree

A decision tree is a model that predicts the class of a target variable by setting simple decision rules derived from the data features. A decision tree consists of nodes, branches and leaf nodes. The algorithm splits the data at a node based on a conditional statement. Then, the data is connected by a branch to the successor node corresponding to the outcome of the conditional statement of the previous node. This way, the data is repeatedly split in smaller samples until the data reaches one of the leaf nodes: the concluding nodes that predict the outcome. There are two types of decision trees:

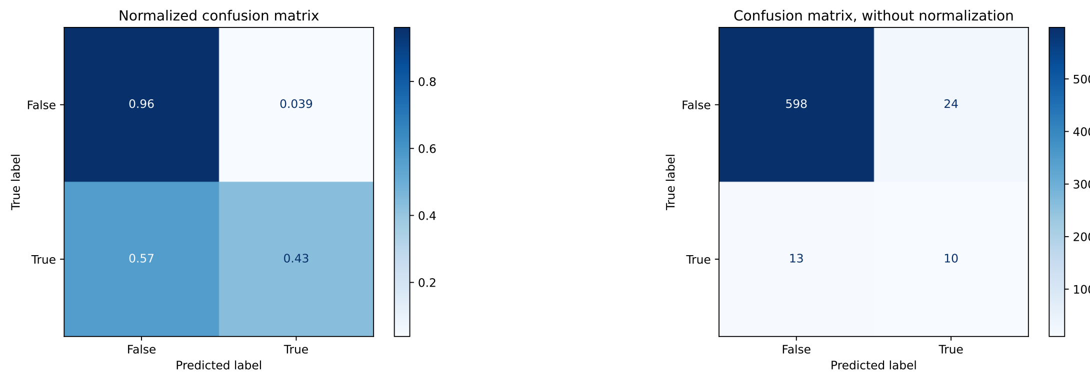
regression trees and classification trees. In case of a regression tree the target variable is a value, whereas the target variable of a classification tree is categorical or discrete. A classification decision tree is implemented for the purpose of predicting a bed exit. The data set, the self-made reference data as described in Chapter 3, is split into a training and test set. The *train_test_split* method of the Scikit-Learn library is used to split 25% of the data into the test set and 75% into the train set. The test set is useful for estimating the performance of the classifier. The classifier, implemented by the class *DecisionTreeClassifier*, is developed while using the set of training samples of the labeled data. This class obtains various parameters to adapt the depth of the tree and its default values give an optimal classifier. The classifier determines the threshold per node to split the data to the successor node. The method *fit* is then used to train the algorithm.

In Table 5.2 the statistical measures as described in the previous section are calculated for different input features.

Input feature	Sensitivity (%)	Specificity (%)	False discovery rate (%)	False alarm rate (%)	Accuracy (%)
Raw data of 4 FSR sensors and 6 PE sensors	26.1	96.6	78.0	3.4	94.1
All features as in Table 4.1	13.0	95.2	90.9	4.8	92.2
$FSR_{peaktopeak}$ and σ_{FSR} with weighting the past time frame	17.4	97.4	80.0	2.6	94.6
$FSR_{peaktopeak}$, σ_{FSR} and σ_{PE} while weighting the past time frame	21.7	96.8	80.0	3.2	94.1
$FSR_{peaktopeak}$	13.0	95.5	90.3	4.5	92.6
$FSR_{peaktopeak}$ while weighting the past time frame	26.1	96.3	79.3	3.7	93.8
σ_{FSR}	21.7	95.0	86.1	5.0	92.4
σ_{FSR} with weighting the past time frame	43.5	96.1	70.6	3.9	94.3

Table 5.2: Statistical measures of the decision tree classifiers with different input features

The input features that score highest in sensitivity, specificity and accuracy and lowest in false discovery and false alarm rate are the most relevant for our algorithm. In Table 5.2 it is observed that the standard deviation of the FSR sensors weighted with the previous time frame scores best in terms of sensitivity, false discovery rate and in accuracy. Its normalised confusion matrix is shown in Figure 5.2.



(a) Normalized confusion matrix

(b) Confusion matrix

Figure 5.2: The confusion matrices of the classifier with σ_{FSR} (with weighting the past time frame) as input

The decision tree is given in Appendix B.1. Due to the large amount of nodes and conditional statements, the tree could not be visualised without being unreadable and therefore the decision statements can be read in this code. Additionally, several decision tree classifiers were created using the feature $X_{combined}$ as input while varying the weights of the features. After trying different weights, it seemed that the statistical measures of the classifier were not sufficient for predicting a bed exit. The sensitivity of the classifier did not reach a higher value than 4.3%. Therefore, the feature $X_{combined}$ was discarded. Furthermore, it was examined which time frame would work best as input for the decision tree. In Figure 5.3 it is shown how the decision tree performs with respect to different time frames. As can be seen in the graph, the time frames of three and five seconds clearly give the highest sensitivity. The false alarm rate does not differ significantly. The earlier a bed exit is predicted, the higher the chance a bed-fall can be prevented. That is why time frames of five seconds are preferred over time frames of three seconds.

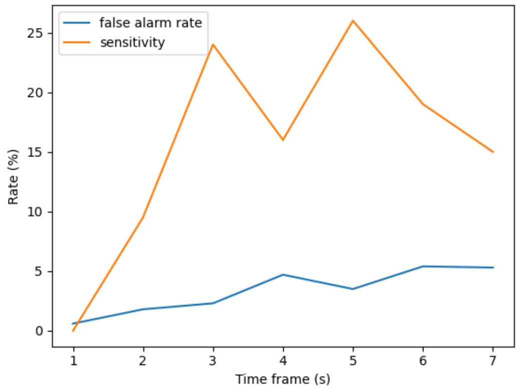


Figure 5.3: The false alarm rate and sensitivity rate when using various time frame windows

5.1.2. Support Vector Machine

Like the decision tree algorithm, the support vector machine is often applied to classification and regression problems. An SVM tries to find a hyperplane between the space of the input features to distinguish several classes. SVMs maximise the margin around the separating hyperplane, so that the distance between data points of both classes is maximised. The decision function is then defined by a subset of training samples, the support vectors. An SVM works for linearly separable data. Through the use of kernel functions non-linear regions can be separated by an SVM as well. This is done by mapping the data to a higher dimensional space to achieve linear separation.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \tag{5.7}$$

The kernel function returns the inner product between two points in the transformed space, see Equation 5.7, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are input vectors. There are different types of kernel functions. Two examples are the polynomial kernel function as in

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d, \tag{5.8}$$

where d is the degree of the polynomial and the radial basis function as in

$$K(\mathbf{x}_i, \mathbf{x}_j) = exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\right). \tag{5.9}$$

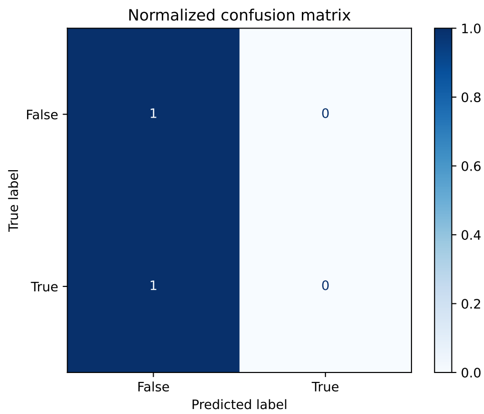
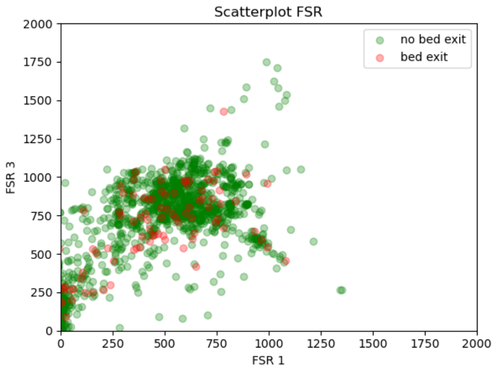
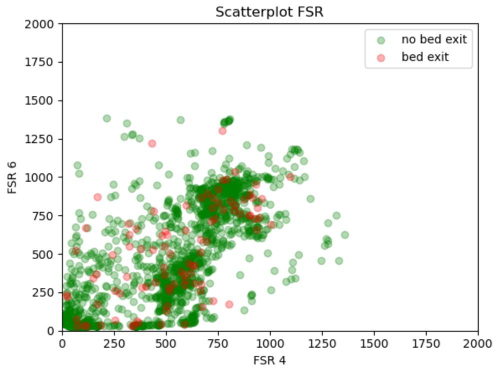


Figure 5.4: The confusion matrix of the SVM

A support vector machine works well with high dimensional spaces and is less sensitive to outliers due to the support vectors of the hyperplane. For the prediction of a bed exit, an SVM is implemented. The implemented SVM without using a kernel trick resulted in the confusion matrix shown in Figure 5.4. This indicates that the SVM was not able to classify a true bed exit prediction. Even while varying the input features, the classifier was not able to predict a true bed exit. To find out why the support vector machine would not succeed in predicting a bed exit, the raw FSR sensor data was visualised in relation to a bed exit in the next time frame.



