



**University of
Zurich**^{UZH}

Comparison of Feature Engineering Methods and Classifiers for Recognizing Physical Activity Types in Older Adults Using Real-Life IMU and GPS Data

GEO 511 Master's Thesis

Author

Yuman He
17-725-334

Supervised by

Dr. Eun-Kyeong Kim
Hoda Allahbakhshi

Faculty representative

Prof. Dr. Robert Weibel

30.04.2021

Department of Geography, University of Zurich

Abstract

Physical Activities (PA) are crucial for human beings to stay healthy both physically and mentally. The physical activities of older adults show different characteristics than that of other age groups, such as lighter intensities and lower speeds. The MOASIS data is large-scale real-life mobility data collected from older adults in Switzerland. In this paper, IMU and GPS dimensions of MOASIS data are used to study the physical activity classification of the older population in real-life conditions. This paper focuses on feature engineering for machine learning methods, including feature calculation, feature extraction, and feature selection. First of all, this paper does a literature review of some of the papers under this theme, and summarizes the research gaps within this topic. The research gaps include: the application and comparison of dimension reduction and machine learning methods on such a real-life dataset focused on this specific age group, the application of GPS data for feature calculation in PA recognition, distinctive features extraction for PA types of older adults, the influence of validation methods on results of machine learning methods. Targeting the above research gaps, this paper puts forward three research questions: the comparison of different machine learning and dimension reduction methods, the comparison of the results of their application on this dataset, the impact of different dimensions of sensor data on the classification results. The results show that first, the most commonly used PCA feature extraction method can indeed improve the results of the KNN classifier in this data to a large extent, but it cannot help in improving the results of the unsupervised classifier Kmeans, which generally performs poorly in PA recognition. Second, Extra-tree performs best when considering the balance between time and accuracy among the classifiers compared. And the Recursive Feature Elimination method (RFECV) has the highest accuracy among the filter, wrapper and embedded feature selection methods based on the Extra-tree classifier. However, the differences in accuracy among the three methods are tiny. In addition, this paper concludes that the two validation methods compared (stratified k-fold validation and holdout validation) may affect the selection of hyper-parameters in model training. Finally, the feature importance ranking by different feature selection methods and the distinctive features for different PA types based on this dataset are also presented. For future studies, this paper suggests that more attention should be paid to the application of different sensor dimensions in PA recognition. Moreover, more fine hyper-parameter adjustment of different models should be investigated.

Acknowledgements

I would like to express my appreciation to people who helped me with this thesis:

- Prof. Dr. Robert Weibel for the coordination of the whole process.
- Hoda Allahbakhshi and Dr. Eun-Kyeong Kim for the supervision.
- Dr. Oliver Burkhard for the Shiny-application for data manual labelling.

Table of Contents

Acknowledgements	2
Table of Contents	3
List of Figures	5
List of Tables	8
1 Introduction	10
1.1 Motivation.....	10
1.2 Aims of this thesis.....	10
1.3 Research Questions.....	11
1.4 Structure of the thesis.....	12
2 Background	14
2.1 Introduction.....	14
2.2 Sensors.....	14
2.3 Datasets.....	15
2.4 Machine learning and deep learning methods.....	16
2.4.1 Supervised learning.....	16
2.4.2 Unsupervised learning.....	17
2.4.3 Deep learning.....	18
2.4.4 Dimension Reduction.....	19
2.5 Model Validation and Comparison.....	21
2.5.1 Validation methods.....	21
2.5.2 Model evaluation metrics.....	22
2.6 Research gaps.....	23
3 Data	25
3.1 introduction.....	25
3.2 MOASIS.....	25
3.3 Labelling of MOASIS data.....	27
4 Methods	32
4.1 Pre-processing.....	32
4.2 Segmentation.....	33
4.3 Feature calculation.....	35
4.3.1 Accelerometer features.....	35
4.3.2 GPS features.....	39
4.4 Dimension reduction.....	40
4.4.1 Feature transformation.....	40
4.4.2 Feature selection.....	41
4.5 Classifiers.....	46
4.5.1 Kmeans.....	46
4.5.2 SVM.....	46

4.5.3 Linear Discriminant Analysis.....	47
4.5.4 Decision tree.....	48
4.5.5 Extra tree.....	49
4.5.6 Random forest.....	50
4.5.7 K-Nearest Neighbours.....	51
5 Results.....	52
5.1 preprocessing.....	52
5.2 Feature calculation.....	52
5.3 Segmentation.....	56
5.4 Dimension Reduction.....	57
5.4.1 Feature transformation.....	58
5.4.2 Feature Selection.....	64
5.5 Comparison of GPS and Accelerometer sensors.....	80
6 Discussion.....	82
6.1 RQ2 - Differences in performances by classifiers and dimension reduction methods.....	82
6.1.1 Dimension reduction methods with K-means and KNN.....	82
6.1.2 Feature selection methods with Extra Tree Classifier.....	83
6.1.3 Comparison of feature selection methods.....	84
6.1.4 Comparison of validation methods.....	85
6.2 RQ-3 Features from the two sensors and distinctive features.....	85
6.2.1 Feature comparison from two sensors.....	85
6.2.2 Features for different activity types.....	86
7 Outlook.....	88
Literature.....	89
8 Appendices.....	94
8.1 Appendix 1: Tables and Figures.....	94
8.2 Appendix 2: Code.....	97

List of Figures

Figure 2.1 Four-class SVM classifier (Zheng et al., 2015).....	17
Figure 2.2 Nested cross Validation (Outer loop: 5-fold CV, Inner loop: 2-fold CV) (Grootendorst, 2019)	22
Figure 3.1 uTrail tracker sensors in MOASIS study.....	25
Figure 3.2 Valid hours distribution of participants in MOASIS study.....	26
Figure 3.3 Valid approximate walking hours distribution of the 14 top participants.....	27
Figure 3.4 the Interface for the Labelling Application Shiny App	27
Figure 3.5 Mapping participants locations to high resolution DEM	28
Figure 3.6 Comparison of indoor and outdoor walking	29
Figure 3.7 Ascending segment (a) and the increasing altitudes displayed in the console (b)	29
Figure 3.8 Other activities.....	30
Figure 3.9 Different activities labelled in different colours.....	31
Figure 3.10 the amount of the 7 types of physical activities.....	31
Figure 4.1 PA classification steps	32
Figure 4.2 low pass filter (Left) result in gravitational acceleration, high pass filter (Right) result in body acceleration (Bayat et al., 2014).....	33
Figure 4.3 Activity-wise Boxplot for accelerometer and gyroscope signals (Dehzangi et al., 2018)	35
Figure 4.4 Pseudo code of ReliefF algorithm (Robnik-Šikonja et al. 2003).....	43
Figure 4.5 Hyperplane that maximizes the margin in SVM (Cortes et al., 1995).....	46
Figure 4.6 Hyperplane that maximizes the margin in Kernel SVM (Cortes et al., 1995).....	47
Figure 4.7 feature transition before and after LDA in PA recognition (Khan et al., 2010).....	48
Figure 4.8 Decision Tree example of what to do when when different situation happens in weather (Patel et al., 2018).....	48
Figure 4.9 Decision tree generated from PA data (Yang, 2009).....	49
Figure 4.10 Illustration of Extra Tree Classifier working (Bhati et al., 2020).....	50

Figure 5.1 Acceleration signal before and after applying the low-pass filter.....	52
Figure 5.2 GPS speed before (left) and after (right) the recalculation	52
Figure 5.3 Total acceleration signal comparison for different activity types.....	53
Figure 5.4 Total acceleration box-plot comparison for different activity types.....	53
Figure 5.5 Speed comparison for different activity types	54
Figure 5.6 Speed box-plot comparison for different activity types.....	54
Figure 5.7 Activity-wise peaks and peak heights.....	55
Figure 5.8 Activity-wise peak width and time intervals.....	55
Figure 5.9 the activity-wise cross-correlation results of the total acceleration	56
Figure 5.10 classification results without feature selection in the window size of 180 (without overlap (left), with overlap(right)).....	56
Figure 5.11 classification results without feature selection in the window size of 90 (without overlap (left), with overlap (right)).....	57
Figure 5.12 Confusion matrix for KNN without PCA (Stratified-10-fold).....	61
Figure 5.13 Confusion matrix for KNN with PCA.....	62
Figure 5.14 Comparison of confusion matrix for KNN with (Right) and without (Left) PCA....	62
Figure 5.15 Confusion matrix of KNN with PCA by holdout validation.....	63
Figure 5.16 Confusion matrix of KNN with PCA by Stratified 10-fold (Right) and holdout (Left) validation.....	63
Figure 5.17 Confusion matrix for Extra Tree with out ReliefF (Stratified 10-fold)	66
Figure 5.18 Comparison of confusion matrix for Extra Tree with (Left) and without (Right) ReliefF by stratified 10 fold.....	67
Figure 5.19 Comparison of confusion matrix for Extra Tree with (Left) and without (Right) ReliefF by holdout validation.....	67
Figure 5.20 Feature importance comparison of Extra Tree Classifier before (Left) and after (Right) ReliefF feature selection.....	68
Figure 5.21 Confusion matrix by Extra Tree Classifier with the feature set selected by GA (a) and the whole feature set (b)	73
Figure 5.22 Feature importance comparison of the Extra Tree Classifier trained by the whole feature set (a) and GA selected feature set (b)	73

Figure 5.23 Cross validation scores with minimum number of features by RFECV with Extra Tree Classifier	77
Figure 5.24 Confusion matrix by Extra Tree Classifier with the feature set selected by RFE (a) and the whole feature set (b).	77
Figure 5.25 Feature importance comparison of the Extra Tree Classifier trained by the whole feature set (a) and RFE selected feature set (b).....	78
Figure 5.26 classification results from GPS sensor (right) and Accelerometer (left) data by different classifiers.....	81
Figure 8.1 RFECV feature selection scores with the the number of features selected.....	97

List of Tables

Table 2.1 Common open-source PA classification datasets.....	15
Table 2.2 Most Accurate Classifiers and Feature Selection Methods.....	20
Table 2.3 validation methods used in studies.....	22
Table 3.1 threshold based on GPS speed values for the 4 activities.....	29
Table 4.1 Sampling frequencies and window types in literature.....	34
Table 4.2 Most common features in the time and frequency domains for accelerometer sensor.	35
Table 4.3 Statistical features with brief descriptions (Zhang et al. 2011).....	36
Table 4.4 Physical features with brief descriptions (Zhang et al. 2011)	36
Table 4.5 The number of features calculated and used in the end for classification.....	37
Table 4.6 Distinctive features to separate different types of physical activities (Bao et al., 2004; Zhang et al., 2011; Zheng et al., 2015).....	38
Table 4.7 Accelerometer features calculated for analysis.....	38
Table 4.8 GPS features applied.....	39
Table 4.9 Dimension reduction category.....	40
Table 4.10 PCA process.....	41
Table 4.11 Feature selection methods.....	42
Table 4.12 Filter methods	42
Table 4.13 pseudocode for Genetic Algorithm	44
Table 4.14 pseudocode for Recursive Feature Elimination.....	45
Table 4.15 Decision Tree Algorithms.....	49
Table 5.1 Classification Results for PCA combined with K-means.....	58
Table 5.2 Classification results for PCA and KNN	59
Table 5.3 Classification results for PCA and KNN (stratified 10-fold).....	60
Table 5.4 Classification results for PCA and KNN (holdout validation).....	60
Table 5.5 Classification results for ExtraTree and ReliefF (stratified 10-fold).....	64
Table 5.6 Classification results for ExtraTree and ReliefF (holdout validation).....	65

Table 5.7 Important features for different classes by Extra Tree Classifier without feature selection.....	69
Table 5.8 important features for different classes by Extra Tree Classifier with ReliefF feature selection.....	70
Table 5.9 Top 20 feature categories comparison before and after ReliefF.....	71
Table 5.10 GeneticSelection with Extra Tree Classifier.....	72
Table 5.11 Top 20 feature categories comparison before and after genetic algorithm.....	74
Table 5.12 important features for different classes by Extra Tree Classifier with GA feature selection.....	75
Table 5.13 RFECV with Extra Tree Classifier.....	76
Table 5.14 Top 20 feature categories comparison before and after RFE algorithm.....	78
Table 5.15 important features for different classes by Extra Tree Classifier with RFE feature selection	79
Table 5.16 Feature selection methods and results.....	80
Table 8.1 GeneticSelection with chosen features by Extra Tree Classifier	94
Table 8.2 RFECV with chosen features by Extra Tree Classifier	96
Table 8.3 Overview of scripts containing important steps for the PA classification.....	97

1 Introduction

1.1 Motivation

Physical activity (PA) is widely known as a key factor for human health in all age groups. For older adults, a more active than sedentary lifestyle has remarkable positive consequences which can reduce their risks of having certain chronic diseases (Chodzko-Zajko 2014).

Physical activity levels have since many years been used as the common measurement of the necessary daily and weekly physical activity for the population. Besides duration and frequency, physical activity levels are described by types and intensities - not all physical activities have an equivalent impact on health. This makes physical activity recognition a crucial area in the research of life quality enhancement. Other than health support, PA recognition is also the core assistive technology in contexts such as Smart homes (Francisco et al., 2015), skill assessment (Kranz et al., 2013), etc.

PA recognition is primarily a classification problem. There are two common ways to record PAs for detection, namely by wearable sensors or cameras. The recorded signals or videos are feed into the classification process as the input data, and the PA patterns are produced as the output data from the process. The accelerometer that measures body acceleration in different directions is the most common sensor for physical activity measurement. IMU sensors are lightweight and portable devices that contain other sensors measuring other aspects of motion besides accelerometer, including gyroscope for rotation and magnetometer for direction.

Physical activity recognition using wearable sensors has received growing attention along with the pervasiveness and miniaturization of wearable devices, such as Smartphones, Smart bands, etc. In this field, as the state-of-the-art technologies, machine learning and deep learning methods are becoming the mainstream approach than other methods, such as the cut-point method, which segments signals by certain determined thresholds. Machine learning methods can automatically detect PA patterns with high recognition performance.

1.2 Aims of this thesis

Given the benefits of PA for older adults described in section 1.1, this study aims to contribute to the healthy aging of older adults by investigating the PA classification on the dataset MOSIS that widely records real-life mobility of older individuals in Switzerland with available IMU and GPS data. More specifically, In this work, different classification models in conjunction with feature selection and feature extraction (feature transformation) algorithms are to be explored and and evaluated to select a high accuracy and low cost process with a compact and robust discriminative feature space for a large real-life dataset PA collected from older adults, with a focus on walking activities detection. The contributions of the thesis are the following:

- This thesis provides a literature review of some of the feature selection methods with classifiers in PA analysis that are applied on different types of datasets.
- This thesis applies and compares the most common classifiers and feature selection methods on a large real-life dataset to give an idea of what's the advantages and disadvantages of those methods on a large and real-life condition dataset other than the laboratory settings.
- This thesis researches the combination of different sensors in PA recognition, more specifically, the GPS sensor, in PA types detection with a focus on walking types. Based on the location information provided by the GPS sensor, this thesis incorporates contextual information for PA classification. Besides, this research inspects distinctive features selected by different feature selection methods in the recognition of PA types. Moreover, the performance differences of the models tested by different validation manners are inspected.

1.3 Research Questions

As illustrated by the above section about the contributions of this paper, on the basis of the research gaps that are expanded in the next Chapter, the research questions of this thesis are formed as follows:

- 1) What are the most common dimension reduction methods and classifiers in PA recognition? How do they influence the classification results? What criteria should be taken into consideration when comparing different algorithms and classifiers?

Supervised machine learning classifiers are the most common automatic classifiers to detect physical activity types. In general, supervised classifiers exhibit relatively high performance. In supervised learning, random forest, Extra-tree, and SVM (support vector machine) are considered the most accurate models by many studies (Zhang et al., 2011; Peterek et al., 2014; Zheng et al., 2015; Allahbakhshi et al., 2019; Allahbakhshi et al., 2020). Dimension reduction methods consist of feature transformation and feature selection methods. Common feature transformation methods are PCA, LDA, etc. Common feature selection methods can be categorized as the filter, wrapper, and embedded types (Chandrashekar et al., 2013). In general, filter methods are faster and simpler, while giving lower performance. Wrapper methods have higher computational costs, but give better results. Embedded methods can reduce time cost compared to wrapper methods, and can also maintain good performance. Accuracy is the most common metric applied literature in the comparison of classifiers and dimension reduction methods. Time cost is also sometimes presented in some papers. Other performance metrics, such as recall, precision, f1-score are sometimes given as well.

- 2) What are the differences in the results when applying the most common PA classifiers and feature selection methods on the MOASIS dataset?

From the research by Ordóñez et al. (2016), the classification performances of different classifiers and feature selection methods have different results on different datasets. In other words, the feature selection method and classifier that generate high

performance on one dataset might not perform that well on another dataset. Nevertheless, the extra tree and random forest are the most accurate classifiers in a large proportion of studies. However, the computational cost is considered rather high by random forest. In this respect, deep learning classifiers without manually feature design process have also relatively low computational costs when running (Chandrashekar et al., 2014). In terms of feature selection methods, filter methods are in general faster than wrapper methods, while wrapper methods provide better accuracy improvement results. This thesis aims to find out the best model and method for a real-life dataset for older adults. This suggests that the model are supposed to be suitable for a large dataset in the computational sense, as well as good at detecting activities performed most by older adults, especially the recognition of different walking activities.

- 3) What are the differences in the results when applying extra sensors besides the common accelerometer sensor on the MOASIS dataset? What are the distinctive features from different dimensions of sensors for different PA activity types for older adults? What are the differences among distinctive features selected by different classifiers and feature selection methods (feature extraction methods reshape the original features and thus are not taken into consideration)?

In general, extra sensors help in improving the accuracy of the classification. For example, an additional gyroscope increased the classification accuracy in PA classification by 15%, an extra magnetometer improves the PA recognition performance to 5% (Ordóñez et al., 2016). From Allahbakhshi et al.'s (2020) study, the GPS sensor helps to increase the model performance as well. This study incorporates more features from the GPS sensor besides the speed attributes from the research of Allahbakhshi et al.(2020). The location attribute creates more features for the GPS sensor (e.g distance), and has been proved to be efficient in deriving trip purpose prediction (Gonzalez et al. 2008, Gong et al. 2014). Therefore, the additional GPS sensor that provides the contextual information for PA classification is expected to improve the classification results to a certain degree.

Distinctive features are more helpful than other features in distinguishing all classes or a part of classes to be detected. For example, in PA types recognition, variances of total acceleration and the three axes signals are vital in separating running, jumping and walking activities ((Bao et al., 2004). And total acceleration mean and total acceleration entropy are the key features in detecting different walking types (walking left, walking right, walking forward/backward, walking stairs) (Zheng et al., 2015). Different classifiers and feature selection methods give different importance score among features (Dehzangi et al.. 2018). Under this theme, a close inspection of distinctive feature differences in PA types study among various classifiers and feature selection algorithms are not common. Nevertheless, this thesis expects to detect more distinctive features in PA types recognition, especially in walking activities. And different feature selection methods are expected to be helpful in selecting more distinctive features from different sensors.

1.4 Structure of the thesis

After the introduction into this theme, this thesis is structured as follows. Chapter 2 provides the related background about the works done on this topic. In this Chapter,

the research gaps based on the previous work are given. Chapter 3 describes the data employed in this thesis, where the manual labelling process and the characteristics of the labelled data are presented. Chapter 4 illustrates the methods applied in this analysis, where the data processing procedure, different classifiers and feature selection methods are introduced. Chapter 5 presents the results obtained from applying the designed methods to the data. In this part, classification results by different feature selection methods combined with some classifiers are presented. Also, feature importance of the two sensors is discussed with respect to the features' contribution in classification. Chapter 6 puts the results under the research questions, and gives a critical evaluation and discussion of the performed analysis. Chapter 7 concludes the findings in this thesis and gives an outlook for future work in this topic.

2 Background

2.1 Introduction

PA recognition studies have the focus on either activity intensity or activity type recognition. Activity intensity recognition is mostly related to the concept of energy expenditure, represented by categories such as inactivities, moderate/vigorous activities (Kuppevelt et al., 2019). Activity types are more specific, and can be further grouped into posture and motion, such as sitting, standing, running and cycling (Huynh et al., 2005). PA recognition in this thesis refers to activity types recognition considering PA types recognition is a more fine category. Also, activity types applications are more generalizable in other scenarios. According to Lara et. al (2013), research in physical activity recognition can be categorized into seven different categories: (1) selection of attributes and sensors; (2) obtrusiveness; (3) data collection protocol; (4) recognition performance; (5) energy consumption; (6) processing; and (7) flexibility. This thesis contributes to the (1), (4) categories. The following background part will be introduced in this vein.

2.2 Sensors

Inertial Measurement Unit (IMU) is a device that measures the uniaxial, biaxial, or triaxial angular velocity and acceleration of an object. IMU can be used to detect and measure acceleration, tilt, impact, vibration, rotation, and multi-degree-of-freedom motion. The "6-axis IMU" refers to a gyroscope and accelerometer are mounted on three orthogonal axes, with a total of six degrees of freedom, to measure the angular velocity and acceleration of an object in three-dimensional space. With an extra magnetometer, the device can measure the heading angle and increase measurement accuracy. This is known as the "9-axis IMU".

Among the sensors, the gyroscope is used to measure the rotation movement of the device itself, but it cannot determine the orientation of the device. The accelerometer is used to measure the force on the device and is good at detecting the movement of the device relative to an external reference, such as the ground. But its measurement of the position of the equipment relative to the ground is not very accurate. Magnetometer is mainly used to measure the direction of the current equipment with its angle to the four directions. In simple words, the gyroscope tells "one turned around," the accelerometer tells "one went a few more meters," and the magnetometer tells "one went west."

Accelerometer sensor is widely employed in physical activity studies. There are plenty of well-established efforts in PA types classification based on accelerometer data (Huynh et al., 2005; Troped et al., 2008; Mannini & Sabatini., 2010; Reiss and Stricker., 2011; Sprint, 2016; Ignatov & Strijov, 2016; Kuppevelt et al., 2019; Allahbakhshi et al., 2019;Allahbakhshi et al., 2020).

Studies have designed and tested various settings with different accelerometer numbers, types (uni-axial (2D), dual axial (2D), and tree-axial (3D)), and positions placed on the body (Reiss and Stricker 2011, Allahbakhshi et al. 2020). In general, a larger number of accelerometers and multiple axial accelerometers can provide better performance as more information can be extracted. The placement of accelerometers also has a significant impact on the recognition of different PA patterns. The most studied positions are the chest, wrists, hips, knees, and feet. However, it is hard to conclude the best position for accelerometer placement given various study designs and objectives. For instance, a sensor placed on the wrist is preferable when trying to detect activities with similar lower-body, but significantly different upper-body movement. Nevertheless, the results from various studies have shown that the recognition of these activities is possible even with just one 3D-acceleration sensor, such as the accelerometer from a smartphone (Ignatov & Strijov., 2016).

A number of them also have included gyroscope data and magnetometer data on the basis of the accelerometer data (Jiang et al., 2015; Ordóñez et al., 2016). In general, more sensors provide higher accuracy in classification (Ordóñez et al., 2016). The results from Zhang et al.'s (2011) research found out that the application of gyroscope and magnetometers data both improves the performance compared to accelerometer data alone. There are studies that have incorporated other sensor types in the recognition of PAs. Study shows that the employment of GPS data also helps in improving the classification results. There are also other auxiliary sensors in PA recognition, such as heart rate sensors, etc.

2.3 Datasets

Table 2.1 shows some of the most popular open-source PA classification datasets with sensor settings and activity types that are applied in studies. The characteristics of the data in the MOASIS study will be discussed in the next Chapter.