(a) Scatterplot of the raw FSR values of sensor 1 and 3



(b) Scatterplot of the raw FSR values of sensor 4 and 6

Figure 5.5: Scatter plots of the FSR values

In Figure 5.5 there is no clear margin of separation between a bed exit and no bed exit. This means that it is indeed not possible to distinguish a bed exit from a non-bed exit by finding a decision boundary between the raw FSR data. Another SVM is developed by calculating the standard deviation with weighted time frames as input. Figure C.1 gives a scatter plot of the standard deviation of the FSR values with weighting the past time frame. For these visualisations it is again true that there is no clear margin of separation. It is thus not possible to distinguish a bed exit from no bed exit by constructing a straight line which separates the data in both classes. In case the data set is inseparable in the current dimensions, the kernel trick is used, which maps the non-linear separable data set in a higher dimensional space to find a hyperplane that can separate the classes. The polynomial kernel function and the radial basis kernel function are used respectively. The result is the same normalized confusion matrix as when no kernel trick is used (see Figure 5.4). Thus, even when applying the kernel trick, the support vector machine is not able to predict a bed exit. Although the data seems to show no clear separation, decision trees in contradiction to SVMs can fit linearly inseparable data sets. This explains why the decision tree is able to classify a bed exit prediction, but the SVM is not.

5.2. Prediction Model without Machine Learning

It is also researched whether it is possible to construct a prediction model without using machine learning. By applying thresholds, that are determined by analysing FSR values in the data set, the 'prediction' signal was made to determine whether someone was either about to leave the bed (1) or not (0). In Figure 5.6 the signal is plotted. As can be seen in the picture, the prediction signal is similar to the pd_ref signal, the reference signal, apart from some glitches. This means that the model detects a bed exit, whereas, it does not predict a bed exit. The thresholds were set by looking at a visualisation of the input features, as shown in the Appendix in Figures C.2, C.3, C.4, C.5, C.6, C.7 and C.8. In those figures it is observed that all features give peaks just before or at the moment of a bed exit. The PE_mean and PE_sum features have peaks everywhere and it is thus not useful to set a threshold for those features.

Additionally, the FSR_sum is a noisy signal, which makes it too sensitive for setting a threshold. The plot of the differences between certain FSR sensors does not give consistent information on when someone is about to exit the bed as well. Setting a threshold for the FSR_mean and raw FSR values could be used for the detection of a bed exit, however, predicting a bed exit with those features will be difficult as there is no clear pattern observed before someone leaves the bed. The standard deviation of the FSR values and the peak to peak FSR values give peaks when someone exits the bed. The peaks sometimes occur at other moments as well. After looking at several bed exits and the plots of the features, predicting a bed exit seemed unfeasible at least five seconds beforehand by setting thresholds manually.

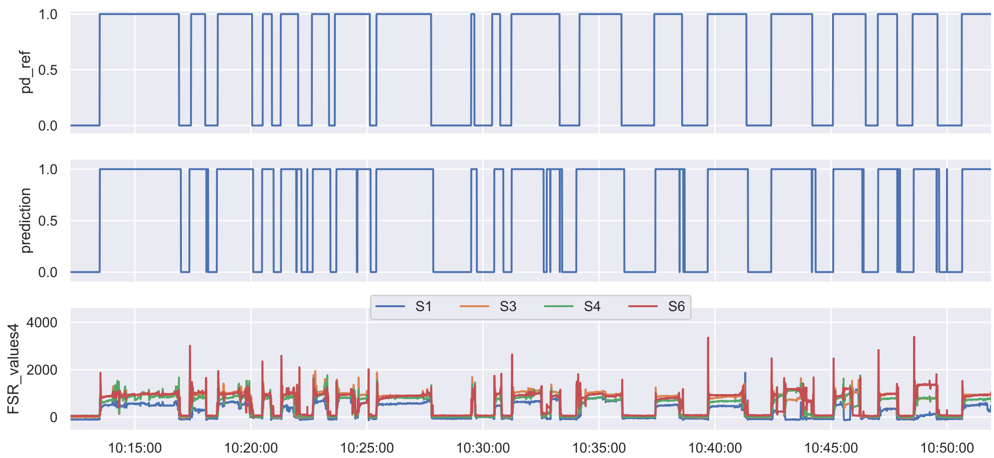


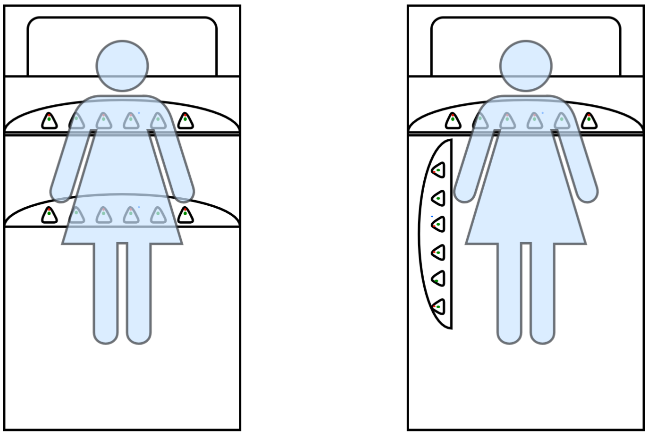
Figure 5.6: Prediction of a bed exit without machine learning

6
Additional Research

In this chapter more research on the data sets was done through testing. Different routines to get up out of bed are simulated and the results of these data sets are analysed. From analysing the different movements it is decided whether adding an extra sensor plate is of added value when predicting a bed exit. The results are then compared to the patient data and conclusions will be made for further work.

6.1. Testing

To look at other options with the device, more testdata was created adding another sensor plate in a different position. The test setups used are shown in Figure 6.1. The second sensor plate was added around the hip area and at the right side of the bed, which is also the side used for standing up.



(a) Setup 1 (b) Setup 2

Figure 6.1: The two testing setups

The results of the testing can be seen in Figure A.1. As with the reference data, pd_ref shows the bed status of the patient. The two devices each have their own frame. It is seen that the added device shows significantly larger FSR values at each time sample in both test setups, so more information is gathered from two sensor plates in comparison to only having one sensor plate.

Specific movements were simulated to further analyse the sensors during certain situations. The routines chosen include advised protocols for fall prevention for eldery, especially who experience any kind of dizziness (e.g. orthostatic hypotension), which are common conditions among older people [26] [29]. The visualised data of a few of these situations can be seen in Appendix A.1. This also creates some reference for the unlabeled data sets of patients who are at a higher risk of falling, as stated in Chapter 3. Figure A.2 shows what happens when someone rolls over to the left, then to the right and then leaves the bed using their arms. The device placed around the chest area clearly shows that when rolling to the left, FSR4 and FSR6 have the highest pressure with a small Δ_{46}. When rolling to the right, FSR1 and FSR3 show the most pressure and Δ_{13} is small. When a bed exit is about to happen, the added plate for both setups is useful for obtaining more information from the FSR patterns. For this reason, the extra sensor plate is useful for creating more features that are crucial for predicting a bed exit. When combining the two devices, the FSR patterns are more interpretable and a clear situation of how someone exits the bed can be sketched from the data sets.

To check this for another routine before a bed exit, someone now sits up straight in bed from a lying down position, lies down again and then leaves the bed using his or her arms. This routine is shown in Figure A.3. While someone sits up straight, the values of the original sensor plate drop in both setups and it seems as if someone is getting out of bed, which is not true. Therefore, the added plate is essential to know that no bed exit should be predicted. When a bed exit is about to happen, more deviation in the values of the FSR sensors at the hip occurs. The FSR sensors of the plate on the side of the bed show a high peak and more movement. Thus by using the earlier derived features with the extra sensor plate, a better prediction will be possible. In both cases, the second plate is of added value.

The following bed exit movement is especially interesting as this is based on the safety protocol for leaving the bed for institutionalised patients [11]. Figure A.4 shows what happens when someone first sits on the side of the bed, then waits a few seconds and then gets up to get out of bed. The FSR values of the original plate drop when someone sits on the side of the bed. By only considering these values, the algorithm will think that someone is getting out of bed. However, this is not yet the case and the added plate confirms there is still pressure on the bed. To conclude, the added plate is of high relevance for a correct bed exit prediction. Another example to verify that this is the case can be seen in Figure A.5.

For the following routine the same fall prevention protocol as above was performed, only this time it was chosen to leave the bed on the other side of the bed. This bed exit is shown in Figure A.6 and again someone first sits on the left side of the bed, then waits a few seconds and then leaves. Since the added plate is placed on the right side of the bed, this device does not show any pressure the crucial moments before the bed exit and will not provide more information. The sensors at the hip show deviation in FSR values and it is clear that the hip device is useful.

The plate at the side is especially of essence when located at the preferred bed exit side. In general it is advised to design the environment of institutionalised patients according to their preferences [12], so the preferred bed exit is mostly known. To conclude the added side plate is a better addition compared to the sensor plate located under the hip.

6.2. Patient data

The created reference data was compared to the unlabeled data sets of patients in nursing homes who are at a higher risk of falling. A part of this data can be seen in Figure 6.2. The bed_status signal is a signal generated by Momo Medical that is already being used for continuous data of patients at nursing homes. A status of 2 means that the patient is in bed and lies above the sensor plate. When the signal is 1, the patient is presumably sitting upright in bed and 0 means that the patient is out of bed. As this data is unlabeled, this is the only reference available to know what the status of the patient is. However, since this signal is generated and not 100% accurate, it is not reliable enough to know the definite status of the patient at all times.

The visualised data frame shows a similar pattern to the tossing and turning data set shown in Figure A.2. Only this patient rolls over to the left, then to the right, then back to the left and then presumably gets out of bed. The FSR values drop at 08:17 but the bed_status gives that the patient is still in bed until 08:23. The status should have gone to 1 or 0 at 08:17 but this is not the case, so it is unclear whether the patient is actually out of bed or is sitting upright in bed. If a second plate was added, the definite state of the patient in bed is known. Figure A.7 shows a similar situation where the FSR values drop but the patient is still in bed according to the bed status, so another BedSense is needed. In conclusion for real patient data another sensor plate would be of essence.

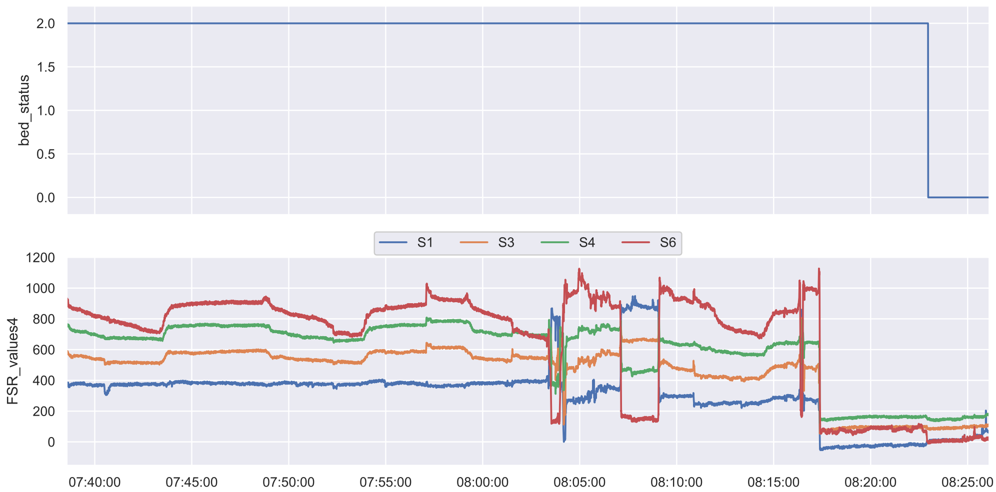


Figure 6.2: Visualisation of a bed exit of a patient who is at higher risk of falling

7 Discussion

7.1. Model results

Regarding the results presented in the previous chapter, the classifier predicts a bed exit with a sensitivity rate of 43.5%. This chapter elaborates on several factors that possibly have affected the performance of the classifier. The scatterplots of the raw FSR sensor data and weighted standard deviation show that both data sets are linearly inseparable. The decision tree algorithm shows that it is possible to classify the problem, but the SVM method is not able to do this.

7.2. System design

The current model is based on features extracted from labeled data. The button used to generate the reference signal for the labeled data, was manually pushed which can give some slight inconsistencies in matching the timestamps to the exact timestamp that the bed exit is occurring. This data set is created by two healthy people on two different mattresses with similar body types. The algorithm should also be tested for more people with different body types and the data should include more bed exits where someone gets out of bed in the way elderly do. Also, people do not always know that they are about to get up 5 seconds before. For elderly, getting up could take longer than for healthy younger people. Further research shows that it is easier to design a prediction model for institutionalised people who follow an advised routine for fall prevention when leaving the bed. The protocol for this is to sit up from a laying down position, sit on the side of the bed and wait, before getting up out of bed. This means that a prediction can be done earlier on than 5 seconds when this routine is a returning pattern in the sensor data.

Moreover, the data set is divided in time frames of 5 seconds to make the data set usable for the prediction model. In real life, there are no time windows and the data is continuously being sent. Also, the time frames of 5 seconds are chosen to predict the bed exit before it happens. The model predicts a bed exit beforehand but there is still no guarantee that a bed exit is always predicted 5 seconds ahead.

The machine learning algorithm requires more samples than the reference data at this moment obtains, which is 98 bed exits. However, patients in nursing homes will only leave their bed a few times, especially during the night. These bed exits occur with much more time between each exit than the exits generated in the reference data and are spread over a larger time period. Since the unlabeled patient data does not contain much information on the exact movements of the patient, no immediate conclusions can be drawn from this data set and it is not usable for the algorithm. Therefore, it is hard to test the algorithm with 'real' patient data.

Finally, machine learning models are like a black box. Without exactly programming the whole algorithm, it uses techniques to learn from input data. Once the model is trained, it returns the decisions made. The exact steps taken to make the decisions are hard to discover.

8 Conclusion

To conclude, the goal of developing an algorithm which predicts a bed exit has been reached. The algorithm satisfies the must-have requirements. Not all should-have requirements are met, such as predicting a bed exit at least 5 seconds beforehand and predicting a bed exit for different body types. The could-have requirements are not met as they were less of a priority.

The bed exit prediction algorithm is based on features that are derived from the FSR and PE sensors of a device which lies under the mattress at the level of someone's chest. The algorithm is a decision tree classifier and predicts the bed exit ahead for self-made reference data. The feature which is used as input for the algorithm is the standard deviation of the FSR sensors. The feature includes the standard deviation within the current 5 second window and the weighted standard deviation from the previous 5 second window. The decision tree gives a sensitivity of 43.5%, a false alarm rate of 3.9% and a classification accuracy of 94.3% when the data is split in windows of 5 seconds. It is concluded that it is not possible to predict a bed exit by using a support vector machine or by setting thresholds manually. Detection of a bed exit is feasible using machine learning or setting thresholds manually.

An additional sensor plate at the level of the hip or on the side of the bed is useful for gathering more information regarding a bed exit. The extra sensor plate is valuable for distinguishing when someone sits up straight. In this case, the sensor plate at the level of the chest detects little pressure. Especially for patient data an extra sensor plate is of added value to define the bed status of the patient, especially on the side of the bed when the preferred bed exit side is known.

8.1. Recommendations

From the scatter plots it seems almost impossible to predict a bed exit based on the raw FSR data or the standard deviation of the FSR values. In addition, human behaviour is hard to predict. Therefore, it is unfeasible to make a reliable bed exit prediction 5 seconds beforehand while using the sensor plate as input source. The performance of the decision tree classifier is therefore limited in predicting a bed exit. It is thus recommended to look for other ways to predict a bed exit. Adding an extra sensor plate for predicting a bed exit would result in more reliable predictions. When the preferred bed exit side is known, it is recommended to use the extra sensor plate on the side of the bed. Otherwise, it is recommended to use an extra sensor plate at hip level. Furthermore, the single sensor plate could be used in combination with other sensors in order to achieve a more reliable system, for example a camera or infrared sensor.

8.2. Future Work

The algorithm does not comply with all design requirements. Therefore, some improvements that could be useful for further development of a reliable bed exit prediction model are listed below:

- Using the features described in this report for two sensor plates as input for the decision tree algorithm will probably give more reliable predictions. This should be tested in order to evaluate the performance of the classifier when using two sensor plates.

- The decision tree works with features derived from data that is split into windows of 5 seconds. To imitate realtime situations, a sliding window of 5 seconds should be used instead of splitting the whole data set into windows of 5 seconds.
- The algorithm is based on a data set containing bed exits from two people with similar body types performed on two different mattresses. In the future labeled patient data should be used as input for machine learning as this is the target group for the bed exit prediction. This data needs to have many bed exit samples for the machine learning algorithm to be able to train itself. Furthermore, it is recommended to have data of people with different body types and bed exits that are performed on different mattresses.
- The heart and respiratory rate could reveal more insights in a bed exit prediction. Namely, it is believed that patients' heart rate or respiratory rate changes before they leave bed [8]. Including the heart rate and respiratory rate in the input features could give more accurate bed exit predictions. In Figure 8.1 an example of the implementation of the system using the heart and respiratory rate is shown.

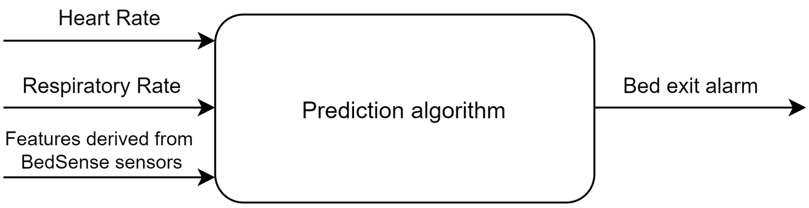


Figure 8.1: An example of implementing the whole system

Bibliography

[1] J. Agostini, D. Baker, and S. Bogardus. Chapter 26 . prevention of falls in hospitalized and institutionalized older people. 2001.

[2] Bernard Boulay and Monique Thonnat. Applying 3d human model in a posture recognition system. *Pattern Recognition Letters*, 27:1788–1796, 11 2006. doi: 10.1016/j.patrec.2006.02.008.

[3] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 161–168, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143865. URL https://doi-org.tudelft.idm.oclc.org/10.1145/1143844.1143865.

[4] CBS. *Prognose: 19 miljoen inwoners in 2039*, 2019. URL https://www.cbs.nl/nl-nl/nieuws/2019/51/prognose-19-miljoen-inwoners-in-2039. (Accessed on October 13, 2020).

[5] CBS. *115 duizend mensen in verzorgings- of verpleeghuis*, 2020. URL https://www.cbs.nl/nl-nl/achtergrond/2020/13/115-duizend-mensen-in-verzorgings-of-verpleeghuis. (Accessed on October 13, 2020).

[6] T. Chen, R. Hsiao, C. Kao, H. . Piao Lin, S. Jeng, and D. Lin. Vision-assisted human motion analysis for bed exit prediction model construction. In *2017 International Conference on Information, Communication and Engineering (ICICE)*, pages 542–545, 2017. doi: 10.1109/ICICE.2017.8479094.

[7] Chiu-Kuel Li Min-Zheng Liu Po-Chieh Kuo Chien-Nan Lee, Su-Chen Yang. Alarm system for bed exit and prolonged bed rest. *2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, pages 1–5, 2018.

[8] ClinicalTrials. Predict near future initiation of bed exit, January 2014. URL https://clinicaltrials.gov/ct2/show/study/NCT01774708.

[9] F. Finbarr D. Oliver and P. Seed. Preventing patient falls. *Age and Ageing*, 31(1):75–76, 2002. doi: 10.1093/ageing/31.1.75.