Table 2.1 Common open-source PA classification datasets

datasets	Number of activities	Type and number of sensors	Activity types	Number of Participants	literature
PAMAP 2	lying, sitting, standing, ironing, vacuuming, ascending stairs, descending stairs, normal walk, nordic walk, cycling, running, rope jumping	3 IMUs, each with an 3D accelerometer, gyroscope, and magnetometer	12	9	Saez et al. (2015), Baldominos et al. (2017)
The MobiFall dataset	Standing, sitting, lying, stairs up, stairs down, walking	a tri-axial accelerometer and a gyroscope	6	-	Sukor et al. (2018)
USC-HAD	walking (forward, left, and right), walking (upstairs, downstairs), jumping, running, standing, sitting, and sleeping	a 3-axis accelerometer	10	14	Zhang et al. (2015)
UCI:M. Lichman (2013)	Walking up, walking down, walk, sit, stand, lay	IMU	6	40	Dehzangi et al. (2018)
UCI:	Walking, walking upstairs,	a 3-axis	6	30	Peterek et

Anguita (2012)	walking downstairs, sitting, standing, lying	accelerometer, 3 axial angular velocity sensor			al. (2014)
OPPOR TUNIT Y	Walk, Stand, Sit,Lie,Null	7 IMU plus 12 Accelerometers	5	-	Ordóñez et al. (2016)

2.4 Machine learning and deep learning methods

Machine learning as the state-of-art method, has been explored widely in PA recognition (Huynh et al., 2005; Troped et al., 2008; Mannini & Sabatini., 2010; Reiss and Stricker., 2011; Zhang et al., 2011; Peterek et al., 2014; Zheng et al., 2015; Sprint, 2016; Ignatov & Strijov, 2016; Kuppevelt et al., 2019; Allahbakhshi et al., 2019; Allahbakhshi et al., 2020).

2.4.1 Supervised learning

Supervised machine learning classifiers that use the labelled data to train the classifier and automatically detect physical activity types of the test data are the most researched methods. It is the most common automatic classifier to detect physical activity types from accelerometer data. Supervised learning in general exhibit relatively high performance. In supervised learning, random forest, Extra-tree, and SVM (support vector machine) are considered the most accurate models by many studies (Zhang et al., 2011; Peterek et al., 2014; Zheng et al., 2015; Allahbakhshi et al., 2019; Allahbakhshi et al., 2020).

Zhang et al., (2011) designed a human activity recognition framework based on Support Vector Machine (SVM) classifier with a focus on feature selection techniques and achieved an accuracy of 93.1%. The authors designed a new type of features, namely physical features other than the traditional statistical features, based on a sensor fusion manner for PA classification (see Chapter 4.3.3 Feature selection). Three feature selection methods are employed to investigate the best feature sets, and physical features (90% accuracy) are proved to be more effective than statistical features (82% accuracy). In their work, the performance was further improved by 3.8% by extending the single-layer framework of one classifier to a multi-layer framework with multiple classifiers in a hierarchical manner. From the multiple structure framework, how different physical features contribute to the classification accuracy was clearly presented. In their results, SFS (sequential forward selection) (wrapper) method achieved the best performance compared to Relief-F (filter) and single feature classification (SFC) (wrapper) methods. The study indicates that the employment of physical features and multiple-layer classifiers could improve classification accuracy. However, the structure of the hierarchical is designed manually based on expert knowledge, the calculation of most physical features also based on the prior knowledge about the sensor orientation (x-axis points to the ground). Therefore, a more automatic and generalizable framework could be structured in a data-driven approach.

Peterek et al. (2014) compared three supervised classification algorithms: the Linear Discriminant Analysis, the Random Forest, and the K-Nearest Neighbours. Besides, two feature extraction methods were also tested for better classification performance: the Correlation Feature Selection Method and the Principal Component Analysis.

From their analysis, the easiest state for recognition was lying, with both LDA and the RF achieved 100 % of accuracy. The LDA classifier outperformed the RF and KNN classifiers. However, the authors applied the simple holdout validation method, which might result in performance bias. Different validation methods that can reduce performance bias can be implemented to further analyse the models, such as k-fold validation.

Zheng et al. (2015) proposed a hierarchical recognition scheme to classify 10 activities based on Least Squares Support Vector Machine (LS-SVM) and Naive Bayes (NB) algorithm. From a preliminary investigation, a four-layer structure with 5 classifiers was selected out of 4 options considering the small number of classifiers and high average accuracy rate (above 90%). As shown in Figure 2.1, the second layer of the framework distinguished running, jumping, walking and static activities. The third layer recognised subclass activities in the static activities category. Besides, it differentiated 2D and 3D walking activities. The fourth layer recognised the walking activities in subclasses. In this study, only six features were computed and applied in the model, with one pair of features for each classifier. The LS-SVM is an advanced version of the standard SVM that extends traditional SVM for binary non-linear classification problems to multi-classification method. The result showed a promising recognition accuracy. However, the classification accuracies for walking activities are relatively lower, due to the similarity characteristic among different walking patterns.

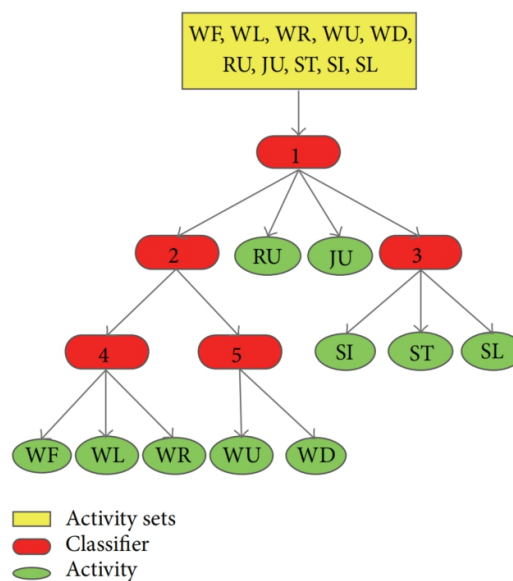


Figure 2.1 Four-class SVM classifier (Zheng et al., 2015)

2.4.2 Unsupervised learning

In PA recognition, unsupervised learning is less common. Unsupervised learning can automatically separate the datasets into clusters that exhibit similar characteristics by their underlying structure. In this way, compared to supervised learning, unsupervised learning has the advantage of without the need for the training dataset with labels as reference. And in real life, labelling the data is a time-consuming and costly process

in supervised learning. However, without the labels, the unsupervised classifiers in general exhibit lower performance than the supervised classifier.

Unsupervised learning is in general not considered as an effective method in PA classification. However, there are also efforts in developing clustering methods in PA recognition. The MCODE method by Lu et al. (2017) is an example. MCODE is first applied to protein networks, and it is an efficient clustering algorithm based on density. It takes input as an undirected graph constructed by the euclidean distance of the features, and outputs the clusters after a vertex weighting process. Lu et al. (2017) reached an accuracy of 74% in classifying physical activities using the MCODE unsupervised learning. Kuppevelt et al. (2018) applied a hidden Semi-Markov Model (HSMM) to identify 10 states of behaviours, and achieved similar results as the ones by the cut-points approach.

2.4.3 Deep learning

Deep learning is also applied by many studies in PA detection. In comparison with machine learning, deep learning skips the feature engineering process of machine learning that describes the characteristic of the training data. Deep learning uses a multi-layer perceptron structure that originates from neural network to learn the inherent law and representation level of the training data. In simple terms, deep learning learns good features by itself compared to machine learning. In this vein, deep learning classifiers present more model parameters and have higher level of difficulties in the model training process.

Ordóñez et al. (2016) applied a deep architecture based on the combination of convolutional and LSTM (Long-short-term memory recurrent) recurrent layers in activity recognition datasets from wearable sensors for the first time. The intention to incorporate the recurrent layers was to deal with the limitation of convolutional kernels that were only able to learn temporal dynamics within the duration of the kernel. This structure was proved to provide a very good trade-off between performance and training/recognition time in comparison to a standard CNN. And this model was able to learn from multiple signals and fusing them without any specific preprocessing compared to machine learning models. The result showed that the gyroscope data improved a 15% performance while both gyroscope and magnetometers improved the performance by 20% compared to accelerometer data alone. This approach was however only tested on small-size datasets, how will it perform on large-scale data and how to tune the model parameters will still need to be investigated.

Jiang et al. (2015) assembled accelerometer and gyroscope signal sequences into a novel activity image, and designed a Deep Convolutional Neural Network (DCNN) to automatically learn features, rather than defining the features manually. The architecture contained two convolutional layers (5 kernels of 5*5 size in the first layer and 10 kernels of 5*5 size of the second layer) and resulted in a 120-dimensional feature vector. In addition, this paper also designed a DCNN+ architecture which added a binary-class SVM classifier to classify uncertain classes after the DCNN classifier if the activity classes distribution of resulted DCNN was not sharp enough. From the result, the two methods achieved superior performance in terms of both recognition accuracy and computational cost compared to SVM classifier, and a

feature selection method with random forest. It is noteworthy that the SVM method provided second highest accuracy but relatively much longer computational time compared to the other three (an average of 10 milliseconds of three datasets compared to an average of fewer than 2 milliseconds for the other three). Therefore, it requires further investigation of a better method for large datasets with regard to the balance of accuracy and computational time.

2.4.4 Dimension Reduction

Sukor et al. (2018) compared human activity recognition performance with and without the PCA dimension reduction method. In their study, time and frequency domain features were also compared using different classifiers (DT, SVM, MLP-NN). The result showed that total average accuracy of the three tested classifiers was increased by 4.21% when dimensionality reduction using PCA was applied to the original features. And frequency-domain features showed higher total average accuracy compared to time-domain features, with a difference of 8.90%. This work proved the efficiency of the PCA method. On the basis of this, more dimension reduction methods can be explored and compared. In addition, this thesis only used the simplest validation method - train/test split with 70% of the data as training dataset and 30% as testing dataset. In this sense, more validation methods can be tested for the comparison. Moreover, the study also gave the performance result of different metrics, including the F-score, recall, precision, recall. A more comprehensive comparison of how and why the dimension reduction method function differently influence the different metrics can be analysed.

Chandrashekar et al. (2014) implemented two filter methods (Correlation, MI) and two wrapper methods (SFFS algorithm, CHCGA algorithm) with the performance with the classifiers (Support Vector Machine (SVM), Radial Basis Function Network (RBF)) as the objective functions. The comparison was done on seven different datasets. From the result, feature selection in general helped reducing the complexity of information and increases accuracy. However, it is noteworthy that the paper concluded that different feature selection algorithms perform differently on different data. The author suggested selecting a final feature selection algorithm by predefined baseline classification performance values among different algorithms. The paper used the number of reduced features and classification accuracy to compare the feature selection techniques. However, there are other selection considerations that can be taken accounts, such as simplicity, stability, computational requirements, and storage.

Dehzangi et al. (2018) evaluated dimension reduction algorithms (both feature selection and transformation) combined with different classifiers with the purpose to improve robustness without decreasing the prediction accuracy. The test dataset contained six physical activities from 30 participants from 19 to 48 with an IMU strapped on their waist. The results showed that Ensemble bagged classifier provided the highest accuracy among five classifiers (Decision tree, KNN, SVM, Neural network). Also, within dimension reduction algorithms, Neighborhood component analysis algorithms (96.3% accuracy with 9 features) with Ensemble classifier, and Random forest with Ensemble classifier yield (96.9% accuracy with 15 features) the

most effective performances compared to the state-of-art accuracy of 97% of the same dataset achieved on a 561-D feature space.

Peterek et al. (2014) compared two feature extraction methods (the Correlation Feature Selection Method and the Principal Component Analysis) with the combination of three supervised classification algorithms: the Linear Discriminant Analysis, the Random Forest, and the K-Nearest Neighbours. Besides what is mentioned above in Chapter 2.2 from the perspective of supervised classifiers, the study also gave insights into feature selection methods. From their results, the correlation feature selection method (CFS) at large outperformed the PCA method. Reduction of the dataset by the CFS increased the precision of walking and walking upstairs but decreased the precision of the rest states. PCA method was claimed to be completely failed. However, in their study, the parameters for the two feature selection methods were set as fixed numbers. Therefore, it is worth investigating the hyperparameter setting for the PCA method as well as trying other different classifier combinations for the PCA method.

Baldominos et al. (2017) explored and compared different feature selection techniques using genetic algorithms (GA) to improve the accuracy and reduce the number of sensor dimensions. In the study, there were 40 dimensions (x, y, z dimensions of each sensor) of the features, obtained by 3 IMUs, each with an 3D accelerometer, gyroscope, and magnetometer. Four feature selection alternatives were proposed and tested, namely, attribute selection (represents a total set of 280 features); dimension selection after feature sensibility (some-or none) (select sensor dimensions first, then select part of features contained in the sensors), dimension selection after feature sensibility (take-it-all or leave-it) (select sensor dimensions first, then take all features contained in the sensors), dimension selection without feature sensibility (select features without considering the sensor dimensions). Afterwards, a classifier was trained using extremely randomized trees. By a LOSO evaluation approach, very high accuracy of 97.45% was reached. This level of accuracy was achieved by the first attribute selection alternative, representing applying GA on the whole data set of 280 features. Nevertheless, other methods also reached high accuracy above 96.63%. This study gave insights in dimension reduction in different sensors. For example, the dimension reduction could be applied after the selection sensor dimensions.

On the basis of the described works, Table 2.2 shows the most accurate classifiers and the combined feature selection methods in these studies.

Table 2.2 Most Accurate Classifiers and Feature Selection Methods

Literature	Most accurate classifier	Classifier types	Accuracy	Number of features	Feature selection method
Zhang & Sawchuk (2011)	SVM	Supervised learning	93.1%	50	SFS
Zheng et al. (2015)	LS - SVM	Supervised learning	95.6%	6	PCA
Ordóñez et al. (2016)	DeepConvLSTM	Deep learning	95.8%	-	-
Jiang et al. (2015)	DCNN + SVM	Deep learning	98.45% (average of three datasets)	120	-
Dehzangi et al.	Ensemble bagging	Fusion method	96.9%	15	Neighborhoo

(2018)	(SVM, KNN)				d component analysis
Sukor et al. (2018)	MLP-NN	Deep learning	97.54%	3	PCA
Peterek et al. (2014)	LDA	Supervised learning	above 90%	211	CFS

2.5 Model Validation and Comparison

2.5.1 Validation methods

Proper validation techniques are helpful to understand the models and estimate unbiased generalization performances. The basis of all validation methods is splitting the data when training the models. On the basis of that, the most basic validation method is the holdout method. In this method, data is simply split into a subset for training and a subset for testing. The common split is 70% for training and 30% for testing, and only evolves on single run of classifier. A variant of this method is to introduce an additional holdout set (often 10% of the data) besides test and training splits (Grootendorst, 2019). The benefit of this method is that one can see directly how the model reacts to previously unseen data. However, this approach is very likely to suffer from sampling bias. Sampling bias represents a systematic error due to non-random sample of a population, causing some members of the population to be less likely to be included than others, resulting in a biased sample (Sampling Bias, 2021).

Cross-validation can minimize sampling bias and help to generalize the model to independent data. Two types of cross-validation categories can be distinguished, namely exhaustive and non-exhaustive validation methods. Exhaustive cross-validation methods are cross-validation methods that learn and test on all possible ways to divide the sample, while non-exhaustive validation methods do not try all possibilities to divide the original sample into a training and a validation set.

K-fold validation is one of the most common non-exhaustive validation methods applied in literature to validate the classifiers. It splits the data into k folds, then trains the data on k-1 folds and tests on the one fold that was left out. It does this for all combinations and averages the result on each instance. 10-fold cross-validation is commonly used, as it finds a nice balance between computational complexity and validation accuracy, but in general k remains an unfixed parameter (McLachlan, 2004).

The other common variants of k-fold validations are stratified k-fold cross-validation, and repeated cross-validation. In stratified k-fold cross-validation, the partitions are selected so that the mean response value is approximately equal in all the partitions. In other words, in the case of N-classes classification, each partition contains roughly the same proportions of the N types of class labels. In repeated cross-validation, the data is randomly split into k partitions several times. The performance of the model can thereby be averaged over several runs. For example, given the number of repeats as p, the total number of runs for k-fold partitions is p*k times. However, this is rarely desirable in practice (Vanwinckelen, 2019).

In terms of exhaustive cross-validation methods, a basic one is the Leave-p-out cross-validation (LpO CV). LpO CV involves using p observations as the validation set and

the remaining observations as the training set. This is repeated on all possibilities to divide the original sample on a validation set of p observations and a training set. A special case for LpO CV is Leave-one-out cross-validation (LOOCV), when p equals 1. Also, for k -fold cross-validation, When $k = n$ (the number of observations), k -fold cross-validation is equivalent to leave-one-out cross-validation (Grootendorst, 2019).

Except for the above-mentioned common methods, there are other methods such as repeated random sub-sampling validation method, nested cross-validation methods, etc. Repeated random sub-sampling validation method creates multiple random splits of the dataset into training and validation data. Nested cross-validation is used simultaneously for the selection of the best set of hyperparameters and for error estimation. For example, Figure 2.2 shows a structure of nested cross-validation with the inner loop for hyperparameter tuning and the outer loop for estimating accuracy.

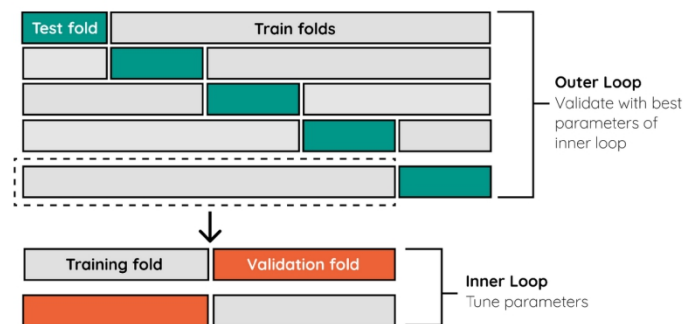


Figure 2.2 Nested cross Validation (Outer loop: 5-fold CV, Inner loop: 2-fold CV) (Grootendorst, 2019)

In PA studies, the k -fold (10-fold / 5-fold) cross-validation method is the most common implemented validation method. Table 2.3 shows the other types of validation method used in some studies.

Table 2.3 validation methods used in studies

Literature	Validation method
Peterek et al. (2014)	Hold-out validation (70%/30%)
Sukor et al. (2018)	Hold-out validation (70%/30%)
Zhang et al. (2011)	Leave-one-subject-out validation
Baldominos et al.(2015)	Leave-one-subject-out validation

2.5.2 Model evaluation metrics

The evaluation metrics applied in this analysis are as follow (Scikit learning, 2021):

Precision: Precision is defined as the ratio of True Positive (TP) to the sum of True Positive (TP) and False Positive (FP). It measures the model's accuracy.

$$\text{Precision} = TP / (TP + FP) \quad (1)$$

Recall: Recall is defined as the ratio of TP to the sum of TP and False Negative (FN). It measures the model's completeness.

$$\text{Recall} = TP/(TP+FN) \quad (2)$$

Accuracy: Accuracy is defined as the ratio of how correctly predict the observation to the total observation. Where TN refers to true negative.

$$\text{Accuracy} = (TP+TN)/(TP+FP+FN+TN) \quad (3)$$

F1 score: F1 score is defined as the average weight of Precision and Recall. F1 score gives best value when its value reaches 1 and at 0 value it gives the worst score.

$$F1 = 2 * (\text{Precision} * \text{recall})/(\text{Precision} + \text{recall}) \quad (4)$$

Area Under the Curve (AUC): AUC represents the area under the ROC-curve from prediction scores.

Cohen's kappa: Cohen's kappa is a score that expresses the level of agreement between two annotators on a classification problem. P_o is the empirical probability of agreement on the label assigned to any sample (the observed agreement ratio). P_e is the expected agreement when both annotators assign labels randomly.

$$k = (p_o + p_e)/(1 - p_e) \quad (5)$$

Matthews correlation coefficient (MCC): MCC takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC is in essence a correlation coefficient value (also referred to as phi coefficient) between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 an average random prediction and -1 an inverse prediction.

Besides the model scores tested by different metrics, there are also other important parameters to evaluate the models, such as the model costs and model complexity. In the process of developing a machine learning model, the number of features is one main factor that has a great impact on the model costs and complexity. The high number of features increases the building time of the recognition models. Also, when trained, they suffer from the curse of dimensionality and poor generalization and affect the accuracy in turn (Dehzangi et al. 2018).

2.6 Research gaps

The research gaps that will be explored and discussed in this thesis are as follows:

First, as presented above, most of the studies apply data that are collected in laboratory settings in the PA detection topic. This thesis intends to investigate PA

classification on a relatively large real-life data set characterised by older adults as participants.

Second, the data itself is featured by a lot of less intensive daily activities performed by older adults. This makes the data contain plenty of walking segments. At the same time, the detection of PA in studies is mostly focused on distinctive types of activities, (i.e. standing, sitting, walking, jogging, etc), which makes the detection of walking types a topic worth exploring.

Third, accelerometer is still the main sensor in PA detection studies. Though a few papers explored the inclusion of other sensors, there is still a lot effort can be spare in inspecting the usage of other sensors. In this case, the location features of GPS are still an unexplored point in walking types detection.

Fourth, though there exist some efforts in implementing feature selection methods in PA recognition. Given the characteristics of the real-life data, on which a comparison of the feature selection methods combined with different classifiers can still contribute to the understanding of the effectiveness of the methods that are compared in this topic.

Fifth, distinctive features for PA types detection are explored by some papers (see Chapter 4.3.1). However, the comparisons in those studies are mostly based on small number of feature sets. This thesis plans to select distinctive features more suitable for the older population based on a more comprehensive set of signal features with the help of different feature selection methods mentioned above.

Sixth, most studies only applied one validation method to test the model. This thesis intends to compare a pair of validation methods to see their differences.

3 Data

3.1 introduction

This thesis uses the dataset collected by the Mobility, Activity and Social Interaction Study (MOASIS) study. In this study, the data is collected by a custom-built mobile device UTrail. UTrail measures the participants' spatial activities by a GPS sensor, physical activity by an IMU sensor, and social interaction by a microphone sensor. For the purpose of this thesis, only the data collected by the GPS (1HZ) and IMU (50HZ) sensor highlighted with the blue frame is employed.

	Sensor	Variable	Sampling rate
Spatial mobility	GPS	timestamp, latitude, longitude altitude, speed	1/sec
Physical activity	IMU	timestamp, acceleration (x,y,z) magnetometer (x,y,z)	50/sec
Social interaction	EAR	timestamp, sound sample	1/12.5min




Figure 3.1 uTrail tracker sensors in MOASIS study

The IMU data in MOASIS study does not contain gyroscope data. It contains data from two sensors, namely accelerometer, and magnetometer. The magnetometer is used for the measurement of absolute direction (see Chapter 2.1 for more details). In this sense, it is not necessary to have this feature for the classification of physical activity types. More specifically, the recognition of the activity types of Descending, Ascending, Slow walk, Fast walk, Stationary, Jogging, and Biking in this thesis is not dependent on the directions compared to the recognition of activity types of Walking right and Walking left. Therefore, in this thesis, only accelerometer data is taken into account from the IMU data for analysis.

3.2 MOASIS

MOASIS study collects the real-life mobility data of a total of 164 participants aged between 65 and 80 from the German-speaking region of Switzerland for a period of two times two weeks in summer 2018. Participants are asked to carry the device laterally on the hips, preferably in a pocket or on the belt. As one of the aims of this thesis is to examine the classification results with and without the additional GPS sensor beside the accelerometer sensor, only the data with the coverage of both sensors are considered as valid data for the analysis.

In the MOASIS dataset, the level of completeness of GPS data is lower than IMU data due to the loss of signal for GPS sensors in some environments. Therefore, this thesis takes the GPS data length as the criterion to select participants to be analysed. Thus, participants with the longest valid GPS data are supposed to be selected from the MOASIS dataset. The distribution of the valid hours of the participants (excluding 17 participants who have no recorded valid hours) processed by preliminary research from Corti (2017, 32) can be seen in Figure 3.2. The red lines mark the data of the top 13 participants with the highest valid hours selected in the first step (only one period of two weeks is used considering the repetitiveness of participants' activity patterns in both periods). With an extra selected Participant 1, a selection of 224 days and a total of approximately 2202 valid hours is completed (Corti 2017, 32).

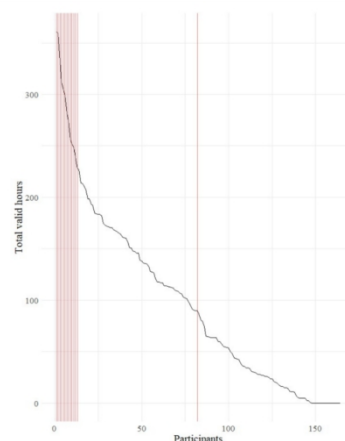


Figure 3.2 Valid hours distribution of participants in MOASIS study

From Corti's inspection, the quality of the data is not as ideal as expected. Due to the large scale of resolution loss, some parts of the recorded data frequency end up as around 0.2 to 2 HZ (Corti 2017, 33). By Corti's research in transportation mode detection, a total of 9,824 valid minutes (stationary status excluded) are screened out of the aforementioned 2020 hours based on the minimum requirement of 0.5 HZ for the analysis. In summary, taking the resolution into consideration, the valid time segment from the top 13 participants plus Participant 1 is 9,824 minutes.

Further, since the aim of this thesis is to study Physical Activity, a part of the data that records the passive-active status (with transportation) of the participants is not useful. Therefore, a second examination of the valid hours for physical activity detection among the selected 14 participants is conducted. For this purpose, a threshold is applied here to exclude participants who have long records of transportation but short records of physical activities (different levels of walking, and jogging). More specifically, a rank of participants is conducted by counting the GPS observations within a speed of 9m/s, which is a knowledge-based high-level walking and jogging speed for older people from literature. It is noteworthy here that biking is not taken into consideration, since its speed range overlaps with the speed ranges of transportations, and is thus not distinguishable by the threshold-based method. From the results shown in Figure 3.3, 10 participants marked by the red frame are finally selected for the further labelling procedure.

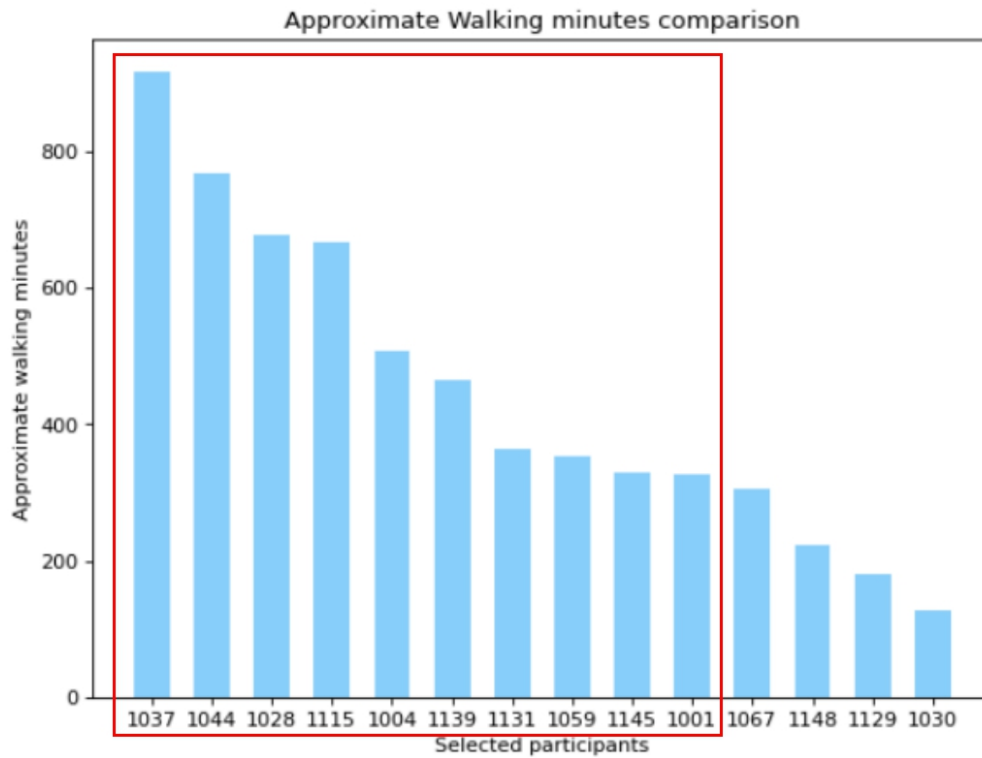


Figure 3.3 Valid approximate walking hours distribution of the 14 top participants

3.3 Labelling of MOASIS data

In the MOASIS project, participants are not required to record their physical activities. Therefore, the physical activity types are labelled manually through a labelling Application Shiny App. The interface of the App is shown below (Figure 3.4).

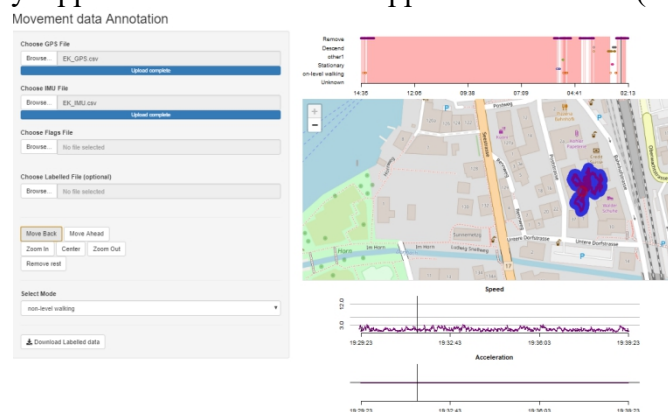


Figure 3.4 the Interface for the Labelling Application Shiny App

On the right side of the interface, the up part bar shows an overview of the whole trajectory of the participant in different colours representing different physical activity modes. In the middle, a map displays the participant's trajectory every 10 minutes. On the bottom, two axes display the speed from the GPS sensor, and total acceleration from the Accelerometer sensor according to the trajectory in the map. By the click of the signal wave on either of the axes, the respective signal recorded location is shown on the map as a red circle. At the same time, the App console gives information on the

altitude of the location. This function is a modification of the App for more convenient labelling of the ascending and descending segments. With the above information, trajectories are labelled using the options from the left side of the interface.

It is noteworthy that given one of the aims of the thesis is to exam real-life data, the selection of physical activity types to be recognised in this thesis is driven by the data itself. Also, the activity types are also constrained by the limitation of the manual labelling strategy. This results in less refined classes of activities. However, it also reveals the real-life situation of the physical activities of older adults.

Before the labelling process, a preprocessing is conducted to improve the data quality. As stated above, this thesis supposes to use data from the GPS sensor. Due to the consideration of the low quality of altitudes obtained by the GPS sensor, a mapping of the observations to the high-resolution DEM (digital elevation model) of Switzerland to obtain more accurate altitudes is conducted by QGIS. Figure 3.5 shows the Participant 1037's trajectories as blue dots distributes on the DEM raster of Switzerland.



Figure 3.5 Mapping participants locations to high resolution DEM

Since the DEM provides only the outdoor ground elevations than altitudes of the activities' observations in indoor buildings, the classification process should only include outdoor activities. Therefore, the first step in labelling is to exclude the indoor walking segments. Indoor walking segments are observable by this App shown as trajectories inside the buildings. One can see the indoor and outdoor walking segments of the same participant (Participant 1044) from Figure 3.6.

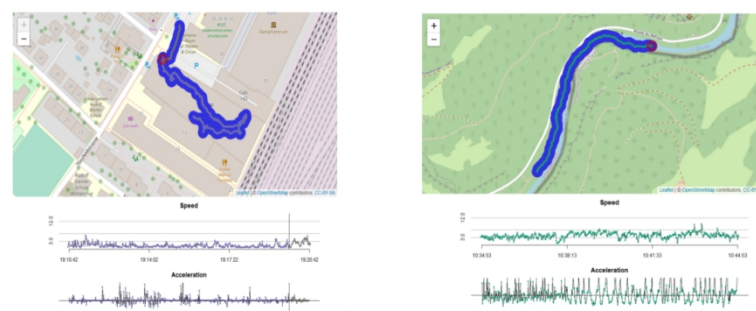


Figure 3.6 Comparison of indoor and outdoor walking

Second, among outdoor activities, different physical activities have different speed values. In the paper from (Park et al., 2011), for older adults, the non-level statistical walking speed is between 1m/s to 4.5m/s, the levelled walking speed is between 3.5m/s to 6m/s, the jogging speed is between 6m/s to 9m/s, and the cycling speed is between 12m/s to 25m/s. Therefore, the label of Slow walk, Fast walk, Jogging, and Biking can be conducted given the knowledge-based threshold.

Table 3.1 threshold based on GPS speed values for the 4 activities

Activities	GPS speeds
non-level walking speed (Slow walk)	1m/s - 4.5m/s
levelled walking speed (Fast walk)	3.5m/s - 6m/s
Jogging	6m/s - 9m/s
Biking	12m/s - 25m/s

In the labelling procedure in this thesis, considering the data itself, the threshold for Slow walk and Fast walk is set as 3m/s. And the Jogging activities are recognized as speeds higher than 6 m/s and below 10m/s. The biking segments are defined as most observations have speed values above 10m/s. Except for the speed from GPS sensor, the acceleration values also exert different characteristics for the manual differentiation of activities. The fluctuation amplitudes of total acceleration differ among activities. Jogging fluctuates most, followed by fast walk, slow walk, and then biking.

Third, as mentioned above, ascending and descending activities are mostly labelled by the altitudes of the participants' locations. Depending on the property of the data itself, the two types of activities are mostly taken from the hiking segments from participants. Figure 3.7 shows one ascending segment labelled from the hiking trip from one of the participants.

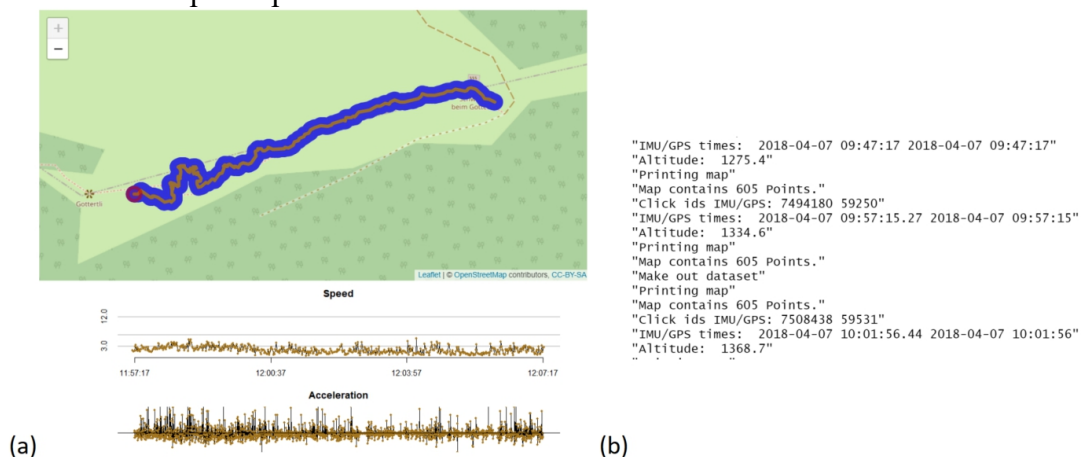


Figure 3.7 Ascending segment (a) and the increasing altitudes displayed in the console (b)

In addition, there exist short periods of stationary points among long walking trajectories. It makes more sense to leave them for classification as it represents a

real-life circumstance. However, long stationary periods are removed from the valid data for PA recognition. Besides, there are other types of activities that are found out during the labelling processing, such as possible golf-playing, rowing activities, etc. However, these activities have a relatively short time frame compared to others, and have uncertainty determined by the environment in visual inspection. Therefore, these activities are categorized as the others activities, which are not included in the classification process. Figure 3.8 shows the removed activity segments in the river that reckoned as a possible rowing segment.

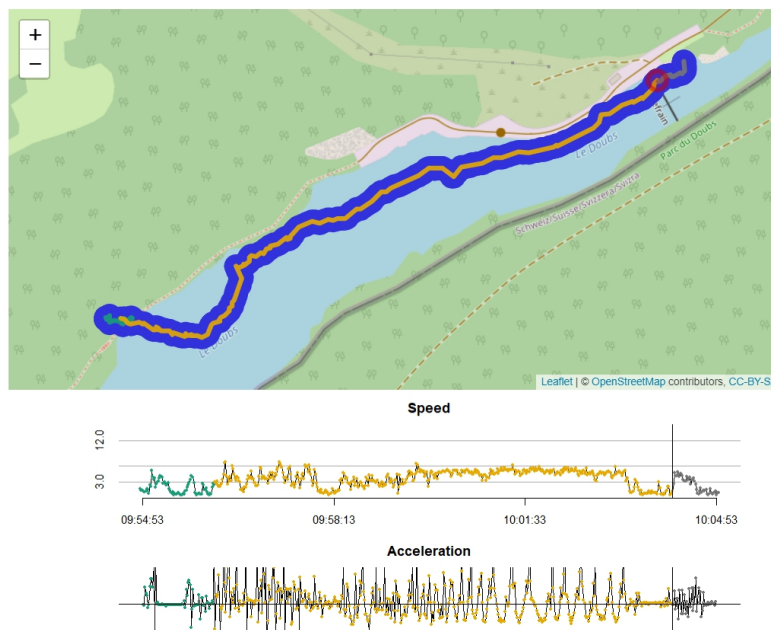


Figure 3.8 Other activities

In the end, there are in total 7 most common physical activity types found out from participants in this dataset, namely Slow walk (outdoor), Fast Walk (outdoor), Descending, Ascending, Jogging, Biking, and Short Stationary (later referred as Stationary). Figure 3.9 shows different activities labelled in different colours in the window size of 10 minutes. Given the window length for trajectories in each display as 10 minutes and the participants' activity pattern, the minimum segment resolutions are around 1.5 to 3 minutes.

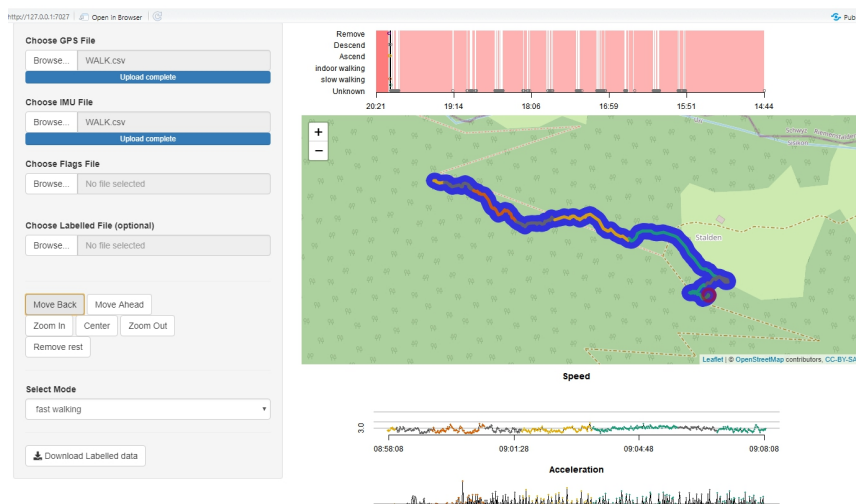


Figure 3.9 Different activities labelled in different colours

Figure 3.10 shows the amount and distribution of the 7 types of activities per minute used from the MOASIS dataset used in this study. The Slow walk has the highest number of records (more than 2500 minutes), followed by Fast walk and Ascending. The other modes of physical activities have lower fewer observations, especially stationary has very little data (around 300 minutes). Overall speaking, the time length distribution of the activity types are not balanced. This implies the importance to select proper cross-validation methods to split the data and train the models in the next steps.

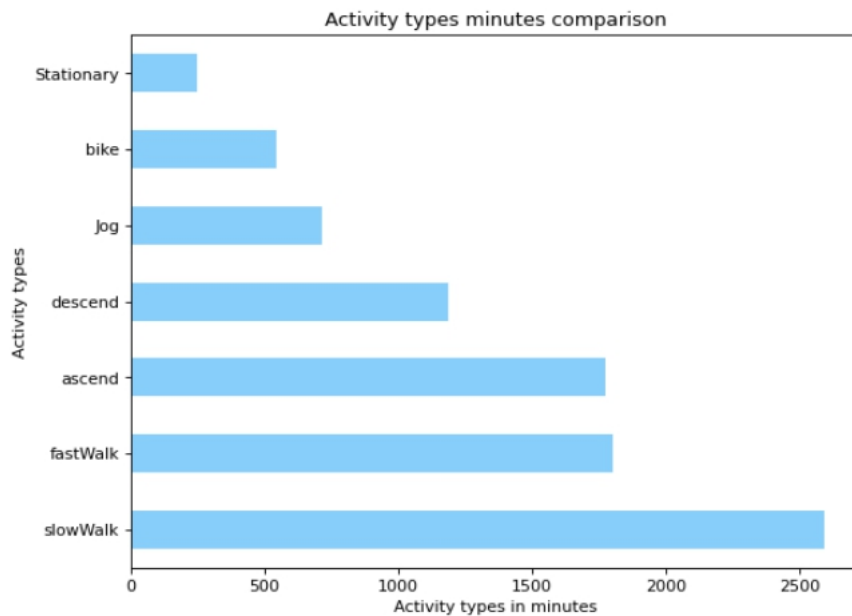


Figure 3.10 the amount of the 7 types of physical activities

4 Methods

This thesis aims to classify the manually labelled real-life physical activity data of older adults to find out the best classifiers for older adults' PA recognition. The whole PA recognition process follows the normal procedure for PA recognition described by (Bulling et al. 2014). The procedure consists of four steps, namely pre-processing, segmentation, feature extraction, classification. Figure 4.1 illustrates the procedure in an intuitive way. This study uses python version 3.7.4 to perform the whole classification procedure. An overview of the scripts for analysis and the key functions can be seen in Table (Appendix 8.2).

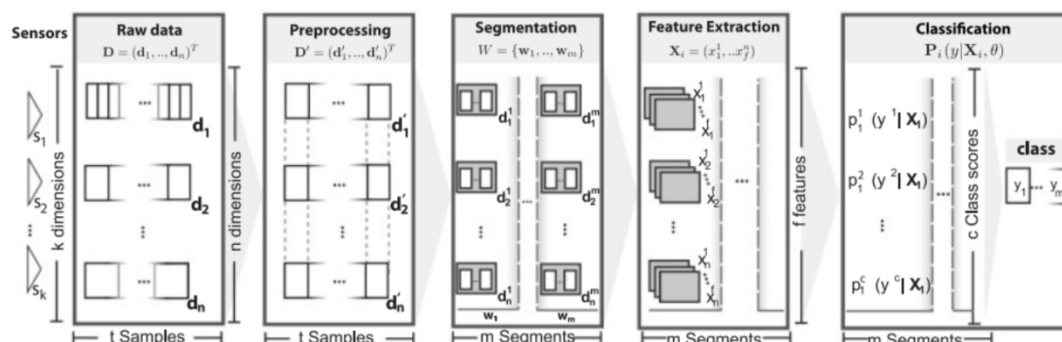


Figure 4.1 PA classification steps

4.1 Pre-processing

In the manual labelling process (Chapter 3.3), an initial pre-processing to remove the long stationary periods and transportation periods, as well as the invalid and incomplete sensor coverage is already realized by visual inspection of sensor signals. After the data cleaning, a further step to process the GPS and Accelerometer data is performed.

First, the acceleration signal is highly fluctuating and is composed of 2 components - the fast-varying component due to physical activities (AC) and the slow-varying one due to the gravitational force acting on the body (DC) (Dehzangi et al., 2018). A low pass filter can separate the slow-varying body component of acceleration. A high pass filter can separate the fast-varying body component of acceleration. Figure 4.2 shows an example of the low pass filter and high pass filter (Bayat et al., 2014) In this thesis, a digital low pass filter from the paper by Bayat et al. (2014) is applied to the signal and leave the fast-varying component to be further analysed. From the paper, the low pass filter is calculated as follows: $A_{DC}[n] = a_1A[n] + b_1A_{DC}[n-1]$, where A_{DC} is the filtered output data and A is the raw input data. The filter coefficients a_1 and b_1 are constants that are computed using sampling rate and cut-off frequency. It is reported that optimal cut-off frequency in order to exclude the gravity component alone would

range from 0.1 to 0.5 HZ (Fujiki et al., 2010). This thesis takes the cut-off frequency as 0.25HZ, and the sampling rate is determined by the IMU sampling rate as 50 HZ. The scripts for the low pass filter can be found in Table 8.5 (Appendix 8.2).

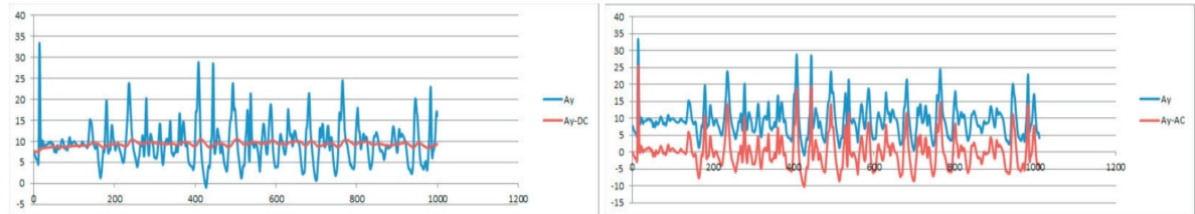


Figure 4.2 low pass filter (Left) result in gravitational acceleration, high pass filter (Right) result in body acceleration (Bayat et al., 2014)

Second, for the GPS data, each entry's speed value is recalculated by averaging the values of the adjacent entries to reduce the abnormal values. The code for the GPS speed recalculation can be found in Table 8.5 (Appendix 8.2) as well.

4.2 Segmentation

Compared to the accelerometer sensor sampling rate (50Hz in this thesis), human activities are performed over a longer time. Thus, a single sample at a specific time instant does not provide sufficient information to define an activity (Dehzangi et al., 2018). To classify the activities, signal segmentation needs to be performed to extract quantitative measures to compare the signals' characteristics. Signal segmentation can be in general performed by two strategies, namely energy-based strategies and sliding window strategies. Energy-based segmentation is based on the fact that different activities are performed with different intensities which are represented as different energy levels. By defining the threshold E (energy of a signal), data segments identified are likely to belong to the same activity (Bulling et al., 2014).

Sliding window strategies divide the continuous signal streams into a certain length of windows. It is assumed that all information of each activity class can be extracted from each single window by choosing a proper window length. A sliding-window strategy can be determined by two factors, the window size and the step size. The window size represents the time length of each segmentation, and the step size is the adjacent windows' distance to each other. When the step size equals to the window size, the segments to be classified has no overlap. Otherwise, the segments extracted are overlapped. In sliding window segmentation, the window size has an direct impact on the delay of the recognition system. The larger the window size, the longer it takes for the next segments to be available for processing. Also, the window size can influence the recognition performance. Therefore, the step size is subject to a trade-off between segmentation precision and computational load. The larger the step size, the less the computational load, but also the less accurately the segmentation borders can be defined (Bulling et al. 2014).

A proper window size depends on the classes themselves to be recognized. For instance, the time to perform different physical activities is in general relatively much shorter than the time for transportation modes. Therefore, the proper window lengths are different for physical activities and transportation modes though the recognition of them usually employs the same types of data (IMU, GPS) and features. The proper window length to classify physical activities has been studied in many papers. The following table 4.1 lists the data quality and window of length applied in the relevant studies. It can be seen that the most common window lengths are in a high resolution of several seconds. However, in this study, rather longer window lengths of 180s, 90s, 60s are tested to investigate the most suitable classification window lengths for the data in this thesis. The selection of the window lengths takes two aspects into account. First, the minimum manual segmentation resolution for different activities is around 90 seconds. Therefore, too short segmentation lengths for classification are not very meaningful. Second, as mentioned in the last section, due to the loss of resolution in the labelling process, the labelled data for classification only has a frequency of 1 record per second. Compared to the general 50 - 100 HZ settings with 2s - 5s window lengths, the number of records for classification in the 1HZ setting should be similar as around 100 - 500 records, which end up with the window length selection of 60 - 180 seconds. This thesis also intends to compare if the overlap in window segmentation would have an influence in the results. To this purpose, windows with and without the overlaps (50%) are also applied and compared in the classification. After the segmentation, the information is then transformed into a feature vector by computing a variety of features within each window.