[10] Eresha Fernando, Michelle Fraser, Jane Hendriksen, Corey Kim, and Susan Hunter. Risk factors associated with falls in older adults with dementia: A systematic review. *Physiotherapy Canada*, 69:1–10, 10 2016. doi: 10.3138/ptc.2016-14.

[11] Medical Guardian. Fall prevention awareness month: Bedroom safety, September 2019. URL https://www.medicalguardian.com/medical-alert-blog/senior-safety-/fall-prevention-awareness-month-bedroom-safety.

[12] Van Vliet-M. van der Leeuw J. Gerretsen-A. Hardeman, F. Aternatieven voor vrijheidsbeperking in de zorg. URL https://www.vilans.nl/docs/vilans/publicaties/Alternatievenbundel_vrijheidsbeperking_2014.pdf.

[13] A. Harma, W. ten Kate, and J. Espina. Bed exit prediction based on movement and posture data. In *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 165–168, 2014.

[14] A. Ibrahim, K. Chaccour, A. H. El Hassani, and E. Andres. Bed-fall detection and prediction: A generic classification and review of bed-fall related systems. *IEEE Sensors Journal*, pages 1–1, 2020. doi: 10.1109/JSEN.2020.3037711.

[15] M. Inoue, R. Taguchi, and T. Umezaki. Bed-exit prediction applying neural network combining bed position detection and patient posture estimation. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3208–3211, 2019. doi: 10.1109/EMBC.2019.8857233.

[16] S.B. Kotsiantis. Supervise machine learning: A review of classification techniques. *Frontiers in Artificial Intelligence and Applications*, 160:3–15, 2007.

[17] Li Tianze, Zhang Xia, Jiang Chuan, and Hou Luan. Analysis of the characteristics of piezoelectric sensor and research of its application. In *2009 18th IEEE International Symposium on the Applications of Ferroelectrics*, pages 1–4, 2009.

[18] C. Lu, J. Huang, Z. Lan, and Q. Wang. Bed exiting monitoring system with fall detection for the elderly living alone. In *2016 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 59–64, 2016. doi: 10.1109/ICARM.2016.7606895.

[19] H. Madokoro, N. Shimoi, and K. Sato. Unrestrained sensors using piezoelectric elements for bed-leaving prediction. In *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pages 1599–1604, 2013. doi: 10.1109/ICCAS.2013.6704185.

[20] Momo Medical. URL https://www.momomedical.com/.

[21] K. Nakasho, H. Madokoro, and N. Shimoi. Implementation of a vital signs monitoring system in combination with a bed-leaving detection system. In *2016 IEEE/SICE International Symposium on System Integration (SII)*, pages 290–295, 2016. doi: 10.1109/SII.2016.7844013.

[22] S. Nukaya, T. Shino, Y. Kurihara, K. Watanabe, and H. Tanaka. Noninvasive bed sensing of human biosignals via piezoceramic devices sandwiched between the floor and bed. *IEEE Sensors Journal*, 12(3):431–438, 2012.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[24] D. Sueaseenak, N. Namjirachot, and K. Sukkit. Accelerometer-based angle measurement system with application in hospital bed. In *2015 8th Biomedical Engineering International Conference (BMEiCON)*, pages 1–4, 2015.

[25] T. Umetani, M. Ishii, Y. Tamura, N. Saiwaki, and K. Yokoyama. Change detection of sleeping conditions based on multipoint ambient sensing of comforter on bed. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4997–5001, 2018. doi: 10.1109/EMBC.2018.8513477.

[26] Franciscus Gasthuis Vlietland. Duizeligheid en vallen door lage bloeddruk, September 2016. URL https://www.franciscus.nl/uploads/folder_pdf/duizeligheid_en_vallen_door_lage_bloeddruk.pdf.

[27] H. Yamaguchi, H. Nakajima, K. Taniguchi, S. Kobashi, K. Kondo, and Y. Hata. Fuzzy detection system of behavior before getting out of bed by air pressure and ultrasonic sensors. In *2007 IEEE International Conference on Granular Computing (GRC 2007)*, pages 114–114, 2007.

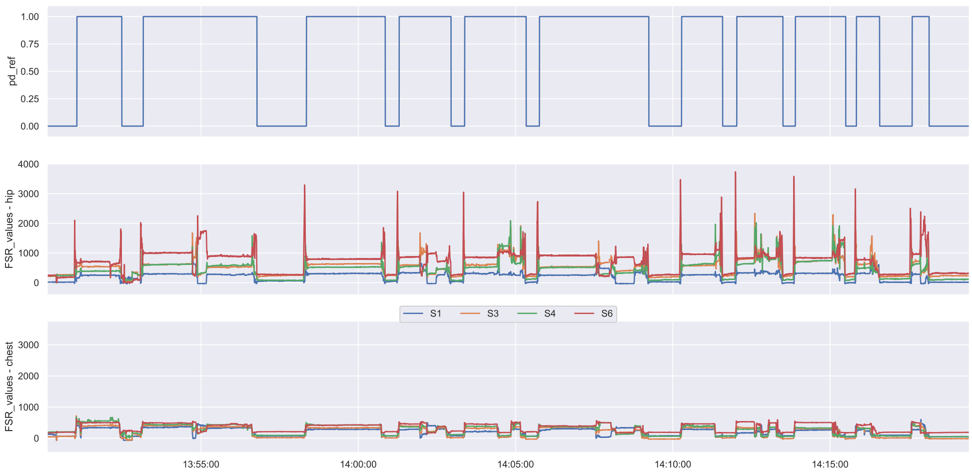
[28] F Schwenk M Mills N Gwynn J D Clemson L Zhang W, Low L. Review of gait, cognition, and fall risks with implications for fall prevention in older adults with dementia. *Dementia and Geriatric Cognitive Disorders*, 48:17–29, 2019. doi: 10.1159/000504340.

[29] Martini Ziekenhuis. Bloeddrukdaling bij ouderen bij het opstaan of na het eten. URL https://www.martiniziekenhuis.nl/media/p1tnpgl3/1495955-bloeddrukdaling-bij-ouderen.pdf.

A

Visualisation of data & matrices

A.1. Further data research

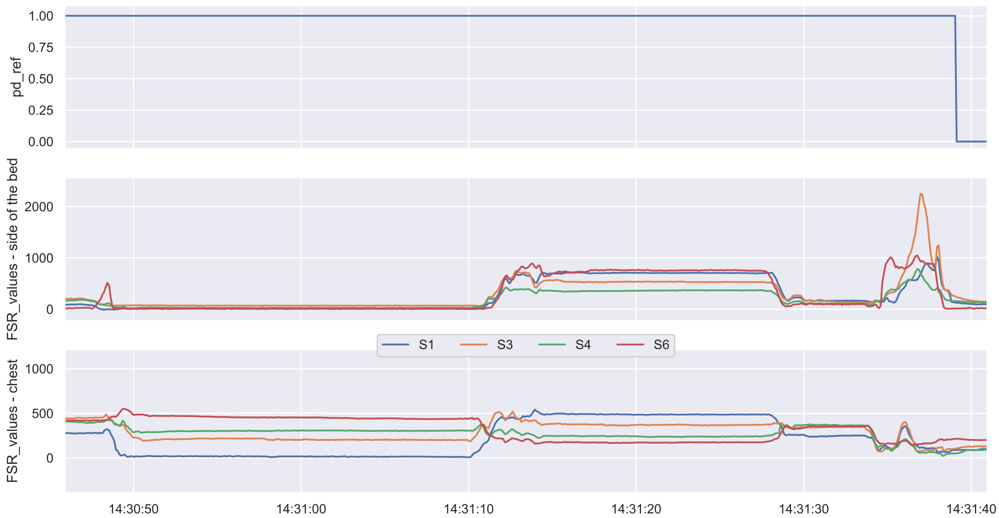


(a) Results of test setup 1 as in Figure 6.1a

(b) Results of test setup 2 as in Figure 6.1b

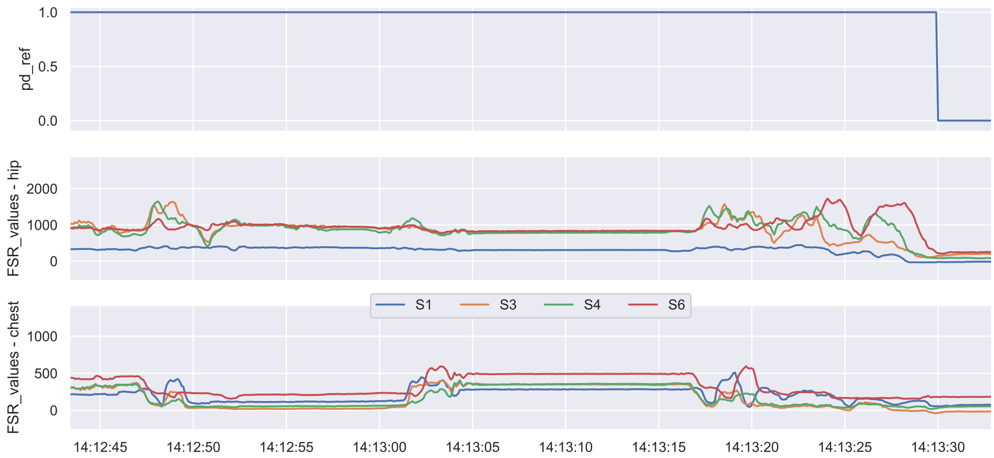
Figure A.1: The FSR values of both sensor plates and the reference signal