Table 4.1 Sampling frequencies and window types in literature

Literature	Sensor	Frequency for Accelerometers	Window length	Overlap/Step size
Zhang et al. (2011)	a 3-axis accelerometer, a 3-axis gyroscope	100HZ	2s	50%
Baldominos et al. (2015)	a 3-axis accelerometer, a heart rate monitor	100HZ, 9HZ	5.12s	-
Saez et al. (2015)	2 accelerometers, a gyroscope, a magnetometer	100HZ	5.12s	-
Ordonez et al. (2016)	5 accelerometers, 5 gyroscopes, 5 magnetometers	30HZ	5s	50%
Peterek et al. (2014)	a 3-axis accelerometer, 3 axial angular velocity sensor	50HZ	2.56s	50%

4.3 Feature calculation

4.3.1 Accelerometer features

4.3.1.1 Time and frequency features

After pre-processing, acceleration signal features as the time and frequency representations are derived from the selected participants' motion signals. These features are then used for activity recognition. This is based on the knowledge that various types of activities exert different characteristics in the time and frequency domain. For example, it can be seen from Figure 4.3 that the central tendency from accelerometer/gyroscope signal amplitudes can be used as a distinctive feature to distinguish laying from rest (Dehzangi et al., 2018) irrespective of the time domain position. However, from the figure, among walking activities, the total acceleration and gyroscope values are not enough in distinguishing the activity types. Therefore, a close examination of different features is the key to perform the most efficient and accurate classification.

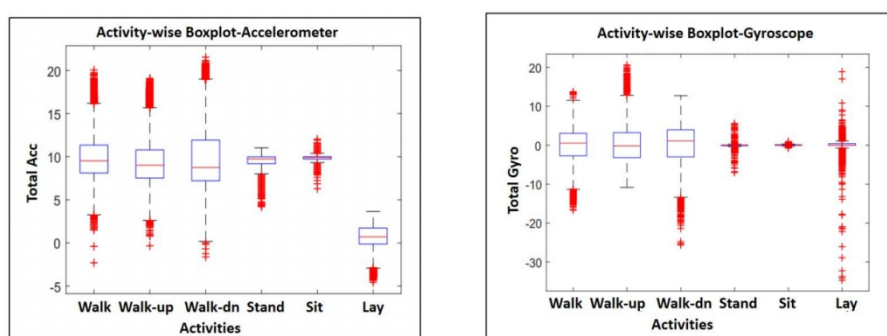


Figure 4.3 Activity-wise Boxplot for accelerometer and gyroscope signals (Dehzangi et al., 2018)

For the accelerometer sensor in PA detection, there are a considerable number of the time and frequency features that can be extracted for the quantitative representation of signals in literature. Table 4.2 summarizes the most common features in the time and frequency domains. The definition of these features can be found in Table 4.3.

Table 4.2 Most common features in the time and frequency domains for accelerometer sensor

Time domain features	mean, median, range, mean, max, average, variance, standard deviation, interquartile range of three axes and total acceleration; kurtosis, skewness of three axes; number of observations falling within each of 10 bins of the three axes; number of peaks (step counting), peak amplitude, peak time interval, peak-to-peak distance of three axes; absolute deviation, zero crossings; lag-one autocorrelation, autocorrelation sequence; (Allahbakhshi et al., 2019, Allahbakhshi et al., 2020)
Frequency domain features	power spectral density, energy of the signal; Mean, amplitude, power of the top (three) dominant frequencies of three axes and total acceleration; spectral entropy, cross-spectral densities, power of dominant frequency; (Allahbakhshi et al., 2019, Allahbakhshi et al., 2020,)

4.3.1.2 Statistical and physical features

Besides the categorization by time and frequency representations, Zhang et al. (2011) categorized features as physical representations and statistical representations. From their definition, statistical features (see Table 4.3) are features computed from each axis of the accelerometer or gyroscope sensors individually. While physical features are usually extracted from multiple sensor channels. During the process, sensor fusion is performed at the feature level. This is based on the definition of physical features as our physical interpretations of human motions. Table 4.4 listed the physical features designed by Zhang et al. (2011) that only need one non-directional accelerometer, and can be applied in this study.

Table 4.3 Statistical features with brief descriptions (Zhang et al. 2011)

Statistical feature	Description
Mean	The average value (DC component) of the signal over the window
Median	The median signal value over the window
Standard Deviation	Measure of the spreadness of the signal over the window
Variance	The square of standard deviation
Root Mean Square	The quadratic mean value of the signal over the window
Averaged derivatives	The mean value of the first order derivatives of the signal over the window
Skewness	The degree of asymmetry of the sensor signal distribution
Kurtosis	The degree of peakedness of the sensor signal distribution
Interquartile Range	Measure of the statistical dispersion, being equal to the difference between the 75th and the 25th percentiles of the signal over the window
Zero Crossing Rate	The total number of times the signal changes from positive to negative or back or vice versa normalized by the window length
Mean Crossing Rate	The total number of times the signal changes from below average to above average or vice versa normalized by the window length
Pairwise Correlation	Correlation between two axes (channels) of each sensor and different sensors
Spectral Entropy	Measure of the uniformity (irregularity) distribution of signal frequency components
Power spectral density	Measure of power of the signal as a function of per unit frequency

Table 4.4 Physical features with brief descriptions (Zhang et al. 2011)

Physical feature	Description
Movement Intensity (MI)	<p>The Euclidean norm of the total acceleration vector after removing the static gravitational acceleration.</p> $MI(t) = \sqrt{ax(t)^2 + ay(t)^2 + az(t)^2}$ <p>This feature is independent of the orientation of the sensing device, and measures the instantaneous intensity of human movements at index t. MI is not used directly. Instead, the mean (AI) and variance (VI) of MI over the window are computed.</p>

Normalized Signal Magnitude Area (SMA)	The acceleration magnitude summed over three axes within each window normalized by the window length. This feature has been used in previous studies and is regarded as an indirect estimation of energy expenditure.
Eigenvalues of Dominant Directions (EDD)	The top two eigenvalues of the covariance matrix of the acceleration data along x, y, and z axis in each window. This feature measures the corresponding relative motion magnitude along the directions. The first two eigenvalues correspond to the relative motion magnitude along the vertical direction and the heading direction respectively.
Dominant Frequency (DF)	The frequency corresponding to the maximum of the squared discrete FFT component magnitudes of the signal from each sensor axis.
Energy (ENERGY)	The sum of the squared discrete FFT component magnitudes of the signal from each sensor axis divided by the window length.
Averaged Acceleration Energy (AAE)	The mean value of the energy over three acceleration axes.
Correlation between Acceleration along Gravity and Heading Directions (CAGH)	The correlation coefficient between the acceleration in gravity direction and the derived acceleration along heading direction.
Averaged Velocity along Heading Direction (AVH)	The Euclidean norm of the averaged velocities along the heading axes over the window.
Averaged Velocity along Gravity Direction (AVG)	The averaged the instantaneous velocity along the gravity direction at each time t over the window.

4.3.1.3 Distinctive features

With the introduction of a wide range of features, it can be seen that a large number of features are calculated to train the classifiers in studies. Table 4.5 lists the number of features calculated at the beginning and selected in the end to train the models (if there is a feature selection process) in literature.

Table 4.5 The number of features calculated and used in the end for classification

Literature	Number of features calculated	Number of features used to train the classifier	Feature selection methods
Zhang et al. (2011)	110	50	SFS
Allahbakhshi et al. (2020)	85(1 sensor)/425 (5sensors)	85/425	-
Dehzangi et al. (2018)	561	9	Neighborhood Component Analysis
Peterek et al. (2014)	561	211	CFS

From the table, most papers used more than 100 features to train the classification models. However, a large set of features would cause redundancy, and result in poor generalization of the models. Therefore, it is important to select distinctive features

among a large number of features. In Table 4.6, distinctive features that can recognize different types of activities are summarized from several studies.

Table 4.6 Distinctive features to separate different types of physical activities (Bao et al., 2004; Zhang et al., 2011; Zheng et al., 2015)

Types of activities	Distinctive Features
Stationary vs. Moving	Movement Intensity (MI)
Standing vs. Sitting	Averaged Acceleration Energy (AAE), Total Acceleration Entropy, Total Acceleration Mean
Running & Jumping vs. Walking	Variance, Eigenvalues of Dominant Directions (EVA), Averaged Acceleration Energy (AAE)
Running vs. Jumping	Eigenvalues of Dominant Directions (EDD), Correlation between Acceleration along Gravity and Heading Directions (CAGH)
Walking left & walking right vs. Walking forward vs. Walking stairs	Averaged Velocity along Gravity Direction (AVG), Averaged Rotation Angles related to Gravity Direction (ARATG), Correlation between Acceleration along Gravity and Heading Directions (CAGH), Total Acceleration Entropy (TAE), Total Acceleration Mean
Walking left vs. Walking right	Correlation between Acceleration along Gravity and Heading Directions (CAGH)
Walking upstairs vs. Walking downstairs vs. Non-level walking	Correlation between Acceleration along Gravity and Heading Directions (CAGH), Averaged Velocity along Gravity Direction (AVG); Eigenvalues of Dominant Directions (EDD), Auto-correlation of total acceleration, Power Spectral Density, Total Acceleration Angle Entropy, Total Acceleration Mean

Based on the above review of accelerometer features for PA classification, this thesis selects two parts of the features (see Table 4.7) from the accelerometer sensor, namely the distinctive features as the first part of features, and other non-distinctive features (other common time and frequency domain features) as the second part of features. The scripts for feature calculation can be found in Appendix 8.2. The selected features in this step are further feed into the next feature selection process. Later, features will be tested on the classifiers to find out their importance in the performance of the classifiers. The top important features ranked by the classifiers will be analysed to see if there are other important features that do belong to the predefined distinctive features reviewed from literature. This step intends to find out other distinctive features for the activity category in this thesis beside what summarized in Table 4.6 above.

Table 4.7 Accelerometer features calculated for analysis

Feature category	Features	Number of features
Distinctive features	Total acceleration entropy (TAE) Eigenvalues of dominant directions (EDD) Movement Intensity (AI, VI) Averaged acceleration energy (AAE) Total acceleration mean Variance of three axes and total acceleration Auto-correlation of total acceleration Power spectral density	16
Non-distinctive features	mean, median, range, minimum, maximum, average, standard deviation, interquartile range of three axes and total acceleration kurtosis, skewness of three axes number of peaks (step counting), peak amplitude (total and mean)	95

	amplitudes in each window), peak time interval, peak width, peak-to-peak distance of three axes zero crossings, average crossings Autocorrelation of signals in three axes Energy of the signals in three axes Mean, amplitude, power of the top (three) dominant frequencies of three axes and total acceleration	
--	--	--

4.3.2 GPS features

For the GPS sensor, its usage in PA recognition is rare. In Allahbakhshi et al.’s (2019) study, features extracted from the GPS sensor are two statistical representations (mean, variance) of the speed. In Wu et al.’s (2011) research in time-activity pattern recognition, only GPS data was implemented to classify the activity patterns by a rule-based model and random forest model. In this study, the features derived were acceleration rate, speed, distance difference, and distance ratio. Acceleration rate was defined as the change in speed between a given point and the previous sequential point. The distance difference was calculated for any three sequential points. The distance ratio was calculated for a series of sequential points as the ratio between (a) distance between the first sequential point and the last sequential point in the series and (b) the sum of the distance for all line segments formed by sequential points in the series. A series was defined as various averaging time intervals ranging from 2 to 60 minutes and centered at each GPS point. The study successfully classifies the indoor, outdoor and in-vehicle status with 88% accuracy. However, the outdoor walking types and outdoor stationary detection by the GPS data in this study were not ideal.

Nevertheless, more GPS features can be inferred to inspect its contribution to the PA recognition topic. For example, the GPS sensor is an indispensable sensor in trip purpose inference (Gonzalez et al., 2008, Deng et al., 2010, Gong et al., 2014). The attributes of GPS data may vary depending on the types of GPS devices. They generally include: valid code marking, date, time, latitude, longitude, altitude, NSAT (the number of satellites that a GPS device used to calculate its position), HDOP (horizontal dilution of precision, measuring how the satellites are arranged in the sky at the time of the record), speed, and heading (Gong et al. 2014). In trip purpose identification, GPS information is generally combined with other types of information, such as GIS information (land use data) to infer the trip purpose of participants (Deng et al., 2010). The features can be extracted in this topic are the duration of the trips, the distance of the trips, heading direction, GPS signal quality, HDOP, etc (Gong et al. 2014). The GPS sensor is also widely used in the Indoor/outdoor detection theme. For example, Bui et al. (2020) used the number of satellites to detect indoor and outdoor environments and reached an overall accuracy of 97%. This thesis decides to calculate the GPS features shown in Table 4. for classification. Three distance features measure three types of (spatial, horizontal, and vertical) distances between the start- and end-points of segments. Three statistical features measure three aspects of the speed values of one activity segment. The other dimensions of GPS data such as HDOP, Number of satellites are not used in the analysis, since these features are only useful in detecting indoor activities from outdoor ones.

Table 4.8 GPS features applied

GPS features	Distance of locations between the start and end points of the segment (spatial)
--------------	---

	distance, horizontal distance, vertical distance); mean, variance, max speed
--	--

In summary, with the combination of 112 features from the Accelerometer sensor and 6 features from the GPS sensor, a total of 118 features are used for the feature selection process. The features from GPS sensor, accelerometer sensor, and both sensors will be tested to check the importance of sensor dimensions on the classification results.

4.4 Dimension reduction

In the past years, the number of features that can be used in machine learning applications has risen drastically from tens to hundreds for a more comprehensive description of the process to be recognized. However, too large domain of feature can result in irrelevant and redundant variables which would even confuse the classifier and decrease the prediction performance. “Curse of dimensionality” describes the phenomenon that the performance might degrade sharply when more features are added, while the training data is not enough to learn all the parameters of the activity models reliably (Zhang et al., 2011). There are two main techniques in reducing dimensionality, namely feature transformation (feature extraction) and feature selection (Masaeli et al., 2010). Feature transformation creates new features by transformations or combinations of the original feature set. Feature selection (variable selection) selects the most distinguishable subset from the original feature set. Feature selection is different from feature transformation in that no new features will be generated, but only a subset of original features is selected and the feature space is reduced (Liu et al., 1998). As to feature transformation, feature construction often expands the feature space, whereas feature extraction usually reduces the feature space (Liu et al., 1998). On machine learning, dimension reduction helps in reducing computation time and improving classification performance.

Table 4.9 Dimension reduction category

Dimension reduction	Description	Advantages and disadvantages
Feature transformation	Feature transformation creates new features by transformations or combinations of the original feature set. By this method, feature relevance is optimized and feature space are usually expanded.	Good generalization ability in new data but difficult to interpret
Feature selection	Feature selection selects the most distinguishable subset from the original feature set. By this method, feature space are usually reduced.	Might result in poor generalization ability in new data, but is helpful in understanding the data itself

4.4.1 Feature transformation

Feature transformation is a process that discovers missing information about the relationships between features and augments the space of features by inferring or

creating additional features (Liu et al., 1998). Transformation-based methods can be either linear or non-linear. Linear transformation-based methods, such as principal components analysis (PCA), linear discriminant analysis (LDA), find a transformation matrix to transform the original high-dimensional data into a lower-dimensional form. For non-linear methods (such as kernel PCA, kernel LDA), the goal is to find a non-linear mapping to a lower-dimensional space that optimizes some criterion.

4.4.1.1 Principal components analysis

PCA is one of the most popular feature transformation approaches that can reduce the dimensionality of data by transforming original features into new mutually uncorrelated features (Sukor et al. 2018). The new features are called principal components which are arranged according to their variances. And the other components that contribute to the lowest variances are usually omitted. The PCA process can be represented as steps below (Sukor et al. 2018) shown in Table 4.5.

Table 4.10 PCA process

Steps of PCA
1. Normalize the data
2. Calculate the covariance matrix
3. Calculate the eigenvectors and eigenvalues of the covariance matrix
4. Choose the components with the highest variances and form a feature vector
5. Derive a new dataset

In Ding's et al (2004) study, the authors explored the connection between the two widely used methods Kmeans and PCA. It is proved that principal components were actually the continuous solution of the cluster membership indicators in the K-means clustering method, i.e., the PCA dimension reduction automatically performed data clustering according to the Kmeans objective function (Ding et al. 2004). This provided an important justification for PCA-based data reduction for Kmeans classification. The authors applied Kmeans clustering on the PCA subspace from the original 1000 dimensions to 40, 20, 10, 6, 5 dimensions respectively. The result showed that as dimensions are reduced, the performance of the Kmeans classifier systematically and significantly improved. For example, for one of the test datasets, the cluster accuracy improved from 75% at 1000-dimensions to 91% at 5-dimensions.

PCA method is also used to maintain a good balance between the computational time criterion and performance accuracy criterion. El Moudden (2016) combined PCA with KNN classifier and at largely reduced the computational time of the high 561-dimensional data from 148080 seconds to 171 seconds with 26 dimensions with a minimized the accuracy score loss from 99.19% to 94.83%

4.4.2 Feature selection

In machine learning, Feature selection eliminates irrelevant variables that provide no extra information about the classes. To remove irrelevant features, a feature selection criterion that measures the relevance of each feature with the output class/labels is applied. From this point of view, compared to feature extraction, the generalization

ability of feature selection results might be lower. This is because that in feature extraction the good features selected can be independent of the rest of the data/ label itself, while feature selection might lead to poor generalization in a new dataset (Liu et al., 1998). However, feature selection can result in a better understanding of the data itself due to the process of feature transformation that projects high dimensional features to a low dimensional feature space lost the representational meanings of the features themselves.

Feature selection methods can be broadly categorized into 3 types: filter methods, wrapper methods, and embedded methods. A description with the advantages and disadvantages of the feature selection methods can be found in Table 4.6.

Table 4.11 Feature selection methods

Feature selection methods	Description	Advantages	Disadvantages
Filter methods	rank the features for a goodness measure and selects the best k features.	faster and simpler	lower results, poor generalization
Wrapper methods	comprise an induction algorithm and a Classifier which provides the fitness value for the induction algorithm	Better classification results	Higher computational costs, overfitting and poor generalization
Embedded methods	include the feature selection as part of the predictor without splitting the training and testing data	Reduced computational time, good generalization	

4.4.2.1 Filter method and ReliefF

Filter methods rank the features by scoring individual variables before applying them to the predictor. Filters are generally considered as most faster and simpler algorithms. Some filter measures applied in the literatures (see Table 4.7) are Pearson correlation coefficient (Chandrashekar et al., 2013), Mutual Information (Chandrashekar et al., 2013), R-square (Tulumet et al., 2013), false discovery rate (Tulumet et al., 2013), and etc. This thesis select ReliefF as the filter method for further analysis.

Table 4.12 Filter methods

Filter methods	Definition
Pearson correlation coefficient	Linear dependencies between variable and target
Mutual Information	Dependencies measured by uncertainty (information content) of one variable under the condition of the other
R-squared	Dependencies measured by the proportion of the variance for a variable in a regression model
false discovery rate	Dependencies measured by the rate of type I errors in null hypothesis testing

ReliefF algorithm is an extension of the Relief algorithm which is applicable to dual-class problems. The core idea behind the Relief algorithms is to estimate the quality of attributes based on how well their values distinguish between instances that are near to each other (Robnik-Šikonja et al. 2003). For that purpose, given a random instance R_i , Relief first searches its two nearest neighbors, one from the same class, called nearest hit H , the other from the different class, called nearest miss M . Second, the quality estimations $W[A]$ for attributes A is updated by the values for R , M , and H . Second, the quality estimation $W[A]$ increases when the attribute A values for R_i and M are different, which indicates that attribute A is desirable in separating the two instances R_i and M . On the contrary, the quality estimation $W[A]$ decreases when the attribute A values for R_i and M are different, which indicates that attribute A is not desirable in separating the two instances R_i and M . In the third step, the whole process is repeated for m times, where m is a user-defined tuning parameter.

ReliefF as the extension of Relief, is not limited to two-class problems. Similarly, ReliefF searches for k of one instance's nearest neighbors (hits H_j and miss M_j) instead of one from the same and the other classes in the first step. In the second step, the quality estimation $W[A]$ is updated by the average of the contribution of all the hits and all the misses. The contributions of each class are different, and will be made by weighting misses of each class with the prior probability of that class $P(C)$. The probability sum of classes of misses is set to 1. Due to the hits' class is missing, in the sum each probability weight is divided by factor $1 - P(class(R_i))$ (the sum of probabilities for the misses' classes). The process is repeated for m times. User-defined tuning parameter k controls the locality of the estimates. Most of the time, it can be safely set to 10 (Robnik-Šikonja et al. 2003).

```

Algorithm RReliefF
Input: for each training instance a vector of attribute values  $x$  and predicted
value  $\tau(x)$ 
Output: vector  $W$  of estimations of the qualities of attributes

1. set all  $N_{dC}$ ,  $N_{dA}[A]$ ,  $N_{dC\&dA}[A]$ ,  $W[A]$  to 0;
2. for  $i := 1$  to  $m$  do begin
3.   randomly select instance  $R_i$ ;
4.   select  $k$  instances  $I_j$  nearest to  $R_i$ ;
5.   for  $j := 1$  to  $k$  do begin
6.      $N_{dC} := N_{dC} + \text{diff}(\tau(\cdot), R_i, I_j) \cdot d(i, j)$ ;
7.     for  $A := 1$  to  $a$  do begin
8.        $N_{dA}[A] := N_{dA}[A] + \text{diff}(A, R_i, I_j) \cdot d(i, j)$ ;
9.        $N_{dC\&dA}[A] := N_{dC\&dA}[A] + \text{diff}(\tau(\cdot), R_i, I_j) \cdot$ 
10.         $\text{diff}(A, R_i, I_j) \cdot d(i, j)$ ;
11.     end;
12.   end;
13. end;
14. for  $A := 1$  to  $a$  do
15.    $W[A] := N_{dC\&dA}[A]/N_{dC} - (N_{dA}[A] - N_{dC\&dA}[A])/(m - N_{dC})$ ;

```

Figure 4.4 Pseudo code of ReliefF algorithm (Robnik-Šikonja et al. 2003)

Compared to Relief, ReliefF ensures greater robustness and can deal with incomplete and noisy data. To deal with incomplete data, missing values of attributes can be treated probabilistically. Nevertheless, the data used in this thesis has been removed records with missing values in the previous preprocessing step.

4.4.2.2 Wrapper methods and Genetic Algorithm

Wrapper methods select features based on the performance of the predictor by a number of search algorithms. Wrapper methods consist of an induction algorithm

such as genetic algorithm or a similar optimization method, and a classifier which provides the fitness value for the induction algorithm (Chandrashekar et al., 2013). Wrapper methods in general give better results but have higher computational costs compared to filter methods. Wrapper methods can be broadly classified into Sequential Selection Algorithms and Heuristic Search Algorithms (Chandrashekar et al. 2013).

Sequential Feature Selection (SFS) Algorithms starts with an empty set and adds one feature per step which can give the best value for the objective function. This process is repeated until the required number of features are added. In this process, each individual feature is permanently included if it is selected. As an extension of the SFS, Sequential Floating Forward Selection (SFFS) algorithm introduces an additional backtracking step and is more flexible. It adds another step which excludes one feature at a time from the subset obtained in the first step and evaluates the new subsets by the value of the objective function. If the value is decreased it will go back to the first step with the new reduced subset. This process is repeated until required performance is reached.