(a) Setup 1: Tossing and turning, then leaving the bed

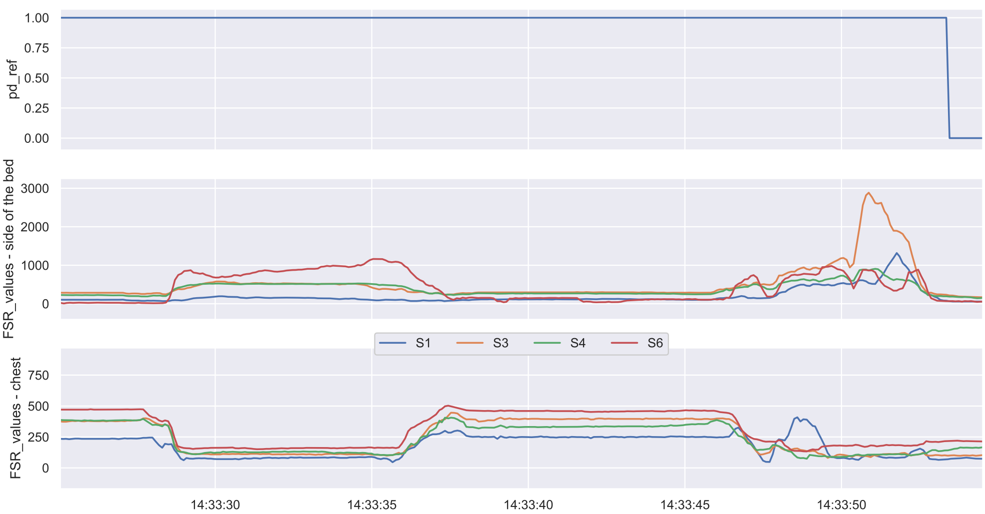


(b) Setup 2: Tossing and turning, then leaving the bed

Figure A.2: Data frames of tossing and turning before the bed exit

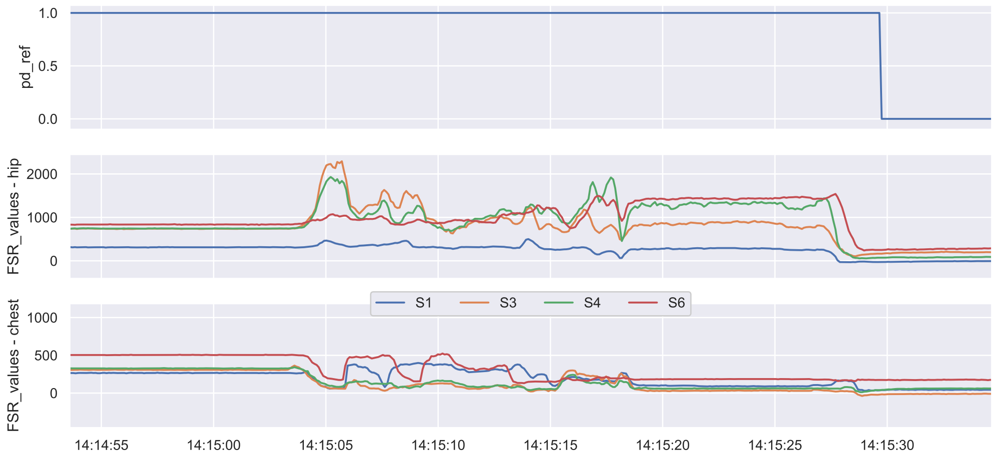


(a) Setup 1: Sitting in bed, lying down then leaving the bed

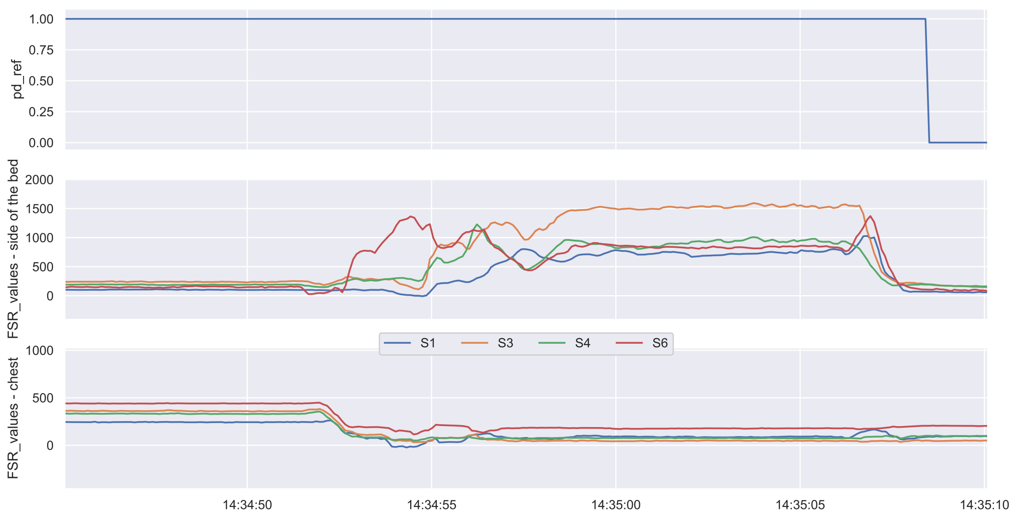


(b) Setup 2: Sitting in bed, lying down then leaving the bed

Figure A.3: Data frames of sitting up in bed before the bed exit

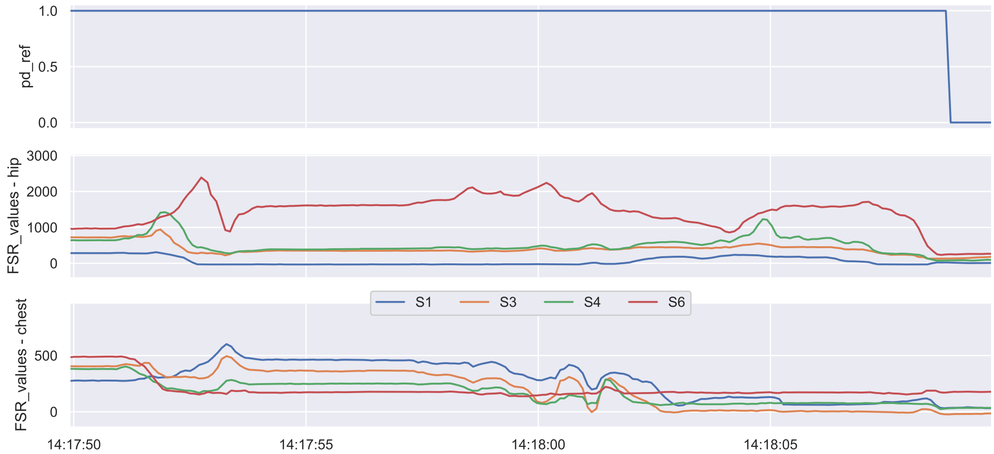


(a) Setup 1: Sitting on the right side of the bed, then leaving the bed

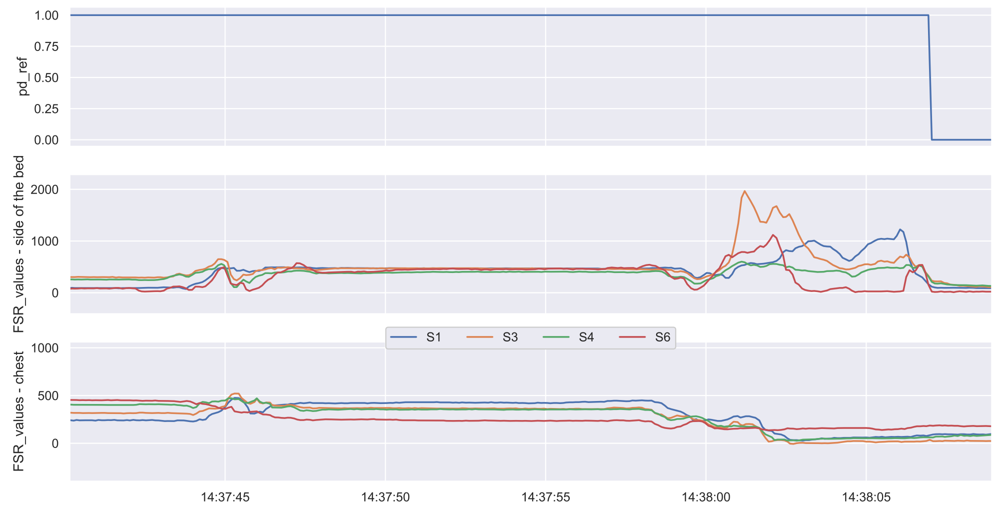


(b) Setup 2: Sitting on the right side of the bed, then leaving the bed

Figure A.4: Data frames of sitting on the bed exit side of the bed before the bed exit

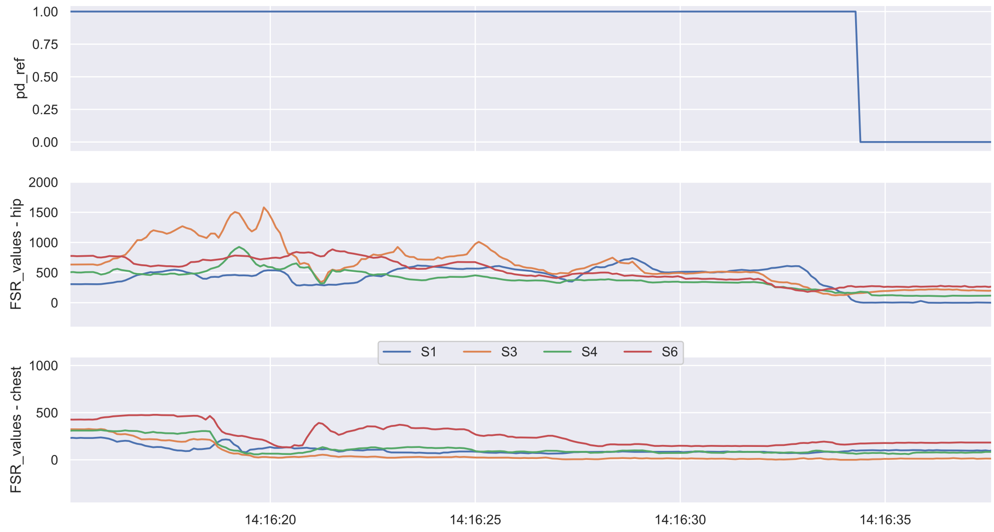


(a) Setup 1: Rolling to the right side of the bed, sitting up then leaving the bed