Genetic Algorithms (GA) are a type of common heuristic search algorithms. GAs are a family of computational models inspired by Darwinian Evolution (Cilla et al. 2009). These algorithms apply a simple chromosome-like data structure on a population of individuals, where each one represents a different solution to a problem, and uses selection and recombination operators to generate new sample point in a search space (Whitley 1994). In feature selection, chromosome bits represent if the feature is included or not, and the global maximum for objective function, which is the predictor performance can be found which gives the best suboptimal subset. This search method does not guarantee to find the optimal solution (Cilla et al. 2009). As one the wrapper methods, one of the drawbacks of the GA method is that the computational cost is high. Besides, using the classifier performance as the objective function tends to cause overfitting and result in poor generalization. The pseudocode of a simple genetic algorithm can be seen below Table 4.13 (Cilla et al. 2009).

Table 4.13 pseudocode for Genetic Algorithm

Algorithm2 Genetic Algorithm
<i>Population</i> ← <i>init</i> while stop condition is not satisfied do <i>Evaluate(Population)</i> <i>NextPopulation</i> ← <i>selection(Population)</i> <i>NextPopulation</i> ← <i>reproduction(Population)</i> <i>Population</i> ← <i>replacement(Population, NextPopulation)</i> end while <i>Solution</i> ← <i>best(Population)</i>

The main drawback of Wrapper methods is that the number of computations required to obtain feature sets. For each feature subset, the predictor creates a new model. If the number of samples is large, the algorithm spends a large amount of time in training the predictor. Another drawback of using the classifier performance as the objective function is that the classifiers are prone to overfitting. Using classification accuracy in subset selection can result in a bad feature subset with high accuracy but poor generalization power.

4.4.2.3 Embedded methods

Embedded methods include feature selection as part of the predictor without splitting the training and testing data. This means that the feature selection is incorporated as part of the training process, thus the computation time is reduced largely. Embedded methods differ from other feature selection methods in the way feature selection and learning interact. Filter methods do not incorporate learning in feature selection. Wrapper methods use a learning machine to measure the quality of subsets of features without incorporating knowledge about the specific structure of the classification or regression function (Lal et al., 2006). Therefore wrapper methods can be combined with any learning machine. In contrast to filter and wrapper approaches, embedded methods do not separate the learning from the feature selection part—the structure of the class of functions under consideration plays a key role (Lal et al., 2006). Therefore, unlike the wrapper methods, the combination of the feature selection method and the classifier is not always commutable.

SVM-RFE method is one of the most popular embedded feature selection methods in literature (Chapelle et al., 2008). SVM-RFE method applies an SVM classifier to perform Recursive Feature Elimination (RFE) that uses the weights of features as the ranking and the change of objective function as the search criteria. Recursive Feature Elimination (RFE) attempts to find the best subset of size k by a kind of greedy backward selection, given that one wishes to employ only $k < n$ input dimensions in the final decision rule (Guyon et al., 2002). It operates by trying to choose the k features which lead to the largest margin of class separation, using an SVM classifier (see Chapter 4.5.2). This is solved by removing the input dimension that decreases the margin in a greedy manner at each iteration of training the least until only k input dimensions remain. The pseudocode of a simple Recursive Feature Elimination (RFE) algorithm can be seen below Table 4.14. The algorithm can be accelerated by removing more than one feature in step 2.

Table 4.14 pseudocode for Recursive Feature Elimination

Algorithm3 Recursive Feature Elimination (RFE) in the linear case
1. Repeat
2. Find w and b (for SVM classifier) by training a linear SVM
3. Remove the feature with the smallest value $ w_i $.
4. Until k features remain

Similar to optimizing the SVM equation and assigning weight to features, the same can be done with Neural Networks. A feed-forward neural network, or multi-layer perceptron (MLP), is a computational model that processes information through a series of interconnected computational nodes. These computational nodes are grouped into layers and are associated with one another by weighted connections. The nodes of the layers are called units (or neurons) and transform the data by means of non-linear operations to create a decision boundary for the input by projecting it into space where it becomes linearly separable (Ordonez et al., 2016). In the multilayer perceptron networks, Network Pruning is commonly used to obtain the optimum network architecture. For example, a penalty can be applied for features with a small magnitude at the node and the nodes connecting to these features are excluded.

4.5 Classifiers

4.5.1 Kmeans

The K-means algorithm is a popular data-clustering algorithm. K-means cluster unsupervised learning method classifies the data by categorizing a dataset into a definite number k of clusters. The method uses k prototypes, the centroids of clusters m_k , to characterize the data. They are determined by minimizing the sum of squared errors of each node X_i 's distance to the centroid m_k of the cluster C_k (Pham et al., 2005). The K-means algorithm implementation in many software packages requires the number of clusters to be specified by the user. To find a satisfactory clustering result, usually, a number of iterations are needed where the user executes the algorithm with different values of K . When K-means clustering is used as the unsupervised classifier, the number of clusters k is equated to the number of classes in the data sets. The standard iterative solution to K-means suffers from a well-known problem: as iteration proceeds, the solutions are trapped in the local minima due to the greedy nature of the update algorithm. K-means Classifier is not considered as an effective method in PA type recognition, but it is sometimes applied in PA intensity detection. Zhao et al. (2018) detected four categories of human activities: light-intensity activity, moderate-intensity activity, vigorous-intensity activity, and fall by a user-adaptive algorithm based on K-Means clustering with inertial sensor signals.

4.5.2 SVM

Support Vector Machines (SVMs) are state-of-the-art large margin classifiers that have gained popularity in PA classification. Support Vector Machines were introduced by Cortes et al. (1995). The basic idea is to find a hyperplane that separates the D -dimensional data perfectly into its two classes. Since no prior knowledge about the data distribution is given, the optimal hyperplane $w \cdot x + b = 0$ is the one which maximizes the margin (see Figure 4.5). The optimal values for w and b can be found by solving a constrained minimization problem.

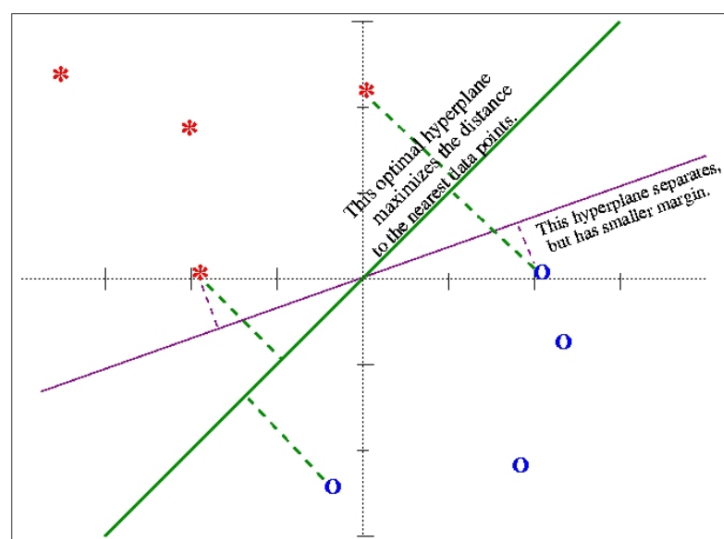


Figure 4.5 Hyperplane that maximizes the margin in SVM (Cortes et al., 1995)

For data that is often not linearly separable, the notion of a “kernel-induced feature space” is introduced which casts the data into a higher dimensional space where the data is separable (see Figure 4.6). Some useful kernels have been discovered and applied in literature such as polynomial kernel, Gaussian RBF Kernel, etc. For polynomial kernels, the functions to map the input data to a new feature space can be represented by the function:

$$K(x_a, x_b) = (x_a \cdot x_b + 1)^p \quad (1)$$

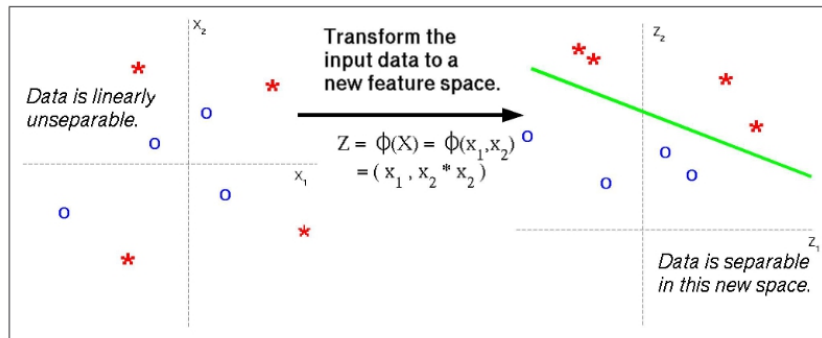


Figure 4.6 Hyperplane that maximizes the margin in Kernel SVM (Cortes et al., 1995)

where p is a tunable parameter, which in practice varies from 1 to 10 (Boswell et al., 2002). By using a larger value of p the dimension of the feature space is implicitly larger, where the data will likely be easier to separate. However, in a larger dimensional space, there might be more support vectors, which might lead to worse generalization (Boswell et al., 2002). For Gaussian RBF Kernel, the kernel result is a Radial Basis Function where σ is a tunable parameter with the support vectors as the centers. So here, an SVM is implicitly used to find the number (and location) of centers needed to form the RBF network with the highest expected generalization performance. The past work from Sunkad et al. (2016) has shown SVM with regularization parameter C as 100 and RBF Kernel can perform well in PA recognition.

4.5.3 Linear Discriminant Analysis

The Linear discriminant analysis (LDA) is a parametric classification technique. The main aim of this method is to find linear combinations of features, which provide the best separation between classes. These combinations are called discriminant functions. The basic principle of LDA is the measurement of metric or cosine distances between new instances and centroid of classes. The algorithm was developed by Fischer in 1931 but in his original form the algorithm was able to classify only into two classes. In 1988, the algorithm was improved for multi-class classification problems (Peterek et al. 2014). Figure 4.7 shows the feature transition before and after LDA in PA recognition (Khan et al., 2010). The LDA has a lot in common with the Principal Component Analysis (PCA) but with the difference that PCA is more used for feature separation and LDA does more feature classification (Peterek et al. 2014).

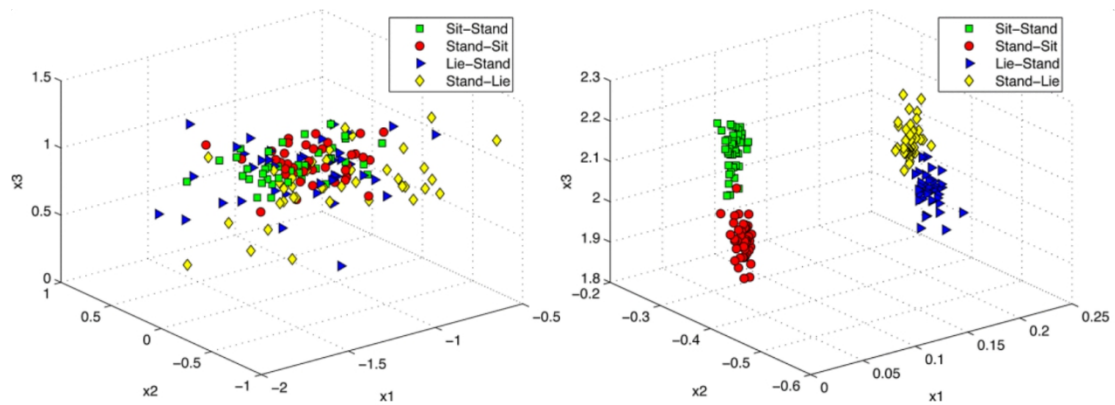


Figure 4.7 feature transition before and after LDA in PA recognition (Khan et al., 2010)

In the study from Peterek et al. (2014), the LDA classifier gave a good performance and was considered as a promising classifier in physical activity recognition. Moreover, Khan et al. (2010) concluded that this method was helpful in reducing high within-class variance, and allowed subjects to carry the sensor freely in any pocket without its firm attachment for PA recognition.

4.5.4 Decision tree

The decision tree classifier is characterized by the rule that an unknown sample is classified into a class using one or several decision functions in a successive manner (Swain et al., 1977). This classification strategy can be described by a tree diagram. In general, a decision tree consists of a root node, a number of interior nodes, and a number of terminal nodes. The root node and interior nodes, together referred to as non-terminal nodes, are linked into decision stages. The set of nodes at a given level in the tree is called a layer. The terminal nodes are associated with the entire set of classes into which a sample may be classified. In the tree, each node consists of a set of classes to be discriminated, the set of features to be used, and the decision rule to perform the classification. Decision Tree is similar to the human decision-making process and so that it is easy to understand. An example of decision tree on what to do when different situations occur in weather can be seen in Figure 4.8.

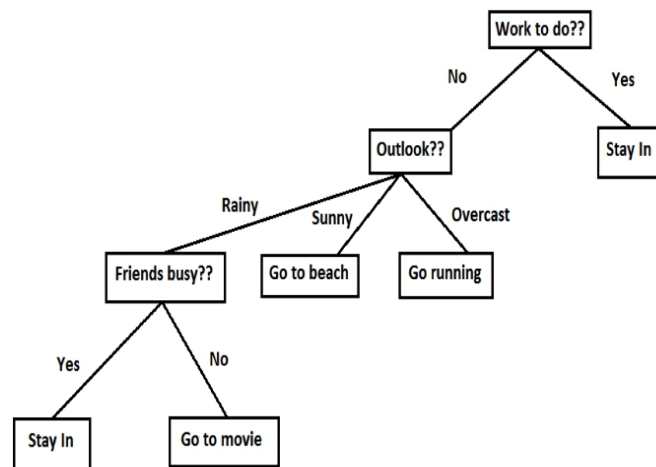


Figure 4.8 Decision Tree example of what to do when when different situation happens in weather (Patel et al., 2018)

The decision tree can be designed in two ways: by the manual design procedure and the heuristic search procedure (Swain et al., 1977). The manual design of trees relies on expert knowledge and only provides optimal results in extremely simple cases (relatively few classes, easily discriminated with a small number of features). Heuristic search described as ‘guided search with forward pruning’ is more suitable for complex problems. This method applies an evaluation function to direct a search through the decision tree. At each stage, the function selects the node with the highest evaluation measure, which is usually a weighted measure of classifier efficiency and accuracy. The generated decision trees can be further pruned to meet the trade-off between error rate and tree size. Figure 4.9 shows a decision tree generated from physical activity data to classify 7 classes shown on the leaves nodes based on the mean and standard deviation of the vertical and horizontal components from the accelerometer sensor.

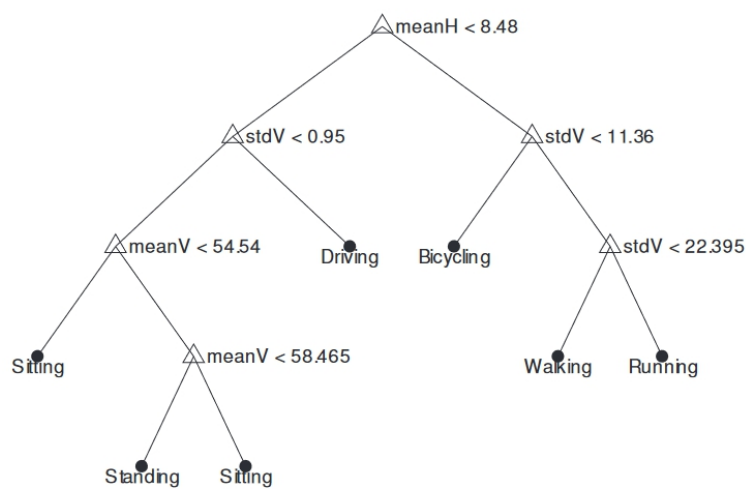


Figure 4.9 Decision tree generated from PA data (Yang, 2009)

There are different types of decision tree algorithms used to split the attributes to test at any node to determine whether splitting is the best in individual classes. The common algorithms are ID3, C4.5, CART, etc. The characteristics of DT algorithms are given in Table 4.15 (Harsh et al., 2018).

Table 4.15 Decision Tree Algorithms

Decision tree algorithm	data types	Numerical data splitting method
CHAID	Categorical	N/A
ID3	Categorical	No restriction
C4.5	Categorical, Numerical	No restriction
CART	Categorical, Numerical	Binary splits

4.5.5 Extra tree

Extra Tree Classifier is also known as extremely randomized trees. It is an ensemble method based on the decision tree classifier. Extra Tree Classifier is the modification of bagging where samples of the training dataset are used to construct the random trees. The graphical representation of the working of this method is given in Figure

4.10. In one Extra Tree Classifier, several different decision trees are clustered into a forest from a single learning set. This method in general provides high performance but sometimes suffers from over-fitting problems due to high inter-dependency among hyperparameters during model building. The hyperparameters can influence the performance of the classifier at large, such as the number of randomly selected attributes at each node, the minimum sample size for splitting a node, the number of decision trees for the ensemble (Padmaja et al., 2020). Extra Tree Classifier is not as popular as the aforementioned supervised learning methods in PA recognition. Padmaja et al. (2020) proposed a novel random-split-point procedure for Extra Tree Classifier in the classification of PA, and achieved an accuracy of 94.16% and 92.63% for two datasets respectively.

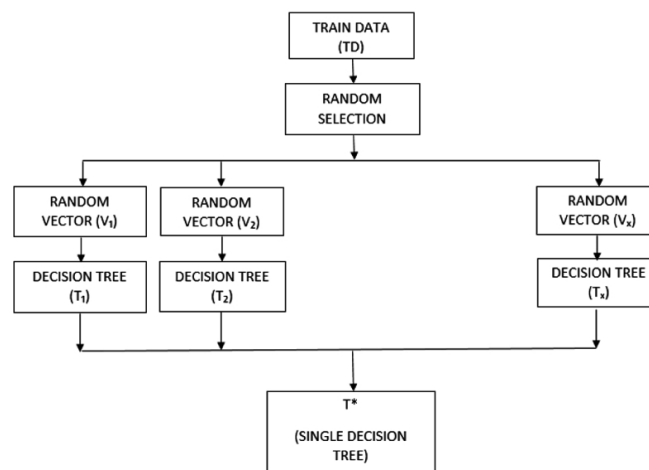


Figure 4.10 Illustration of Extra Tree Classifier working (Bhati et al., 2020)

4.5.6 Random forest

The Random Forest (RF) is a very popular classification and regression algorithm. The RF algorithm belongs to ensemble learning methods. Likewise a regular forest consists of a number of trees, the RF algorithm consists of a number of classification or regression trees (CART). The algorithm does not use all features for the CART construction but only a few of them. The RF was designed by LeoBreiman in 2001. In this paper, the author compared the RF with other ensemble techniques and mentioned that this method had higher accuracy than e.g. Adaboost method (Breiman, 2001). Since that time the RF is used in bioinformatics, medical informatics and so on. The algorithm is resistant to outliers, missing values, or noise. The RF stands out especially in simplicity of parameter tuning but the main problem is its interpretability. For optimal settings of the algorithm only two parameters have to be set: the number of trees in the forest and the number of variables in trees. (Peterek et al. 2014)

There are many similarities between extra tree and random forest classifiers. Both ensembles are composed of a large number of decision trees, and the final decisions are obtained on the basis of the prediction of every tree. Further, both algorithms have the same growing tree procedure. Moreover, when selecting the partition of each node, both of them randomly choose a subset of features. The main difference between these two methods is that random forest uses bootstrap methods, which means that it subsamples the input data with replacement, while extra trees use the whole original sample. Another difference is that when splitting a node, the random forest classifier

chooses the optimum split while extra tree chooses a random split. In this sense, extra tree reduces bias and variance compared to random forest. Besides, extra tree is also faster in terms of computational cost. This is because it selects the split randomly than looking for the optimum split.

4.5.7 K-Nearest Neighbours

The K-nearest neighbours (KNN) is a non-parametric algorithm for classification. The KNN is one of the easiest methods for data classification. Given the training set T and the testing sample x_i , the KNN classifier tries to find sample x_r from the training set T , with a minimal Euclidean distance to the testing sample. Better results are achieved if more than one sample from the training set are found. This algorithm achieves satisfactory results but is not suitable for solving difficult tasks (Peterek et al. 2014).

5 Results

5.1 preprocessing

Figure 5.1 shows the first 3000 observations of the total acceleration signal before and after the removal of the gravity part. The scripts for the low pass filter can be found in Table (Appendix 8.2). It can be seen that before applying the low pass filter, the total acceleration fluctuates around a value above zero, after the application of the low pass filter, the acceleration fluctuates around the zero axis. Figure 5.2 shows the GPS speed of the first 300 observations before and after the recalculation. It can be seen that the recalculated signal is more smooth than the original one, and the speed changing patterns are more clear and observable.

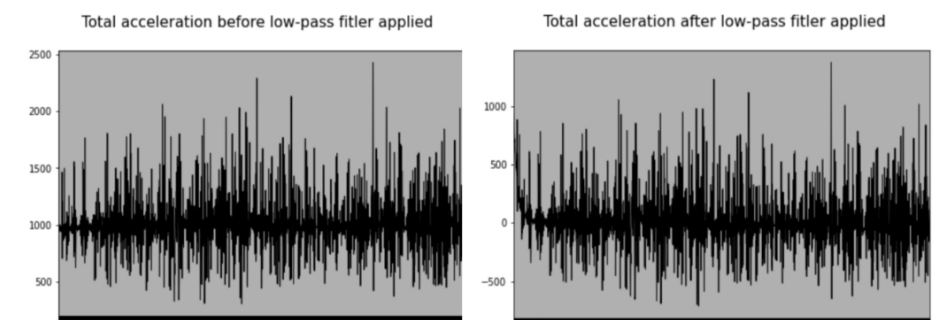


Figure 5.1 Acceleration signal before and after applying the low-pass filter

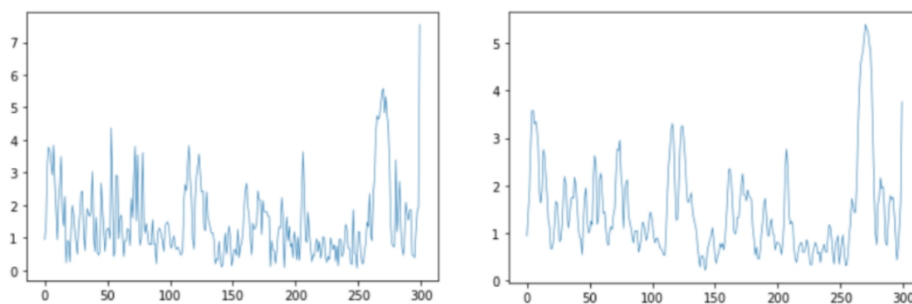


Figure 5.2 GPS speed before (left) and after (right) the recalculation

5.2 Feature calculation

This section shows the features calculation results which will be used to train the classifiers. Figure 5.3 shows the comparison of total acceleration for different activity types in the window size of 180 observations. Figure 5.4 shows the box-plot for the total acceleration comparison for different activity types. The box shows the quartiles

of the dataset while the whiskers extend to show the rest of the distribution. One can clearly see from the figures that jogging has the largest amplitude and variation of the total acceleration signal, followed by ascend and descend. The third largest amplitude types are slow walk and fast walk. The stationary type has close-to-zero value total acceleration. However, one can observe one outlier from the box plot. This suggests the necessity of a more comprehensive feature set for the description of the signals. In this case, the maximum value of the signal will not be a significant feature to recognize stationary activity from others. Also, it is noteworthy that from the box plot, slow walk has a slight higher acceleration amplitude and variation than fast walk. This seems to not correspond to the common sense. But in a large dataset, this slight difference might not be significant in the recognition of the two activities. Also, the other dimensions of the sensors, such as speed from GPS sensors might have more influence in the recognition in this case. Overall speaking, the total acceleration amplitude can be used to distinguish jogging from other activities from the figure 5.4. Among other activities, this feature is not enough for the recognition.