(b) Setup 2: Rolling to the right side of the bed, sitting up then leaving the bed

Figure A.5: Data frames rolling to the right before the bed exit



(a) Setup 1: Sitting on the left side of the bed, then leaving the bed

(b) Setup 2: Sitting on the left side of the bed, then leaving the bed

Figure A.6: Data frames of sitting on the side of the bed before the bed exit

Figure A.7: Visualisation of a bed exit of another client who is at higher risk of falling

B

Decision tree classifier

B.1. If-else statements classifier

```
if ( f2 <= 160.6523895263672 )
    if ( f3 <= 75.47494125366211 )
        if ( f1 <= 186.35369873046875 )
            if ( f3 <= 25.161998748779297 )
                return [[1170.    0.]]
            else
                if ( f3 <= 25.163557052612305 )
                    return [[0. 1.]]
                else
                    if ( f1 <= 29.633407592773438 )
                        if ( f1 <= 29.62986469268799 )
                            if ( f1 <= 21.75347900390625 )
                                return [[72.  0.]]
                            else
                                if ( f1 <= 22.157745361328125 )
                                    return [[0. 1.]]
                                else
                                    return [[26.  0.]]


                        else
                            return [[0. 1.]]

                    else
                        return [[127.   0.]]



        else
            if ( f3 <= 58.55752754211426 )
                return [[0. 1.]]
            else
                return [[1. 0.]]


    else
        if ( f3 <= 75.65874099731445 )
            return [[0. 1.]]
```

```
else
    if ( f3 <= 437.7112121582031 )
        if ( f1 <= 158.8181610107422 )
            if ( f1 <= 53.52194786071777 )
                if ( f1 <= 52.77543258666992 )
                    if ( f1 <= 26.842658042907715 )
                        return [[32.  0.]]
                    else
                        if ( f1 <= 27.141236305236816 )
                            return [[0.  1.]]
                        else
                            if ( f0 <= 42.347429275512695 )
                                if ( f0 <= 41.71875190734863 )
                                    if ( f1 <= 51.507938385009766 )
                                        if ( f0 <= 24.534720420837402 )
                                            return [[17.  0.]]
                                        else
                                            if ( f0 <= 25.242239952087402 )
                                                return [[0.  1.]]
                                            else
                                                if ( f2 <= 97.70656204223633 )
                                                    if ( f1 <= 33.65706062316894
                                                        if ( f3 <= 135.7443847665
                                                            return [[0.  1.]]
                                                        else
                                                            return [[3.  0.]]

                                                    else
                                                        return [[7.  0.]]

                                                else
                                                    return [[0.  1.]]



                                    else
                                        return [[0.  1.]]

                                else
                                    return [[0.  1.]]

                            else
                                return [[13.  0.]]




                else
                    return [[0.  1.]]

            else
                if ( f3 <= 92.63216018676758 )
                    if ( f3 <= 91.8803482055664 )
                        return [[12.  0.]]
                    else
                        return [[0.  1.]]
```

```
            else
               return [[102.   0.]]


        else
           if ( f1 <= 159.1616973876953 )
              return [[0. 1.]]
           else
              if ( f3 <= 301.79942321777344 )
                 if ( f1 <= 228.26116180419922 )
                    if ( f1 <= 164.85276794433594 )
                       if ( f1 <= 163.070068359375 )
                          return [[7. 0.]]
                       else
                          return [[0. 1.]]

                    else
                       return [[31.  0.]]

                 else
                    if ( f1 <= 231.59429168701172 )
                       return [[0. 1.]]
                    else
                       if ( f0 <= 201.65181732177734 )
                          if ( f1 <= 245.6953125 )
                             return [[0. 1.]]
                          else
                             return [[2. 0.]]

                       else
                          return [[9. 0.]]



              else
                 if ( f3 <= 348.48699951171875 )
                    if ( f2 <= 121.23467636108398 )
                       return [[3. 0.]]
                    else
                       return [[0. 3.]]

                 else
                    return [[7. 0.]]




     else
        if ( f0 <= 140.53024291992188 )
           if ( f1 <= 40.786834716796875 )
              if ( f1 <= 21.559277534484863 )
                 return [[1. 0.]]
              else
                 return [[0. 1.]]

           else
```

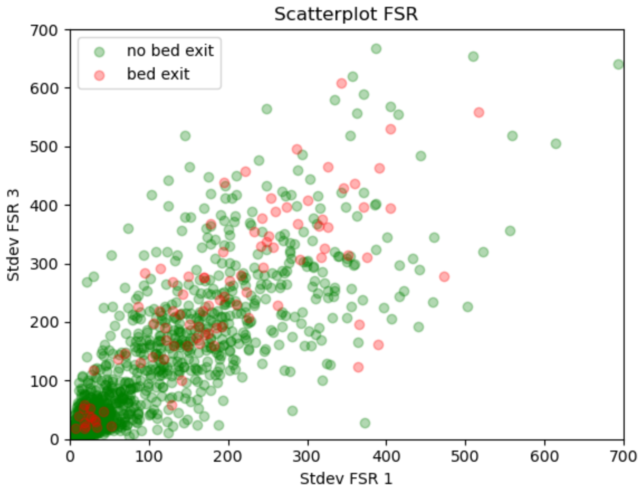
```
                        return [[3. 0.]]

                    else
                        return [[0. 1.]]




else
    if ( f1 <= 309.5775451660156 )
        if ( f3 <= 250.584716796875 )
            if ( f3 <= 141.11216735839844 )
                return [[28.  0.]]
            else
                if ( f3 <= 145.50940704345703 )
                    return [[0. 2.]]
                else
                    if ( f3 <= 212.62289428710938 )
                        return [[25.  0.]]
                    else
                        if ( f3 <= 213.21749877929688 )
                            return [[0. 1.]]
                        else
                            if ( f3 <= 221.66838836669922 )
                                if ( f2 <= 204.5910415649414 )
                                    return [[0. 1.]]
                                else
                                    return [[2. 0.]]

                            else
                                if ( f2 <= 278.424072265625 )
                                    return [[18.  0.]]
                                else
                                    if ( f2 <= 299.91627502441406 )
                                        return [[0. 1.]]
                                    else
                                        return [[3. 0.]]






        else
            if ( f1 <= 138.62994384765625 )
                return [[0. 3.]]
            else
                if ( f2 <= 164.01792907714844 )
                    return [[0. 2.]]
                else
                    if ( f0 <= 170.34017944335938 )
                        if ( f2 <= 286.84654235839844 )
                            if ( f0 <= 127.51759719848633 )
                                if ( f3 <= 261.11012268066406 )
                                    return [[0. 1.]]
```

```
                else
                    if ( f3 <= 530.3535461425781 )
                        return [[18.  0.]]
                    else
                        if ( f3 <= 569.1779174804688 )
                            return [[0.  1.]]
                        else
                            return [[2.  0.]]



            else
                if ( f2 <= 192.50979614257812 )
                    return [[2.  0.]]
                else
                    if ( f2 <= 247.61668395996094 )
                        return [[0.  2.]]
                    else
                        return [[1.  0.]]



        else
            if ( f3 <= 528.2302703857422 )
                if ( f1 <= 213.1251220703125 )
                    if ( f0 <= 119.54912567138672 )
                        return [[1.  0.]]
                    else
                        return [[0.  1.]]

                else
                    return [[0.  6.]]

            else
                return [[4.  0.]]


    else
        if ( f1 <= 164.091796875 )
            return [[0.  1.]]
        else
            if ( f0 <= 226.1895980834961 )
                if ( f0 <= 223.4131088256836 )
                    if ( f1 <= 269.77825927734375 )
                        return [[12.  0.]]
                    else
                        if ( f3 <= 326.8408203125 )
                            return [[0.  1.]]
                        else
                            return [[2.  0.]]


                else
                    return [[0.  1.]]

            else
```