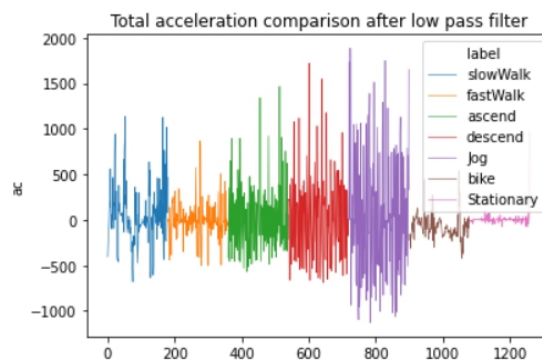


Figure 5.3 Total acceleration signal comparison for different activity types

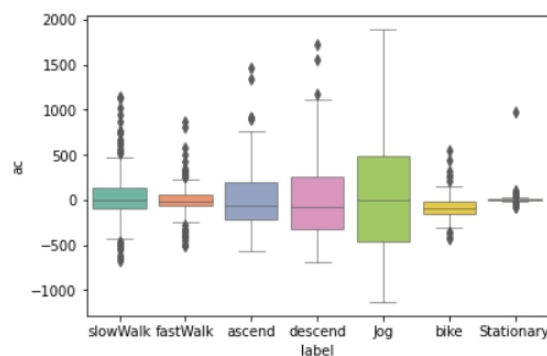


Figure 5.4 Total acceleration box-plot comparison for different activity types

Figure 5.5 shows the speed comparison for different activities in a window size of 180 observations. Figure 5.6 shows the speed comparison in box plot. One can see that biking has the highest speed among other activities. The activity with the second highest speed is jogging, and stationary has the lowest speed. Among the rest for activities, fast walk has a larger speed range. The box plot clearly illustrates that with the speed range feature, one can distinguish biking, jogging, stationary from the other four activities.

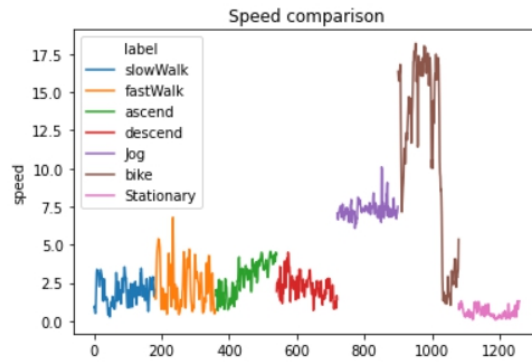


Figure 5.5 Speed comparison for different activity types

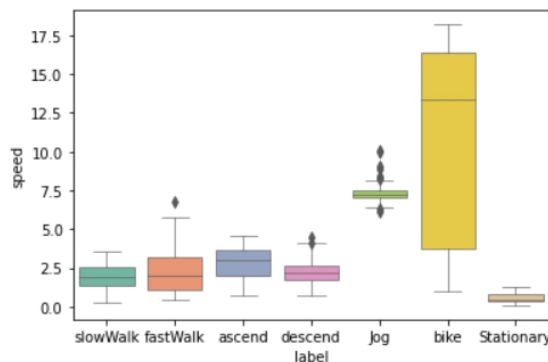


Figure 5.6 Speed box-plot comparison for different activity types

Figure 5.7 shows the number of peaks and the peak heights for each activities in the window size of 180 observations. A peak, or a local maximum of a signal is defined as any sample whose two adjacent neighbours have a small amplitude. The function used to find the peaks takes the signal array and finds all local maxima by simple comparison of neighbouring values (see Appendix 8.2). Two conditions are defined for the peak's property in this thesis. The *threshold* condition defines the minimal vertical distance to the samples' neighbouring samples. The *distance* condition defines minimal horizontal distance in samples between neighbouring peaks. The *threshold* and *distance* conditions are set as 20 and 2 respectively in this thesis by visual calibration of the first widowed-segment for each activity types. This will lead to a more successful detection of the majority of peaks. The peak height or prominence measures how much a peak stands out from the surrounding baseline of the signal and is defined as the vertical distance between the peak and a higher one of the two minimal signal value in range for the peaks. It can be seen that Stationary status has the smallest number of peaks and lowest peak heights. Jogging has the highest peak height, followed by the ascend and descend activities.

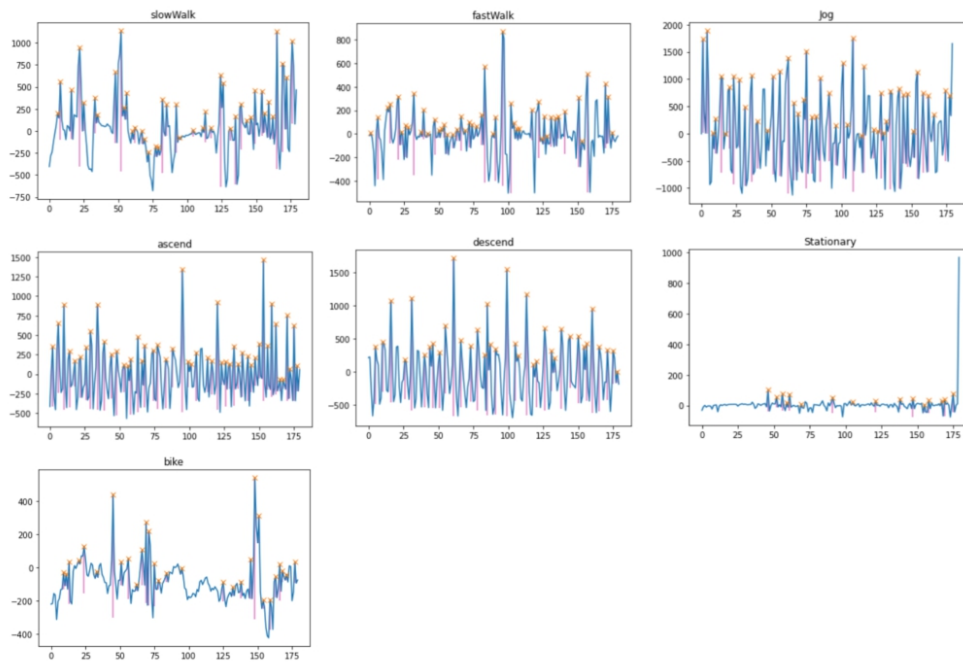


Figure 5.7 Activity-wise peaks and peak heights

Figure 5.8 shows the activity-wise peak width in green and peak interval in red in the widow size of 180 observations. The width of a peak is calculated at half of the height of the peak. The interval measures the distance of the adjacent peaks. It can be seen that jogging has the most narrowed peak width and intervals, while biking and stationary have relatively longer widths and intervals.

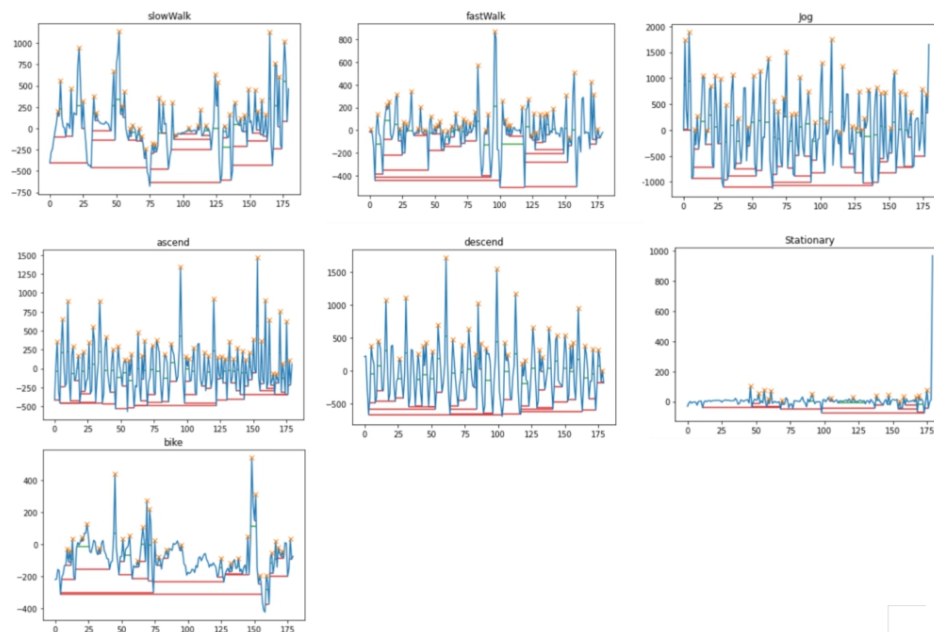


Figure 5.8 Activity-wise peak width and time intervals

Figure 5.9 shows the activity-wise cross correlation results of the total acceleration. It can be seen that slow walk, stationary and bike can be more easily detect by this

feature. The cross-correlation results of other dimensions of the signal (x-axis, y-axis, z-axis) can be found in Appendix 8.2.

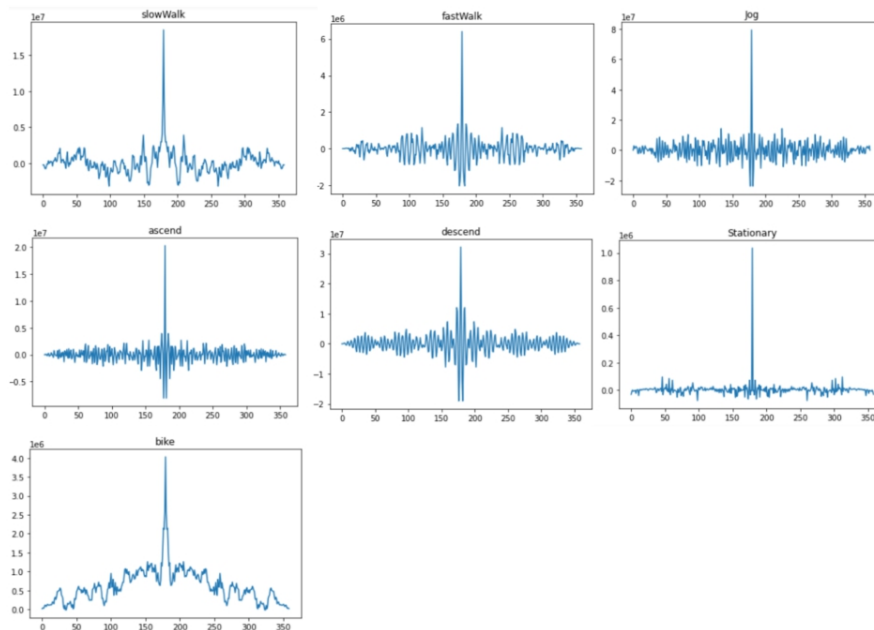


Figure 5.9 the activity-wise cross-correlation results of the total acceleration

5.3 Segmentation

As aforementioned, this thesis applies the sliding-widow segmentation in the window sizes of 90s and 180s. The windows are tested both with and without a 50% overlap size. As a result, four widow segmentation strategies are tested for this data, and the one with the best performance is selected in the end in the comparison process of the feature selection methods. Figure 5.10 and Figure 5.11 display the preliminary classification results for a set of common classifiers. The classification takes the stratified k fold strategy with k equals to 10. The parameters for each classifier in this step are set as the default parameter in python, and can be found in Appendix. The classifiers are ranked by MCC score, the best result in each evaluation metrics is highlighted. One can see that the smaller window size (90 samples), and the overlap (50%) result in higher classification performance in general. Therefore, this thesis takes the sliding widow of 90 samples with 50% overlap for classification.

Model		Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT(Sec)
rf	Random Forest Classifier	0.76	0.94	0.71	0.77	0.76	0.70	0.70	1.43
et	Extra Trees Classifier	0.74	0.94	0.69	0.75	0.73	0.67	0.67	1.31
dt	Decision Tree Classifier	0.70	0.81	0.68	0.71	0.70	0.63	0.63	0.21
nb	Naive Bayes	0.29	0.69	0.40	0.45	0.29	0.20	0.22	0.03
knn	K Neighbors Classifier	0.35	0.65	0.33	0.35	0.35	0.20	0.20	0.87
svm	SVM - Linear Kernel	0.21	0.00	0.19	0.09	0.11	0.04	0.06	0.13

Model		Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT(Sec)
rf	Random Forest Classifier	0.81	0.96	0.77	0.82	0.81	0.76	0.76	2.70
et	Extra Trees Classifier	0.81	0.96	0.77	0.81	0.81	0.76	0.76	2.23
dt	Decision Tree Classifier	0.75	0.84	0.72	0.76	0.75	0.69	0.69	0.54
nb	Naive Bayes	0.32	0.69	0.42	0.45	0.33	0.22	0.25	0.06
knn	K Neighbors Classifier	0.38	0.67	0.35	0.37	0.37	0.22	0.23	0.76
svm	SVM - Linear Kernel	0.23	0.00	0.19	0.12	0.12	0.05	0.07	0.46

Figure 5.10 classification results without feature selection in the window size of 180 (without overlap (left), with overlap(right))

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	0.78	0.95	0.72	0.78	0.77	0.72	0.72	2.23
et	0.75	0.94	0.70	0.76	0.75	0.68	0.68	1.85
dt	0.72	0.82	0.69	0.72	0.72	0.65	0.65	0.48
nb	0.34	0.69	0.45	0.47	0.34	0.24	0.27	0.04
knn	0.32	0.62	0.29	0.31	0.31	0.15	0.15	0.48
svm	0.21	0.00	0.15	0.12	0.10	0.01	0.01	0.29

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	0.83	0.97	0.78	0.84	0.83	0.78	0.78	4.01
et	0.82	0.97	0.78	0.82	0.82	0.77	0.77	2.86
dt	0.80	0.88	0.79	0.81	0.80	0.76	0.76	1.27
nb	0.31	0.70	0.42	0.44	0.32	0.22	0.24	0.05
knn	0.33	0.63	0.31	0.32	0.32	0.17	0.17	0.69
svm	0.18	0.00	0.17	0.10	0.07	0.02	0.03	0.66

Figure 5.11 classification results without feature selection in the window size of 90 (without overlap (left), with overlap (right))

The preliminary results also give an idea about the best classifiers for this data. Among the classifiers, random forest performs the best in terms of a variety of evaluation metrics, closely followed by the Extra Tree Classifier and the decision tree classifier. However, decision tree takes way less time than the other two, and extra tree is also faster in general than random forest.

An overview of the confusion matrix performance results of PA detection with other classifiers besides extra tree can be found in Appendix 8.1.

5.4 Dimension Reduction

In this thesis, a total of one feature transformation and four feature selection methods are implemented for PA recognition. Three classifiers are selected to combine the dimension reduction methods. Kmeans is selected first among the unsupervised classifier and combined with PCA considering the inner connection of these two commonly used methods (Ding 2004). However, as the result is not ideal, another supervised learning method KNN (El Moudden 2016) is applied with PCA to further check PCA's influence on the classification performance. For feature selection methods, one filter method ReliefF, one sequential wrapper method REFCV, one heuristic wrapper method Genetic algorithm, a one embedded method random forest are implemented. It is noteworthy that the embedded method random forest is itself a classifier. And the other feature selection methods are combined with extra tree, the second best classifier regarding the performance beside random forest.

The validation method for the comparison is stratified 10-fold validation. The reason is that 10-fold validation is commonly used in literature. And stratified k-fold that splits the data according to class proportions can deal with unbalanced data well. Besides, 10-fold validation, the hold-out validation method with 10% data as the test data is also applied on the KNN classifier to compare the differences of the results from different validation methods. The performance is examined and ranked by accuracy score, as it is the most basic metric in literature. Other metrics are also given for a more comprehensive understanding of the models' performances. The performance metrics shown are the averaged metric scores of the 10 folds. All classification, dimension reduction, and validation functions are performed at the random state id of 42.

5.4.1 Feature transformation

In this thesis, the PCA method is selected for the feature transformation process. The Kmeans classifier is applied as the unsupervised classifier and KNN is applied as the supervised classifier.

5.4.1.1 K-means with PCA

For the unsupervised K-means classifier, a grid search with the value of 3, 10, 20 is applied for the parameter for the number of times the k-means algorithm will be run with different centroid seeds. For the PCA method, the hyper-parameter for the number of components is also implemented by the grid search with the value of 7, 16, 60, 100. Therefore, a total combination of 12 models are run. Table 5.1 shows the results of the K-means with PCA by the stratified 10-fold classification validation method. In the table, the *number of dimensions* column represents the parameter for the PCA method and the *number of time* column represents the parameter for the KMeans classifier. In the accuracy column, the left numbers represent the classification performance with the PCA feature transformation, the right numbers in the brackets represent the classification performance without the PCA feature transformation. Overall speaking, one can see that the K-means classifier provides rather poor classification results as expected, though PCA indeed helps in increasing the accuracy score to roughly one time more. Besides, by tuning of the two hyper-parameters, there are no significant changes shown in the results.

Table 5.1 Classification Results for PCA combined with K-means

Number of Dimensions	Number of time K-means run	Accuracy
7	3	0.218 (0.1155)
7	10	0.224 (0.1151)
7	20	0.222 (0.1153)
16	3	0.218 (0.1155)
16	10	0.223 (0.1151)
16	20	0.223 (0.1153)
60	3	0.218 (0.1155)
60	10	0.220 (0.1151)
60	20	0.221 (0.1153)
100	3	0.221 (0.1155)
100	10	0.223 (0.1151)
100	20	0.220 (0.1153)

5.4.1.2 KNN with PCA

In the second step, PCA is also combined with KNN, as the combination is proved to show acceptable results in some researches (El Moudden et al. 2016, Peterek et al. 2014). In the process, two hyper-parameters are tuned, namely *number of dimensions transformed* for the PCA algorithm and the *number of neighbours* for the KNN classifier. Table 5.2 shows the classification results by unsupervised KNN classifier after feature transformation by the PCA method validated by stratified 10-fold classification and holdout validation (90%/10% split). In the *accuracy* column for stratified 10-fold classification, the left numbers represent the classification performance with the PCA feature transformation, the right numbers in the brackets represent the classification performance without the PCA feature transformation. The entries with the highest observations from different metrics are highlighted.

Table 5.2 Classification results for PCA and KNN

Number of Dimensions	Number of neighbors	Accuracy (stratified 10-fold validation)	Accuracy (holdout validation 10% test)
7	3	0.683 (0.327)	0.670
7	10	0.696 (0.352)	0.694
7	20	0.690 (0.354)	0.683
16	3	0.721 (0.327)	0.710
16	10	0.720 (0.352)	0.723
16	20	0.705 (0.354)	0.702
60	3	0.745 (0.327)	0.75
60	10	0.742 (0.352)	0.751
60	20	0.722 (0.354)	0.730
100	3	0.744 (0.327)	0.753
100	10	0.741 (0.352)	0.752
100	20	0.723 (0.354)	0.728

One can see that PCA method improves the classification performance of the KNN classifier significantly more than one times. And the difference exists between the two validation methods. For the 10-fold validation, the best result is from the combination of 60 dimensions for PCA and 3 neighbors for KNN, while for the holdout validation, the best result is from the combination of 100 dimensions and 3 neighbors. And the holdout validation method exerts a slightly higher accuracy. Also, the 10-fold validation shows that the figure of 3 and 10 for number of neighbors give similar results, while holdout validation method shows that 3 neighbors provides better result than 10 neighbors. In terms of the number of dimensions, both validation methods indicate that the higher the number of dimensions, the better the performance. However, for the 10-fold validation, 60 dimensions and 100 dimensions exert similar performance. While based on holdout validation, 100 dimensions perform better than 60 dimensions. Also, the accuracy scores from KNN without PCA transformation suggest that the higher the number of neighbors, the better the performance, which is contrary from the results from KNN with PCA transformation. This suggests the necessity of parameter tuning when training classification models.

Table 5.3 and Table 5.4 show more detailed results of the two validation methods. From Tables 5.3, for stratified k-fold validation, it can be seen that within the range of 7, 16, 60, and 100 of dimensions, and 3, 10, and 20 of neighbours, the combination of 3 neighbours for the KNN classifier and the number of 60 final transformed dimensions provides the best accuracy result. The best result yields a accuracy score of 0.745 compared to 0.327 without the PCA feature selection process. This combination also yields best or close to best results for other metrics, including Recall, F1-score, MCC, Precision, while it yields relatively lower AUC score.

Overall speaking, other combinations also yield significant better results than the classification before feature transformation. The time costs for some of the combinations are significantly higher than others. For example, the parameter combination with the highest accuracy also has the highest time cost. This indicates the importance to balance classification performance and computational cost when using feature transformation methods. In addition, the result shows that for the analysis in this thesis, in general the higher dimensions and the lower number of neighbours provides a higher performance.

Table 5.3 Classification results for PCA and KNN (stratified 10-fold)

Dimension	Number of Neighbors	Accuracy	AUC	Recall	Precision	F1	KAPPA	MCC	Time spent(s)
7	3	0.683	0.875	0.683	0.683	0.680	0.606	0.607	11.05
7	10	0.696	0.911	0.696	0.700	0.695	0.620	0.620	7.63
7	20	0.690	0.912	0.690	0.696	0.689	0.610	0.611	10.50
16	3	0.721	0.899	0.720	0.724	0.719	0.654	0.655	14.12
16	10	0.720	0.926	0.705	0.731	0.720	0.650	0.651	15.99
16	20	0.705	0.924	0.745	0.713	0.704	0.630	0.631	16.80
60	3	0.745	0.914	0.745	0.745	0.743	0.684	0.685	100.79
60	10	0.742	0.934	0.742	0.746	0.742	0.691	0.679	79.87
60	20	0.722	0.931	0.722	0.728	0.721	0.678	0.652	100.23
100	3	0.744	0.914	0.744	0.744	0.742	0.651	0.683	70.81
100	10	0.741	0.935	0.741	0.745	0.740	0.676	0.677	86.73
100	20	0.723	0.933	0.723	0.729	0.722	0.653	0.654	77.82

From Table 5.4 for holdout validation result, it can be seen that within the range of 7, 16, 60, and 100 of dimensions, and 3, 10, and 20 of neighbours, the combination of 3 neighbours for the KNN classifier and 100 final transformed dimensions provides the best result considering all metrics. Overall speaking, the scores by these two validation methods are at a similar level for different metrics and combinations. Different from the stratified 10-fold validation, the time costs for all the combinations are also relatively low. This indicates that this validation method can help in balancing the classification performance improvement and computational cost in feature transformation process.

Table 5.4 Classification results for PCA and KNN (holdout validation)

Dimension	Number of Neighbors	Accuracy	AUC	Recall	Precision	F1	KAPPA	MCC	Time spent(s)
7	3	0.670	0.867	0.670	0.670	0.669	0.593	0.594	0.295
7	10	0.694	0.908	0.694	0.707	0.696	0.620	0.621	0.319
7	20	0.683	0.912	0.683	0.699	0.682	0.604	0.606	0.279
16	3	0.710	0.903	0.710	0.711	0.708	0.642	0.643	0.345
16	10	0.723	0.926	0.723	0.731	0.723	0.656	0.656	0.385
16	20	0.702	0.921	0.702	0.718	0.702	0.630	0.631	0.401
60	3	0.75	0.923	0.75	0.750	0.748	0.691	0.692	1.046
60	10	0.751	0.937	0.750	0.760	0.751	0.691	0.692	1.162
60	20	0.730	0.936	0.730	0.737	0.729	0.664	0.665	1.244
100	3	0.753	0.923	0.753	0.754	0.752	0.696	0.697	1.553
100	10	0.752	0.939	0.752	0.762	0.752	0.693	0.694	1.875
100	20	0.728	0.935	0.728	0.736	0.727	0.662	0.663	1.843

Further, two confusion matrices comparisons between the classification results with and without feature transformation, and the classification results by two types of validation methods are conducted. Figure 5.12 shows the confusion matrix without feature transformation by the KNN classifier with the *number of neighbors* parameter set as 3.