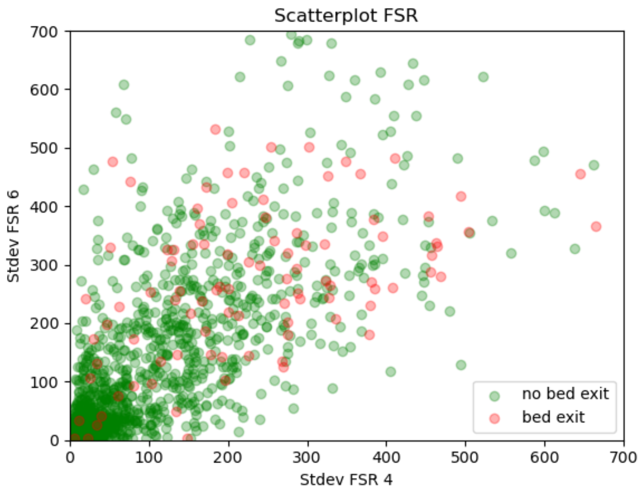
```
                            return [[27.  0.]]




else
    if ( f2 <= 185.00843048095703 )
        return [[0.  5.]]
    else
        if ( f0 <= 215.98397064208984 )
            if ( f2 <= 459.0540008544922 )
                return [[26.  0.]]
            else
                if ( f2 <= 468.6257629394531 )
                    return [[0.  1.]]
                else
                    return [[2.  0.]]


        else
            if ( f1 <= 500.7528991699219 )
                if ( f2 <= 449.9187774658203 )
                    if ( f2 <= 400.23060607910156 )
                        if ( f2 <= 255.54561614990234 )
                            if ( f1 <= 311.11578369140625 )
                                return [[0.  1.]]
                            else
                                return [[8.  0.]]

                        else
                            if ( f2 <= 271.97584533691406 )
                                return [[0.  4.]]
                            else
                                if ( f2 <= 374.7024383544922 )
                                    if ( f2 <= 335.41558837890625 )
                                        if ( f0 <= 336.703369140625 )
                                            if ( f0 <= 316.2476043701172 )
                                                if ( f0 <= 276.3409881591797 )
                                                    if ( f3 <= 254.46609497070312 )
                                                        if ( f3 <= 206.008598327636
                                                            return [[0.  1.]]
                                                        else
                                                            return [[2.  0.]]

                                                    else
                                                        return [[0.  2.]]

                                                else
                                                    return [[3.  0.]]

                                            else
                                                return [[0.  2.]]

                                        else
```

```
                                return [[3. 0.]]

                         else
                            return [[5. 0.]]

                     else
                        if ( f1 <= 439.7502899169922 )
                            if ( f0 <= 258.4779586791992 )
                                return [[0. 1.]]
                            else
                                if ( f0 <= 350.1348419189453 )
                                    return [[2. 0.]]
                                else
                                    return [[0. 1.]]


                         else
                            return [[0. 2.]]




         else
            return [[6. 0.]]

     else
        if ( f3 <= 386.45782470703125 )
            return [[0. 4.]]
        else
            if ( f3 <= 664.6346282958984 )
                return [[2. 0.]]
            else
                return [[0. 1.]]



 else
    if ( f3 <= 404.4341583251953 )
        return [[9. 0.]]
    else
        if ( f3 <= 469.0520324707031 )
            return [[0. 2.]]
        else
            return [[3. 0.]]
```

C
Plots of features



(a) Scatter plot of the standard deviation of the FSR sensors 1 and 3 while weighting the past time frames



(b) Scatter plot of the standard deviation of the FSR sensors 4 and 6 while weighting the past time frames

Figure C.1: Scatter plots of the standard deviations of the FSR sensors while weighting the past time frames

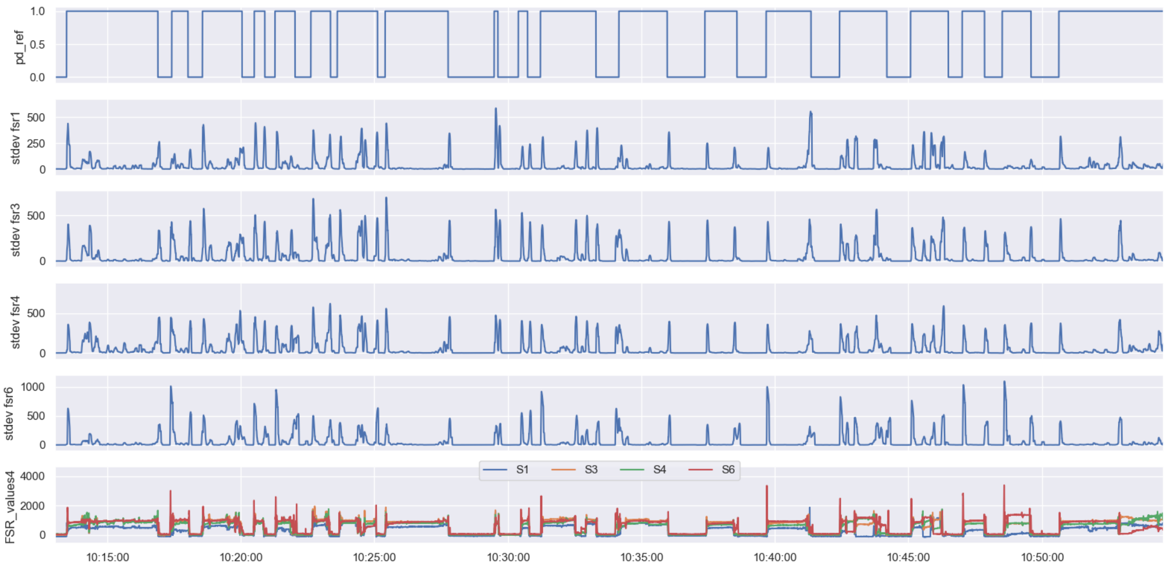


Figure C.2: The standard deviation of the FSR sensors, the reference signal and the raw FSR data

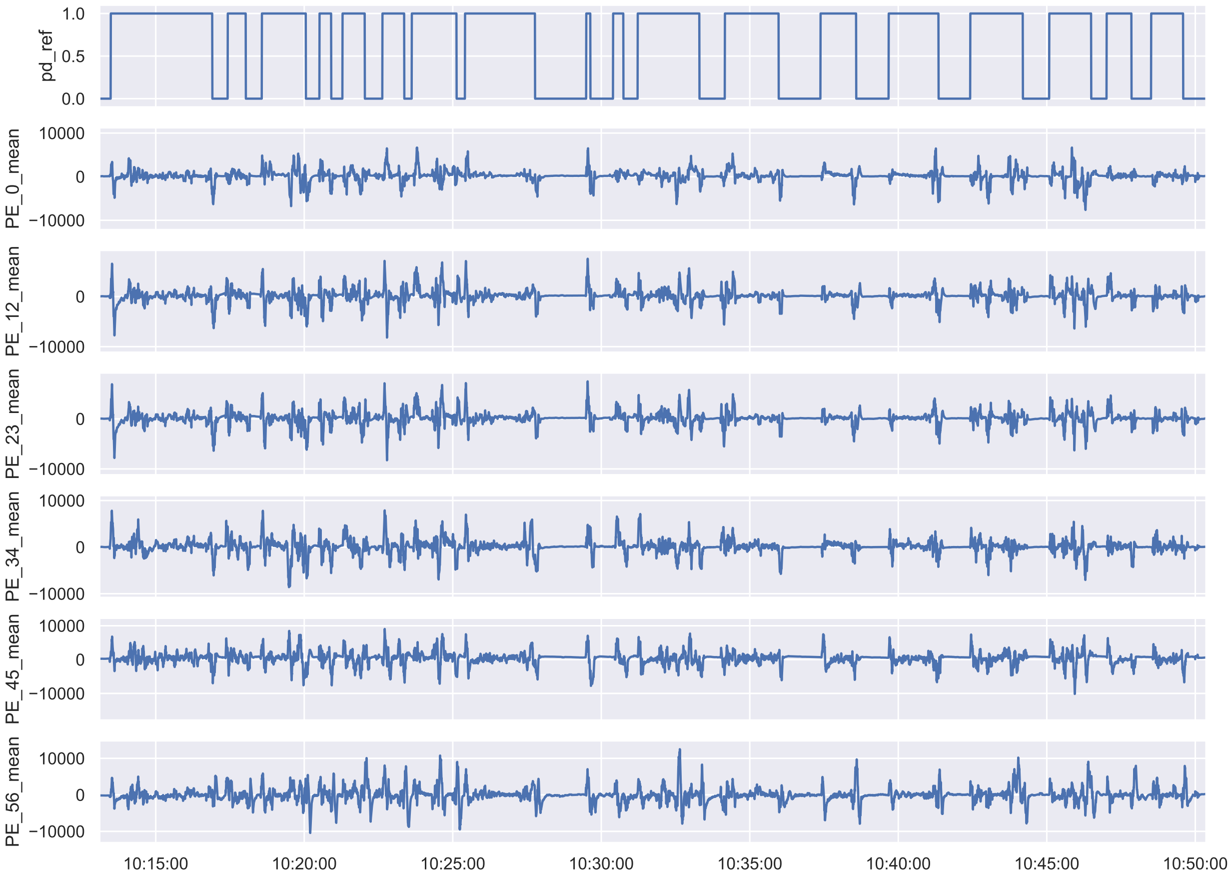


Figure C.3: The mean of the PE sensors and the reference signal

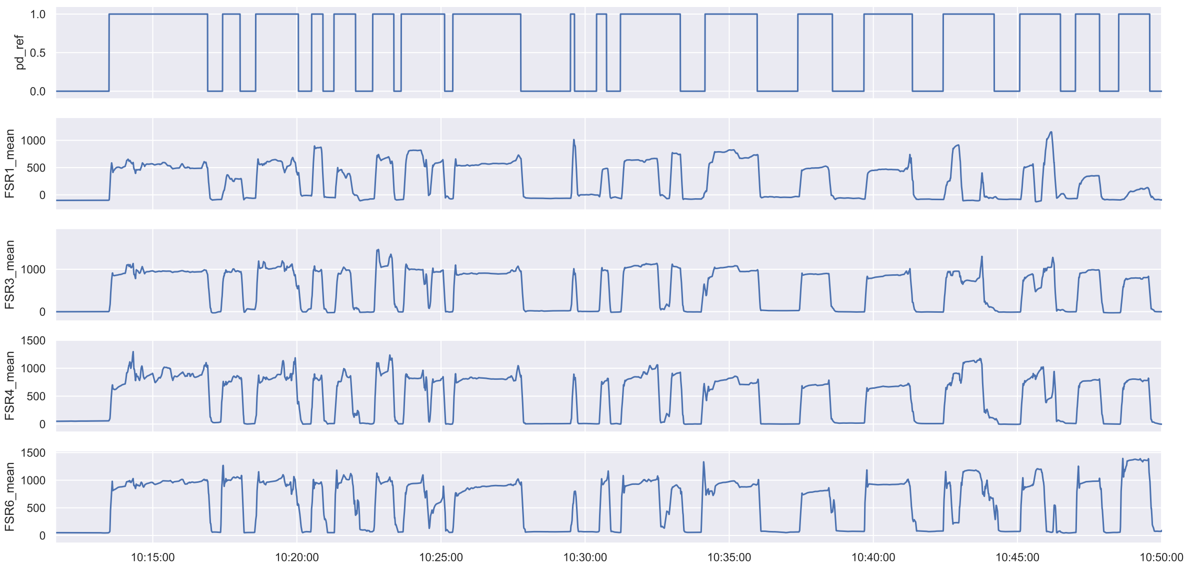


Figure C.4: The mean of the FSR sensors and the reference signal

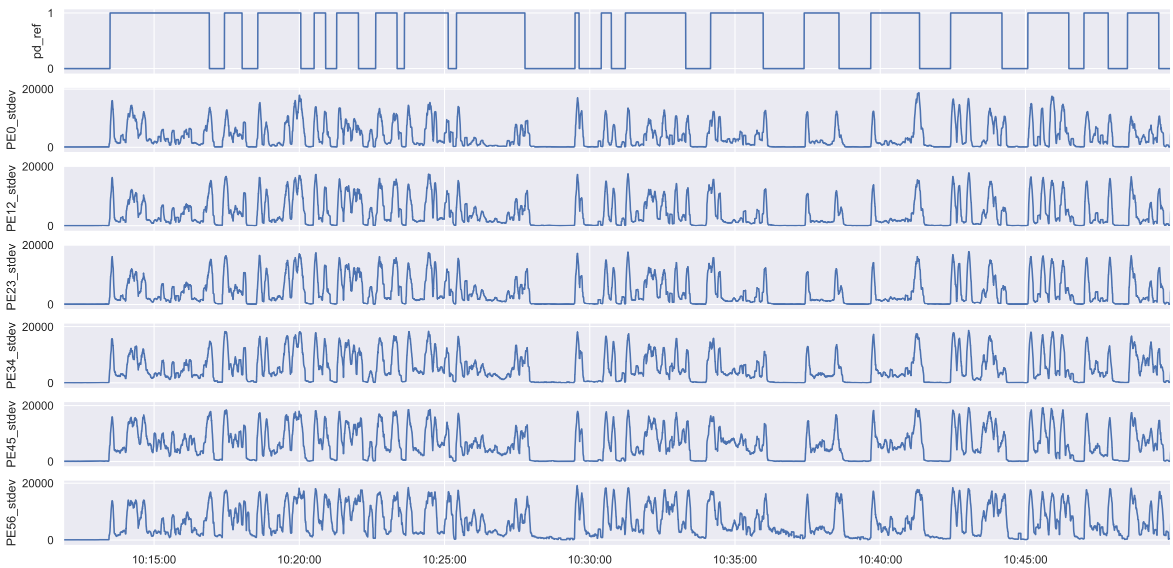


Figure C.5: The standard deviation of the PE sensors and the reference signal

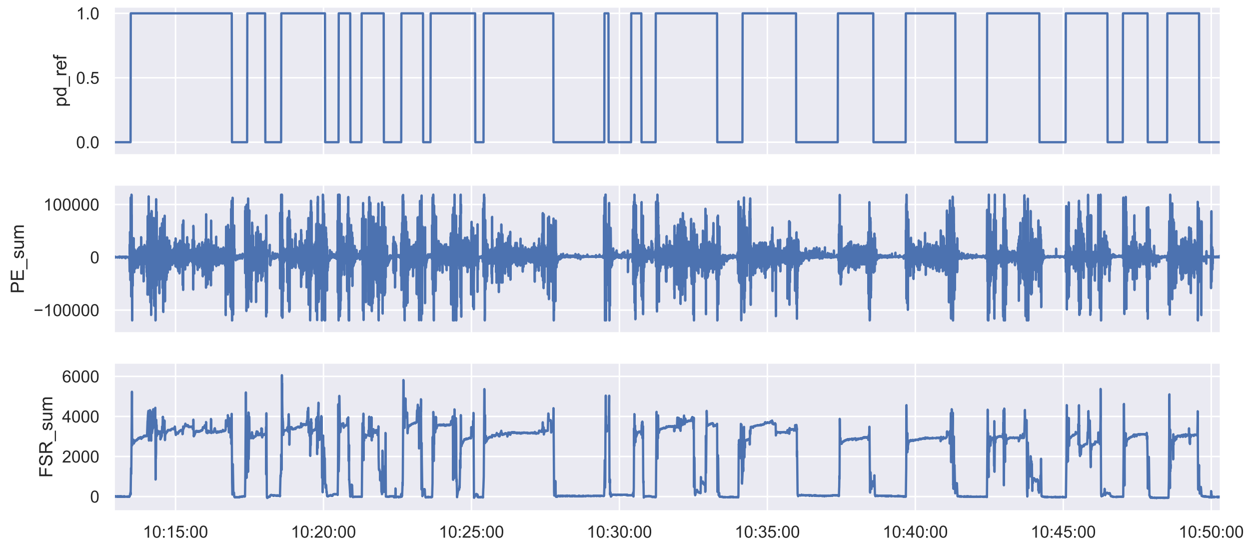


Figure C.6: The sum of the PE sensors and FSR sensors and the reference signal

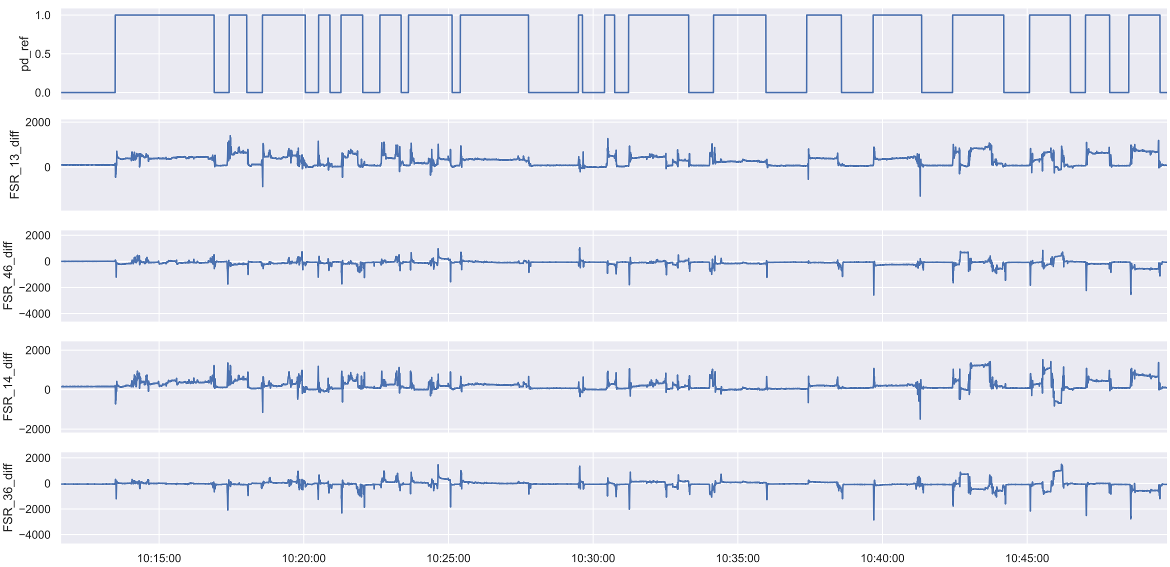


Figure C.7: The differences between certain FSR sensors and the reference signal

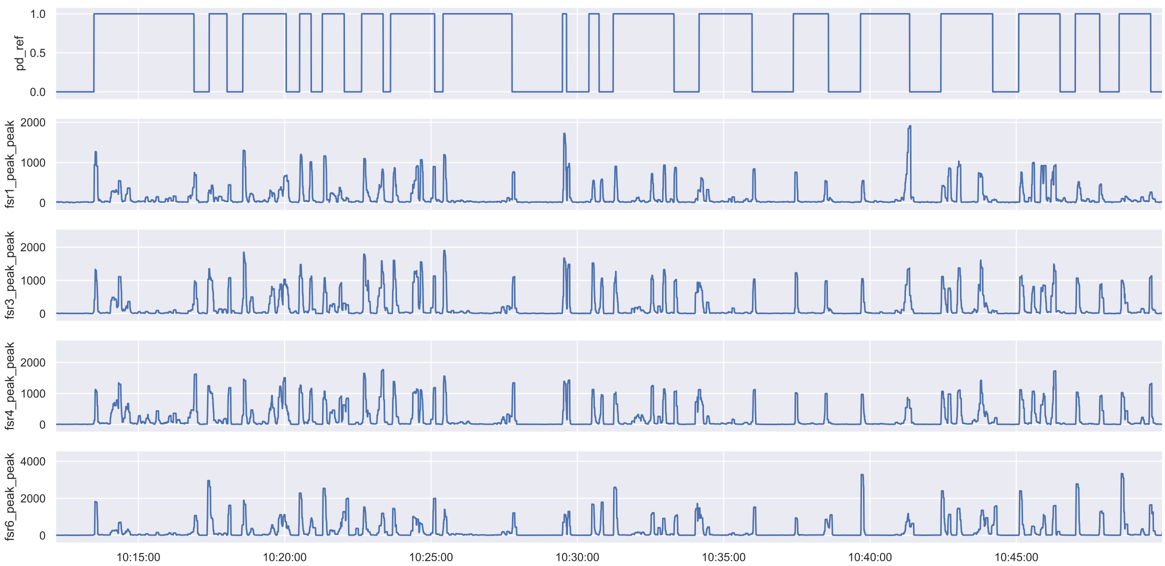


Figure C.8: The peak to peak values of the FSR sensors and the reference signal