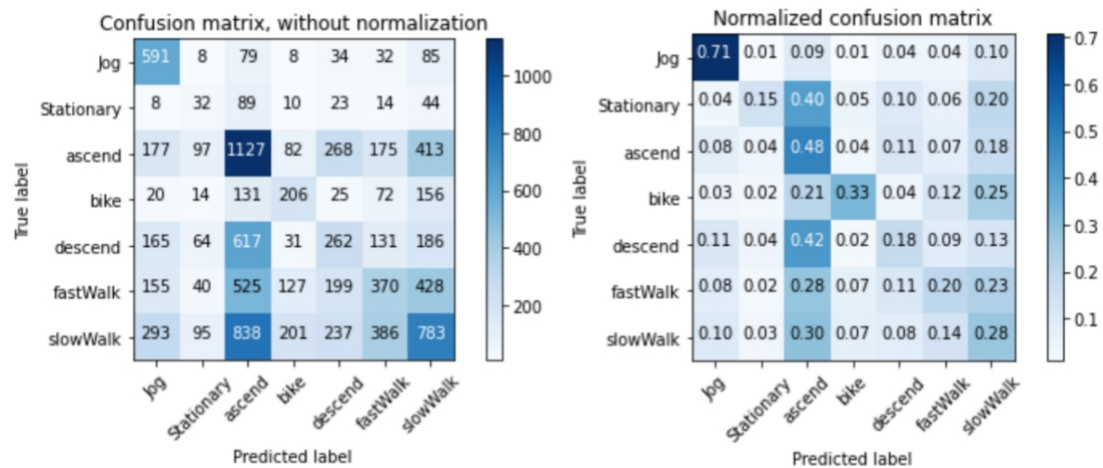


Figure 5.12 Confusion matrix for KNN without PCA (Stratified-10-fold)

In Figure 5.12, the left side shows the confusion matrix without normalization, in which the numbers in the cells represent the number of segments for each activities. The x-axis represents the predicted label while the y-axis represents the true label. The darker the colours of the cells, the larger the numbers in the cells. One can see that the majority of the misclassification incidences locate at classes with larger number of segments. Due to the unbalanced data distribution among the class and the large numbers, a normalized confusion matrix is given on the right side. In the normalized confusion matrix, the numbers in the cells represent the proportions of data accurately classified and misclassified for each class. The darker the colours, the higher the proportions of classes.

One can see that for KNN classifier without PCA transformation, Jogging is the activity that has the highest classification score with a value of 0.71, and the Stationary has the lowest accuracy score with a value of 0.15. The activity with second highest accuracy is Ascending. Jogging is most often misclassified as Ascending (0.09), and while Ascending is most often misclassified as Fast walk (0.18).

Simultaneously, Fast walk is also most often detected as Ascending (0.28). Overall looking, all activities, including Descending, Biking, Stationary and Slow walk are more often classified as Ascending by the KNN method. The second class that are misclassified most as is Slow walk. This suggests the drawback of the KNN method on this dataset as the classes are likely to be classified in several main classes with larger number of observations, such as Slow walk, Fast walk and Ascending. However, KNN is good at recognising the Jogging activity, though Jogging is not one of the classes with most observations. It can also be observed from the colours of the cells in both confusion matrix. And this could be the main reason for very unbalanced accuracy scores in different classes.

Figure 5.13 shows the confusion matrix from the KNN classifier with PCA transformation. From the figure, one can see that in this case, the most accurately classified classes are Jogging, Ascending and Biking. Descending and Slow walk also have an good accuracy level with scores above 0.7. Stationary and Fast walk have relatively lower accuracy. In this figure, one can still notice the Ascending and Slow walk are still the two most classes the other classes being misclassified as. However,

the result is large improved. A closer look at the comparison of KNN classifier with and without PCA can be seen in Figure 5.14.

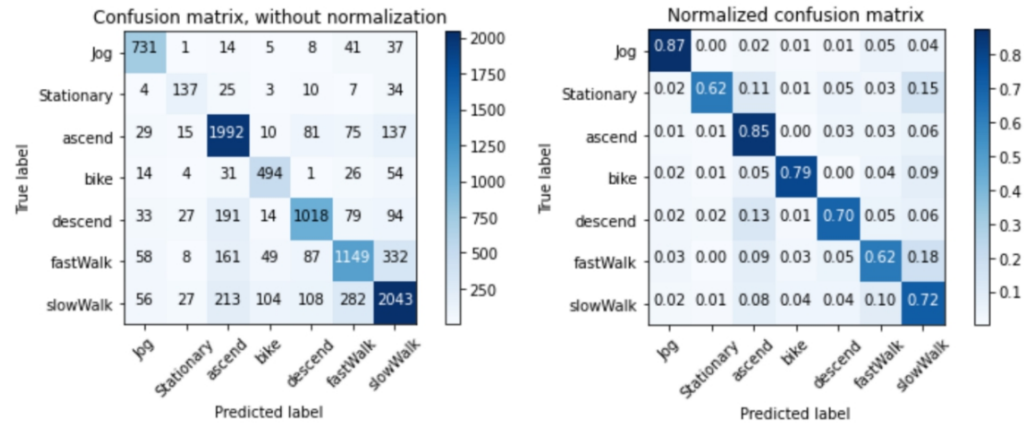


Figure 5.13 Confusion matrix for KNN with PCA

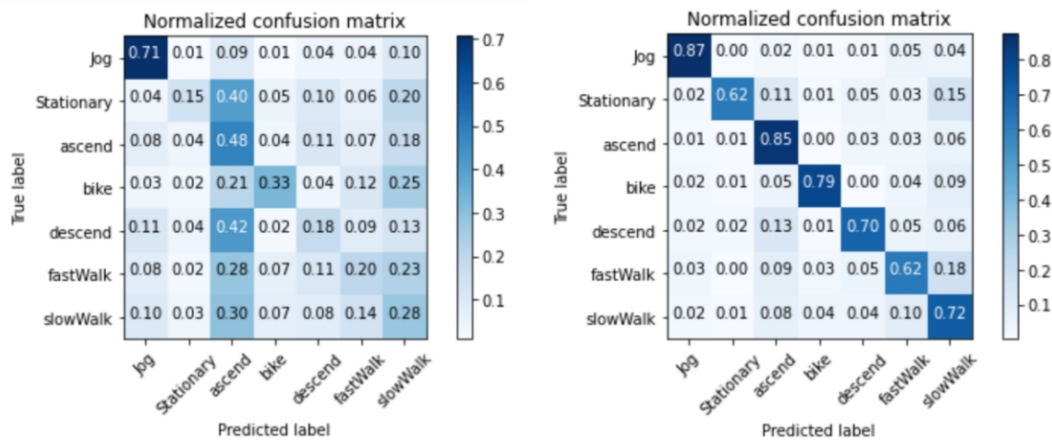


Figure 5.14 Comparison of confusion matrix for KNN with (Right) and without (Left) PCA

Figure 5.14 shows the normalized confusion matrices comparison for KNN with and without the PCA transformation. One can clearly see that PCA improves the classification results of the KNN classifier in all classes. Especially, it helps in increasing the recognition in Jogging to an accuracy score of 0.87 by reducing the errors being classified as Ascending. The result proves that PCA is useful in increasing the performance of the KNN classifier.

Another comparison of the confusion matrix is the comparison of k-fold stratified validation and holdout validation as described in the tables above. Figure 5.15 shows the confusion matrix of the best result (parameter setting as 100 dimensions and 3 neighbors) from the holdout validation. And Figure 5.16 shows the normalized confusion matrices comparison of the best results by stratified k-fold validation (parameter setting as 60 dimensions and 3 neighbors) and k-fold validation (parameter setting as 100 dimensions and 3 neighbors).

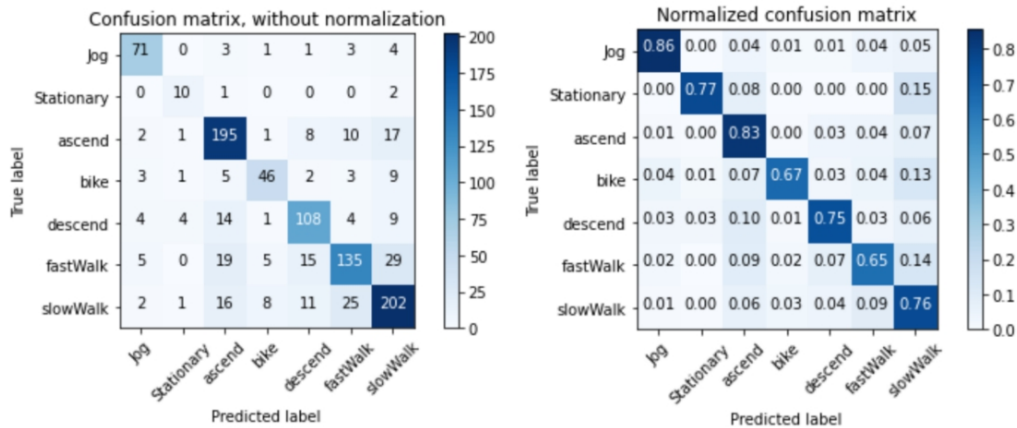


Figure 5.15 Confusion matrix of KNN with PCA by holdout validation

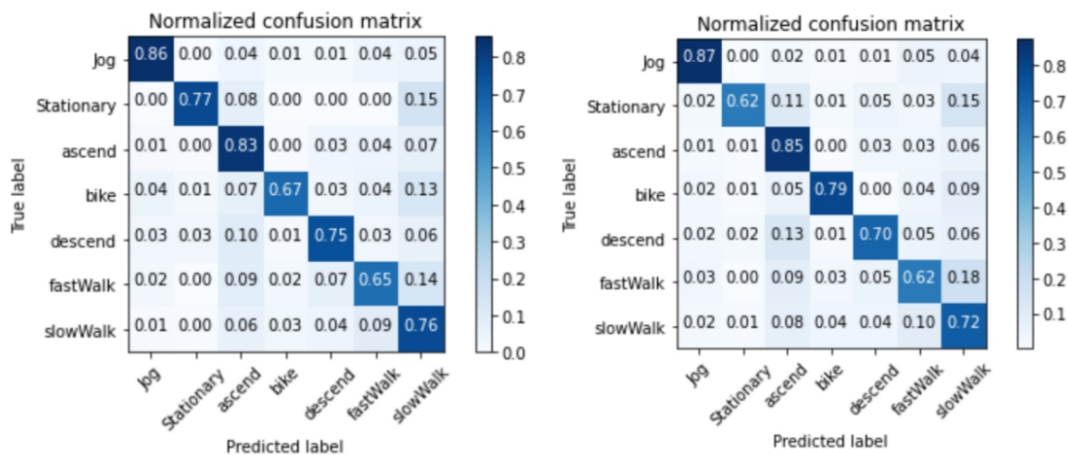


Figure 5.16 Confusion matrix of KNN with PCA by Stratified 10-fold (Right) and holdout (Left) validation

From the Figure, it can be seen that the misclassification from holdout validation is rather balanced within different classes. By this methods, all classes in general are still likely to be classified as Ascending and Slow walk. In this comparison, the accuracy for Stationary increases by the holdout method while for Biking decreases to a certain degree compared to the stratified 10-fold validation. However, in general, the differences of the accuracy scores between the two validation methods are not huge. Though it might result in different hyper-parameter selections when training the models. Besides, from the matrix, one can see that the largest misclassification proportions are Stationary, Fast walk, and Biking to Slow walk. From real life experience, it is understandable that stationary and fast walk being classified as slow walk, as these two activities exert similar characteristics as slow walk. With regard to biking, the reason could be that original real-life biking segments do not always exert typical characteristics for biking, such as a relatively high speed, which is observed in the manual labelling process.

5.4.2 Feature Selection

In this section, three feature selection algorithms from different feature selection categories are compared in terms of their influences on the ExtraTree classifier.

The ExtraTree classifier in this comparison is realized by the ExtraTreesClassifier function from sklearn.ensemble module. The parameter setting for the ExtraTreesClassifier function applied in this thesis can be found in Appendix 8.2. In this analysis, the random state id is set as 42.

5.4.2.1 ReliefF

ReliefF is selected as the filter feature selection method. The ReliefF function is realized by the ReliefF module from python (See Appendix 8.2). In this function, two parameters can be tuned, namely the *number of neighbors* for the quality estimation of features and the *maximum number of features*. For this purpose, two grid searches are implemented, namely the value of 3, 10, 20 for the *number of neighbors*, and the value of 16, 30, 60, 100 for the *number of maximum features*.

As aforementioned, the user-defined tuning parameter *Number of neighbors* controls the locality of the estimates. Though it is claimed that for most of the time, this parameter can be safely set to 10 (Robnik-Šikonja et al. 2003). In this study, the higher values, such as 60, 100 are still implemented for a more comprehensive inspection of sub-feature sets. In summary, a total of 12 combination models are run. Table 5.5 shows the result of the ExtraTree Classifier with ReliefF by the stratified 10-fold classification validation method. In the table, both the *number of dimensions* (maximum number of features) column and the *number of neighbors* are parameters for the ReliefF feature selection method. In the table, the numbers in the first row in the brackets represent the classification performance without the ReliefF feature selection, the rest rows represent the classification performance with the ReliefF feature selection method. The entries with the highest observations from different metrics are highlighted.

Table 5.5 Classification results for ExtraTree and ReliefF (stratified 10-fold)

Dimension	Number of Neighbors	Accuracy	AUC	Recall	Precision	F1	KAPPA	MCC	Time spent(s)
(Without ReliefF method)		(0.806)	(0.962)	(0.805)	(0.813)	(0.806)	(0.756)	(0.757)	(28.75)
16	5	0.772	0.948	0.772	0.779	0.770	0.713	0.715	43.57
16	10	0.765	0.947	0.771	0.778	0.769	0.712	0.713	40.47
16	20	0.772	0.949	0.772	0.778	0.771	0.714	0.715	39.36
30	5	0.785	0.957	0.774	0.803	0.793	0.741	0.738	32.74
30	10	0.790	0.955	0.790	0.798	0.789	0.736	0.746	37.12
30	20	0.785	0.956	0.790	0.799	0.789	0.737	0.733	31.78
60	5	0.814	0.965	0.814	0.822	0.813	0.736	0.768	38.71
60	10	0.817	0.964	0.812	0.820	0.811	0.767	0.765	31.78
60	20	0.825	0.965	0.812	0.820	0.812	0.764	0.766	35.14
100	5	0.797	0.962	0.801	0.809	0.800	0.750	0.751	38.53
100	10	0.806	0.961	0.804	0.812	0.804	0.754	0.756	35.57
100	20	0.815	0.962	0.802	0.810	0.802	0.751	0.754	43.27

It can be seen from the table that the reliefF method is able to improve the Extra Tree Classifier performance. However, it depends on the parameter settings for the feature

selection method, as with some settings the performances are even lower than the classification without the feature selection. The combination that has the highest accuracy is 60 features and 20 neighbors for the reliefF method with the value of 0.806. Besides, it is observable that the best *number of dimension* is 60, while the *number of neighbors* does not show a significant difference in performances.

Table 5.6 shows the results by the holdout validation method with the 90%/10% train/test split. The same as Table 5.5, the first row shows the performance result without the reliefF method. One can see that from this method, the performance score is higher in general. The best combination is 100 dimensions with 20 neighbors, which exerts an accuracy performance of 0.850 compared to 0.818 without the reliefF feature selection. And the best number of dimension is 100 compared to the stratified K-fold validation method. In terms of the *number of neighbors*, this method also does not show any obvious differences as well. This corresponds to the comparison results from the last subchapter of PCA and KNN. Therefore, a good estimate of *the number of neighbors* for distance-related methods could be related to the data quality itself. And the estimation for a proper number of features could be related to both the data itself and the validation method chosen. Regarding the time cost, as expected, the holdout method spends less time than the Stratified method.

Table 5.6 Classification results for ExtraTree and ReliefF (holdout validation)

Dimension	Number of Neighbors	Accuracy	AUC	Recall	Precision	F1	KAPPA	MCC	Time spent(s)
(Without ReliefF method)		(0.818)	(0.962)	(0.818)	(0.824)	(0.819)	(0.772)	(0.773)	(3.08)
16	5	0.784	0.949	0.779	0.785	0.778	0.723	0.724	12.55
16	10	0.773	0.949	0.773	0.781	0.773	0.716	0.717	12.58
16	20	0.777	0.949	0.777	0.783	0.777	0.721	0.722	12.87
30	5	0.798	0.956	0.798	0.813	0.799	0.746	0.749	12.69
30	10	0.796	0.959	0.796	0.809	0.796	0.743	0.746	13.42
30	20	0.785	0.955	0.785	0.796	0.785	0.730	0.733	13.15
60	5	0.822	0.970	0.822	0.831	0.822	0.777	0.779	13.02
60	10	0.831	0.971	0.831	0.837	0.828	0.788	0.790	12.93
60	20	0.828	0.969	0.828	0.834	0.729	0.785	0.786	13.36
100	5	0.841	0.975	0.841	0.850	0.841	0.801	0.803	13.58
100	10	0.847	0.976	0.847	0.855	0.847	0.808	0.810	13.96
100	20	0.850	0.976	0.850	0.857	0.849	0.812	0.814	14.79

Further, two confusion matrices comparisons between the classification results with and without feature transformation, and the classification results by two types of validation methods are conducted. Figure 5.17 shows the confusion matrix without feature transformation for the Extra Tree Classifier by the stratified 10-fold validation.

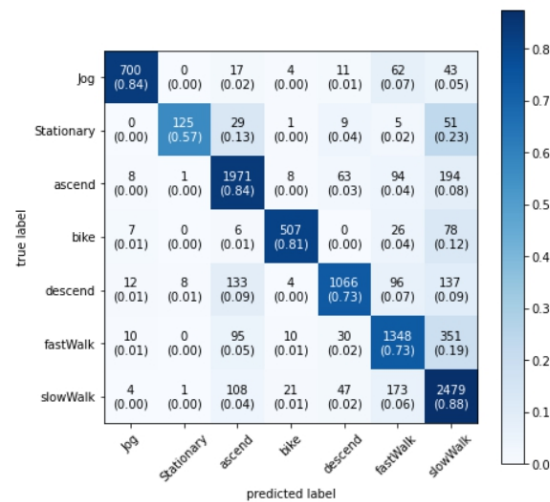


Figure 5.17 Confusion matrix for Extra Tree with out ReliefF (Stratified 10-fold)

In the Figure, the numbers without brackets in the cells represent the number of segments for each activities, the numbers within brackets represent proportions of data accurately classified and misclassified for each class. One can see that the Extra Tree Classifier provides better accuracy in general compared to the KNN Classifier. Among all activities, Stationary activity has the lowest classification score with a value of 0.57, and Slow walk has the highest accuracy score as 0.88. Stationary is most often misclassified as Slow walk (0.23), and while Slow walk is most often misclassified as Fast walk (0.06). The reason for the high rate of misclassification of Stationary to Slow walk could be related to the data quality itself that resulted by manual labelling, by which Stationary segments selected sometimes contain tiny speed and acceleration values. By the Extra Tree Classifier, other types of activities are most often to be classified as Slow walk, and Ascending is the activity that other activities misclassified second most as. This is slightly different from the KNN Classifier, by which Ascending is the activity that other types of activities are most often classified as. The classes the Extra Tree Classifier is good at recognizing besides Slow walk are Jogging, Ascending, and Biking.

Figure 5.18 shows the confusion matrices comparison for the Extra Tree Classifier with and without (accuracy value of 0.806) the ReliefF feature selection method by stratified 10 fold validation. The right side confusion matrix is from the combination of 60 features and 20 neighbors by ReliefF that exerts the best accuracy with the value of 8.25 stated from the Table 5.3 analysed above. From the figure, one can see that ReliefF slightly improves the classification results of the Extra Tree Classifier in some classes and keeps the same accuracy for others. More specifically, by this method, the accuracy results for the most accurately classified two classes namely Jogging and Slow walk remain the same. And the performances for Ascending, Descending, Biking, Stationary, and Fast walk all improve on a small scale from the value of 0.02 to 0.04. The result shows that ReliefF is useful in reducing the dimensionality of the features and increasing the performance of the Extra Tree Classifier.

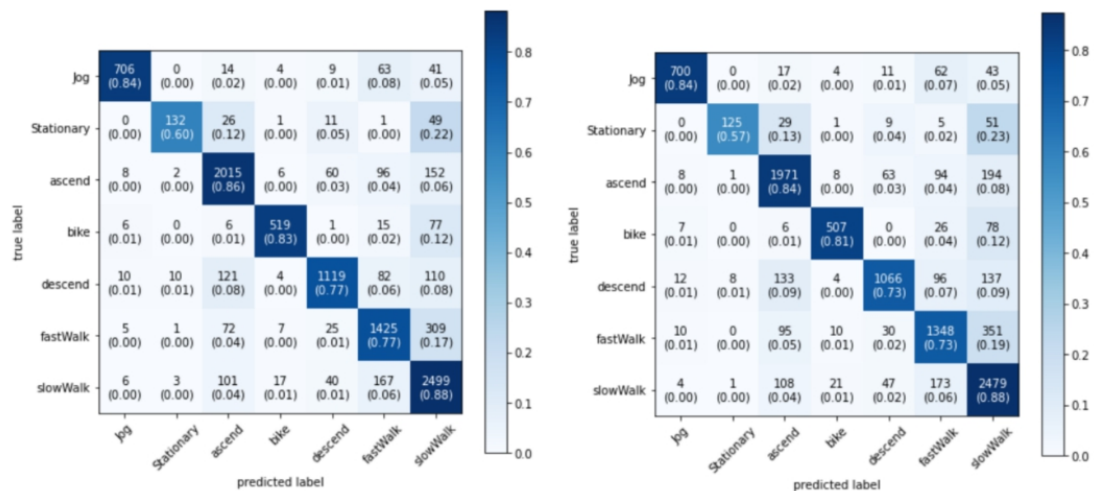


Figure 5.18 Comparison of confusion matrix for Extra Tree with (Left) and without (Right) ReliefF by stratified 10 fold

Figure 5.19 shows the confusion matrices comparison for the Extra Tree Classifier with and without (accuracy value 0.818) the ReliefF feature selection method by holdout validation. The right side confusion matrix is from the combination of 100 features and 20 neighbors by ReliefF that exerts the best accuracy with the value of 8.50 stated from the Table 5.4 analysed above. Similarly as the result from stratified 10-fold validation, one can see that by this validation, ReliefF slightly improves the classification results of the Extra Tree Classifier in Ascending, Descending, Biking, and Fast walk on a small scale from the value of 0.02 to 0.07 and keeps the same accuracy for others including Stationary, Slow walk and Jogging. In terms of the comparison for both validation methods, the differences of the accuracy scores for different classes are not huge, though it might result in different hyper-parameter selections when training the models as concluded from the last sub-chapter.

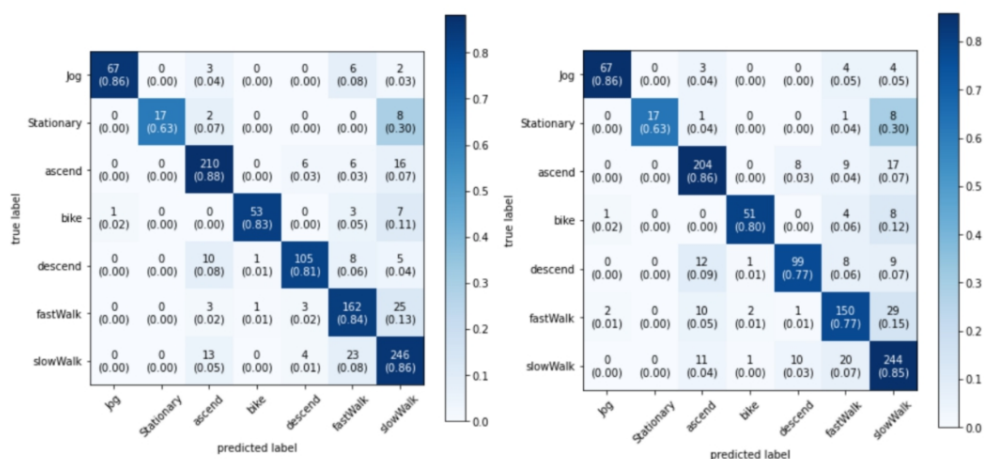


Figure 5.19 Comparison of confusion matrix for Extra Tree with (Left) and without (Right) ReliefF by holdout validation

Last, the feature importance by Extra Tree Classifier before and after the ReliefF method is also given. Figure 5.20 shows the feature importance of the top 20 most

important features for Extra Tree Classifier before and after ReliefF feature selection measured by Shapley values. Shapley values (Kalai et al. 1987) are a widely used approach from cooperative game theory that come with desirable properties.

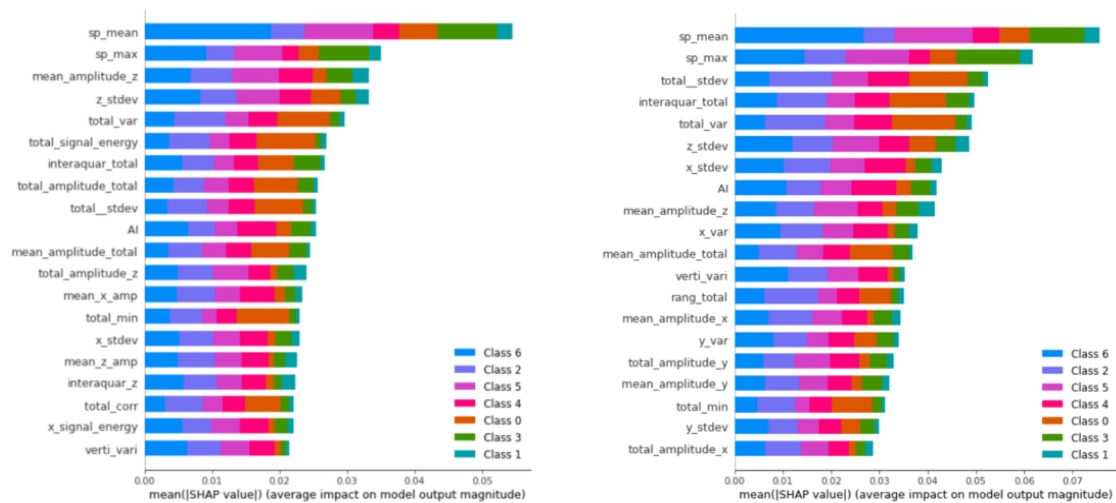


Figure 5.20 Feature importance comparison of Extra Tree Classifier before (Left) and after (Right) ReliefF feature selection

By the Extra Tree Classifier with the original 118 feature sets, the most important 20 features are: mean speed from the GPS sensor, maximum speed from the GPS sensor, mean amplitude of the z axis peaks, standard deviation of z axis signal, variance of the total acceleration, signal energy of the total acceleration, interquartile range of the total acceleration, sum of amplitudes of the total acceleration peaks, standard deviation of the total acceleration, variance of Movement Intensity, mean amplitude of the total acceleration peaks, sum of amplitudes of the z axis peaks, mean amplitude of top three dominant frequencies of x axis, minimum of total acceleration, standard deviation of the x axis, mean amplitude of top three dominant frequencies of z axis, interquartile range of z axis, total acceleration’s autocorrelation, signal energy of the x axis, variance of the altitudes from the GPS sensor.

Among the top 20 features trained by the original feature set, 3 of them are GPS features out of the 7 calculated, and 2 of them are the top 2 most important features, namely the mean and maximum speeds. The rest 17 features contain 4 distinctive features out of the 16 summarized from the literature, 6 time domain features, and 7 frequency domain features (see Table 5.9). The time domain features are statistical descriptions for the x, z and total accelerations. In frequency domain features, 3 of them are peak features in x and total acceleration signals, 1 of them is signal energy features in x axis, 2 of them are dominant frequencies in x and z axis. Overall speaking, for the Extra Tree Classifier, the important features in PA recognition are speed features, statistical features in time domain, dominant frequencies, signal energy, and peaks features in frequency domain.

Figure 5.20 also shows feature importance with regard to different classes. The class number 0 to 6 represents Jogging, Stationary, Ascending, Biking, Descending, Fast walk, and Slow walk respectively. The colours in the figure are arranged in the order of the number of segments of activities, namely slow walk, ascending, fast walk, descending, jogging, biking, and stationary. Looking at features individually, the

mean speed has a relatively large importance values than the rest features, and it has a great impact on the distinguishing slow walk, fast walk and biking.

For slow walk detection, other important features are maximum speed, standard deviation of the z axis, mean value of the movement intensity. For the detection of ascending, total acceleration variation, standard deviation of total acceleration, auto-correlation of total acceleration all have relatively higher impact. With regard to the detection of fast walk, maximum speed and amplitudes of the z axis peaks play an vital role. Mean value of the movement intensity, vertical variance are crucial features in the recognition of the descending class. Mean and maximum speeds are important for biking while total acceleration variation, total signal energy, minimum of total acceleration are key features of jogging detection. For stationary, the main features are also mean and maximum speeds. However, the importance values for features for the stationary activity are rather low. This could be the sign that the classifier is not good at detecting the stationary class, which is reflected by the previous confusion matrix.

Table 5.7 Important features for different classes by Extra Tree Classifier without feature selection

Activity types	Features
Slow walk	maximum speed, standard deviation of the z axis, mean value of the movement intensity
Ascending	total acceleration variation, standard deviation of total acceleration, auto-correlation of total acceleration
Fast walk	maximum speed and amplitudes of the z axis peaks
Descending	Mean value of the movement intensity, vertical variance
Jogging	total acceleration variation, total signal energy, minimum of total acceleration
Biking	Mean and maximum speeds
Stationary	mean and maximum speeds

After the ReliefF feature selection method, one can see from Figure 5.20 that the top most important features are: mean speed from the GPS sensor, maximum speed from the GPS sensor, standard deviation of the total acceleration, interquartile range of total acceleration, variance of the total acceleration, standard deviation of the z axis, standard deviation of the x axis, mean of Movement Intensity, mean amplitude of top three dominant frequencies of z axis, variance of the x axis signal, mean amplitude of top three dominant frequencies of total acceleration, variance of the altitudes from the GPS sensor, range of the total acceleration, mean amplitude of top three dominant frequencies of x axis, variance of y axis, total amplitude of top three dominant frequencies of y axis, mean amplitude of top three dominant frequencies of y axis, minimum of total acceleration, standard deviation of y axis, total amplitude of top three dominant frequencies of total acceleration.

Among the top 20 features trained by the ReliefF selected feature set, 3 of them are GPS features out of the 7 calculated, and 2 of them are the top 2 most important features, namely the mean and maximum speeds. The rest 17 features contain 4 distinctive features out of the 16 summarized from the literature, 7 time domain features, and 6 frequency domain features (see Table 5.8). The time domain features

are statistical descriptions for the x, z, y and total accelerations. In frequency domain features, 4 of them are peak features in x and total acceleration signals, 2 of them are dominant frequencies in total and y axis. Overall speaking, after applying the ReliefF feature selection method, the important features in PA recognition are speed features, statistical features in time domain, dominant frequencies, and peaks features in frequency domain. Looking at features individually, the mean speed still has a relatively large importance values than the rest features, and it has a great impact on the distinguishing slow walk, fast walk and biking as well.

In terms of the key features for different classes, for slow walk detection, important features are mean speed, maximum speed, and standard deviation of the z axis. For the detection of ascending, total acceleration variation, standard deviation of total acceleration, the range of total acceleration, and vertical altitude variance all have relatively higher impact. With regard to the detection of fast walk, maximum speed and standard deviation of the z axis play a vital role. Standard deviation of total acceleration, standard deviation of the x axis, and mean of movement intensity are crucial features in the recognition of the descending class. Mean and maximum speeds are important for biking while total acceleration variation, standard deviation of total acceleration, minimum value of total acceleration, and interquartile range of the total acceleration are key features of jogging detection. For stationary, the main features are also mean and maximum speeds, mean amplitudes of the z axis, standard deviation of the z axis. However, the importance values for features for the stationary activity are still rather low.

Table 5.8 important features for different classes by Extra Tree Classifier with ReliefF feature selection

Activity types	Features
Slow walk	Mean speed, maximum speed, standard deviation of the z axis
Ascending	total acceleration variation, standard deviation of total acceleration, the range of total acceleration, and vertical altitude variance
Fast walk	Mean and maximum speeds, and standard deviation of the z axis
Descending	total acceleration, standard deviation of the x axis, and mean of movement intensity
Jogging	total acceleration variation, standard deviation of total acceleration, minimum value of total acceleration, and interquartile range of the total acceleration
Biking	Mean and maximum speeds
Stationary	mean and maximum speeds, mean amplitudes of the z axis, standard deviation of the z axis

Compared to the features selected from the original data, the top 20 features selected by ReliefF still has the similar numbers of features in different feature categories. Also, in both selected feature sets, the top two features are mean speed and maximum speed from the GPS sensor. Though a part of features has changed, the feature sets are still from the same types of features, such as the amplitudes of dominant frequencies in different sensor signal dimensions (refers to as different axes of signals), the peaks amplitudes, the standard deviation and ranges in different sensor signal dimensions. Another main difference of the feature sets beside the modification of features is the change of feature importance ranking. For instance, in terms of the features from GPS

sensor, ReliefF increases the importance of variance of altitude by 8 positions. Moreover, the ReliefF increases the importance of the y axis features, which are not included in before the feature selection.

As aforementioned from the confusion matrices comparison, ReliefF improved the performance in ascending, descending, biking and jogging. By a comparison of the important features for different classes in table 5.7 and 5.8, it can be seen that the adding of vertical variance weight in ascending, total acceleration features' weights in jogging and descending by the ReliefF method is helpful in improving the model accuracy. Moreover, from Figure 5.20, the weight scores of all the top 20 features are also higher and more balanced after feature selection.

Table 5.9 Top 20 feature categories comparison before and after ReliefF

Feature types			Extra Tree Classifier without feature selection	Extra Tree Classifier with ReliefF feature selection
GPS features			mean speed, maximum speed, variance of the altitudes	mean speed, maximum speed, variance of the altitudes
Acceler ation features	Non- distinctiv e features	Time domain	standard deviation of the x axis, y axis, total acceleration and the z axis, interquartile range of the total acceleration, minimum of total acceleration, interquartile range of z axis	standard deviation of the x axis, y axis, total acceleration and the z axis, interquartile range of total acceleration, range of the total acceleration, minimum of total acceleration
		Freque ncy domain	mean amplitude of the z axis and total acceleration peaks, sum of amplitudes of the total acceleration peaks and z axis peaks, mean amplitude of top three dominant frequencies of x axis and z axis, signal energy of the x axis	mean amplitude of top three dominant frequencies of z axis, x axis, y axis and total acceleration, total amplitude of top three dominant frequencies of y axis and total acceleration.
	Disinctiv e features		Variance of Movement Intensity (VI), Averaged acceleration energy (AAE), variance of the total acceleration, Auto- correlation of total acceleration	variance of the total acceleration, variance of the x axis signal, variance of y axis, mean of Movement Intensity (AI)

5.4.2.2 Genetic Algorithm

This thesis uses the Genetic Feature Selection Module of python (See Appendix 8.2) (Calzolari 2021) to perform the GA selection of features. In the GeneticSelectionCV function, a number of hyper-parameters are available to be tuned for the optimization of the function. The crossover operation represents a reproduction process, for example, the recombination of populations of a generation in different ways, and is usually applied with a high probability. Mutation is defined as a small random tweak in chromosome to get a new solution, and is usually applied with a low probability. The crossover probability and mutation probability in this thesis are set as 35%, and 1.5% respectively. The parameters are taken from the studies from Baldominos et al.(2015). And from their study, the preliminary experimentation had shown no significant difference in the results with alternative set-ups.

For other parameters, the tournament size is set as 3, and the number of generations is set as 10. The tournament size represents the number of participants of individuals in each “tournament” that selects the winner of participants for the crossover and mutation operation. When the tournament size is larger, weaker individuals have a smaller chance to be selected (Miller 1995). For the maximum number of features selected by this algorithm, a grid search for the number of 16, 30, 60, 118 is implemented, and the results can be seen in Table 5.5. It is noteworthy that compared to ReliefF algorithm, the parameter number of features for the Genetic algorithm represents the maximum number of features instead of the fixed number of features. This means that for a given number, all numbers smaller than it are executed as well. For example, given the maximum number of features as 16, all combinations with 2 to 16 features are all run. This results in higher computational time compared to the ReliefF method. Stratified 10 fold validation is selected as the validation method, and the holdout validation method is no longer inspected as the GeneticSelectionCV function does not support the non-cross-validation method.

Table 5.10 shows the Genetic selection method results with respect to different hyperparameter *maximum number of features*, and their accuracy scores accordingly. The accuracy score in the bracket is the accuracy without GA feature selection. For a more detailed illustration of individual features selected, see Table 8.1 in Appendix 8.1. One can see that the hyperparameter *maximum number of features* that gives the best result is the 60. This setting selects 58 features, and gives a performance score of 0.824. It is noteworthy that though the *maximum number of features* parameter is set as the maximum number of features calculated as 118, the accuracy score is still less than the setting of figure 60. This suggests the importance to inspect different figures for this parameter for a better performance result.

Table 5.10 GeneticSelection with Extra Tree Classifier

maximum number of features	Number of Features	Accuracy Scores	Computational Time
16	16	0.807 (0.806)	1426.8s
30	30	0.816 (0.806)	1607.8s
60	58	0.824 (0.806)	2151.5s
118	70	0.817 (0.806)	2220.9s

Figure 5.21 illustrates the comparison of the confusion matrices by Extra Tree Classifier with the feature set of all features and the best set selected by the Genetic Algorithm. The GA selection method improves the classification accuracy in Stationary, Ascending, Biking, Descending, and Fast walk, while keeps the performance in Slow walk and Jogging. The class with the highest accuracy score predicted by GA is still Slow walk.

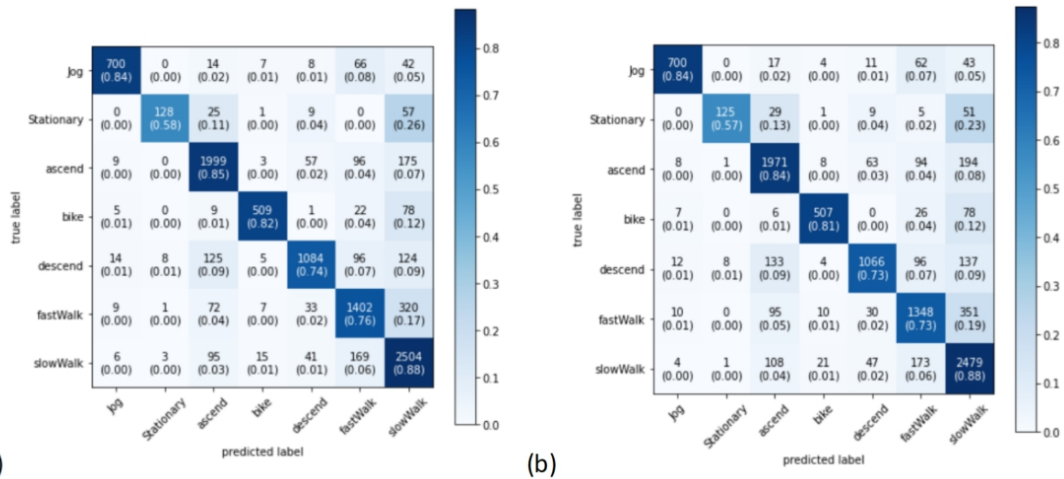


Figure 5.21 Confusion matrix by Extra Tree Classifier with the feature set selected by GA (a) and the whole feature set (b)

Figure 5.22 shows the feature importance comparison of the Extra Tree Classifier trained by the whole feature set and GA selected feature set. The top 20 most important features from the GA feature set are: mean speed from the GPS sensor, interquartile range of the total acceleration, total acceleration’s autocorrelation, variance of total acceleration, maximum speed from the GPS sensor, mean amplitude of top three dominant frequencies of x axis, mean amplitude of the z axis peaks, autocorrelation of the z axis, mean of Movement Intensity, standard deviation of the z axis, total amplitude of the z axis peaks, variance of the x axis, mean amplitude of the x axis peaks, vertical variance of altitude from the GPS sensor, standard deviation of the speed from GPS sensor, standard deviation of the x axis, mean amplitude of top three dominant frequencies of total acceleration, the second dominant frequency’s amplitude of the z axis, Eigenvalues of the horizontal directions, the dominant frequency’s amplitude of the x axis.

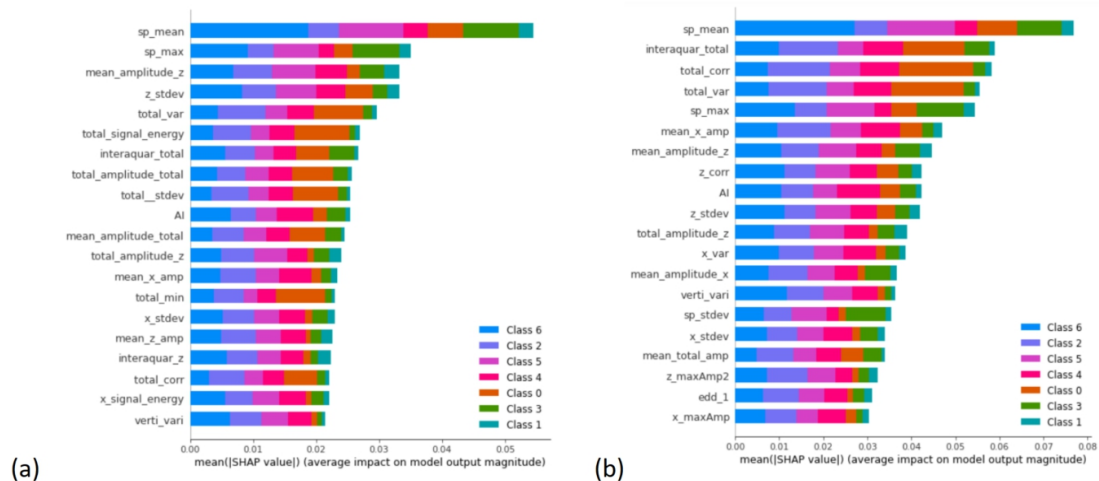


Figure 5.22 Feature importance comparison of the Extra Tree Classifier trained by the whole feature set (a) and GA selected feature set (b)

Table 5.11 shows the top 20 GA selected features in different in categories and the comparison of them between the top 20 features trained by the original data.

Among the top 20 features trained by the GA selected feature set, 4 of them are GPS features out of the 7 calculated. The rest 16 features contain 5 distinctive features out of the 16 summarized from the literature, 5 time domain features, and 6 frequency domain features. The time domain features are statistical descriptions for the x, z and total accelerations. In frequency domain features, 2 of them are peak features in x and z axes signals, 4 of them are dominant frequencies in x, total, z, and y axes signals. Overall speaking, after applying the GA feature selection method, the important features in PA recognition are still speed features, statistical features in time domain, dominant frequencies, and peaks features in frequency domain.

Table 5.11 Top 20 feature categories comparison before and after genetic algorithm

			Extra Tree Classifier without feature selection	Extra Tree Classifier with GA feature selection
GPS features			mean speed, maximum speed, variance of the altitudes	mean speed, maximum speed, variance of the altitudes, standard deviation of the speed
Acceleration features	Non-distinctive features	Time domain	standard deviation of the x axis, y axis, total acceleration and the z axis, interquartile range of the total acceleration, minimum of total acceleration, interquartile range of z axis	standard deviation of the x axis, interquartile range of total acceleration, range of the total acceleration, minimum of total acceleration, autocorrelation of the z axis
		Frequency domain	mean amplitude of the z axis peaks, sum of amplitudes of the total acceleration peaks, mean amplitude of the total acceleration peaks, sum of amplitudes of the z axis peaks, mean amplitude of top three dominant frequencies of x axis, mean amplitude of top three dominant frequencies of z axis, signal energy of the x axis,	mean amplitude of the z axis and x axis peaks, mean amplitude of top three dominant frequencies of x axis, total acceleration, the second dominant frequency's amplitude of the z axis, the dominant frequency's amplitude of the x axis
	Distinctive features	Variance of Movement Intensity (VI), Averaged acceleration energy (AAE), variance of the total acceleration, Auto-correlation of total acceleration	Auto-correlation of total acceleration, variance of the total acceleration and x axis, mean of Movement Intensity, Eigenvalues of the horizontal directions	

In terms of the key features for different classes, for slow walk detection, important features are still the mean speed, maximum speed, and standard deviation of the z axis. For the detection of ascending, autocorrelation of total acceleration, variance of total acceleration, the interquartile range of total acceleration all have relatively higher impact. With regard to the detection of fast walk, mean and maximum speeds play a vital role. Autocorrelation of total acceleration, variance of total acceleration, the interquartile range of total acceleration and mean of movement intensity are crucial features in the recognition of the descending class. Mean, standard deviation and maximum speeds are important for biking while autocorrelation of total acceleration, variance of total acceleration, the interquartile range of total acceleration are key

features of jogging detection. For stationary, the main features are also mean, mean amplitudes of the z axis. However, the importance values for features for the stationary activity are still rather low.

Table 5.12 important features for different classes by Extra Tree Classifier with GA feature selection

Activity types	Features
Slow walk	mean speed, maximum speed, standard deviation of the z axis
Ascending	autocorrelation of total acceleration, variance of total acceleration, the interquartile range of total acceleration
Fast walk	mean and maximum speeds, and standard deviation of the z axis
Descending	autocorrelation of total acceleration, variance of total acceleration, the interquartile range of total acceleration and mean of movement intensity
Jogging	autocorrelation of total acceleration, variance of total acceleration, the interquartile range of total acceleration
Biking	mean and maximum speeds
Stationary	mean, mean amplitudes of the z axis

Compared to the features selected from the original data, the top 20 features selected by GA method still has the similar numbers of features in different feature categories. Also, in both selected feature sets, the top feature is the mean speed from the GPS sensor. Though a part of features has changed, the feature sets are still from the same types of features, such as the amplitudes of dominant frequencies in different sensor signal dimensions, the peaks amplitudes, the standard deviation and ranges in different sensor signal dimensions. Similar to the ReliefF method's result, one of the main differences of the feature sets is the change of feature importance ranking. For instance, in terms of the features from GPS sensor, GA increases the importance of variance of altitude by 6 positions and decrease the importance of the maximum speed by 4 positions. Moreover, the GA increases the importance of the distinctive feature Eigenvalue of the horizontal direction, which is also a feature constructed by sensor fusion, and is not included in the top 20 before the feature selection.

As aforementioned from the confusion matrices comparison, GA improved the performance in Stationary, Ascending, Biking, Descending, and Fast walk. By a comparison of the important features for different classes in table 5.7 and 5.12, it can be seen that the adding of total acceleration and the z axis features' weights by the GA method is helpful in improving the model accuracy. Moreover, from Figure 5.22, the weight scores all the top 20 features are also higher and more concentrated after feature selection.

5.4.2.3 Recursive feature elimination

This thesis uses the Feature_Selection Module of python (See Appendix 8.2) (Calzolari 2021) to perform the recursive feature elimination selection of features. In the RFECV function, two major hyper-parameters are available to be tuned for the optimization of the function. The *step* parameter corresponds to the number of features to remove at each iteration. The *minimum number of features* represents the minimum number of features to be selected. When this parameter is set as 1, all

combinations of feature sets from 1 feature to the whole feature sets (in this case 118 features) will be executed. In this analysis, the *step* parameter is set as 1 and 3 to inspect if the step size would have an impact on the feature selection method output. The *minimum number of features* is first set as 1 to obtain an overview of the performance across all combinations and thus to get an understanding of this algorithm better. Then a grid search of values for 16 and 60 are conducted for *minimum number of features* with the *step* set as 1 to investigate if this feature would influence the performance. Stratified 10 fold validation is selected as the validation method, and the holdout validation method is no longer inspected as the RFECV function does not support the non-cross-validation method.

Table 5.13 shows the RFECV method results with respect to different hyperparameter *step* and the minimum number of features, and their accuracy scores accordingly. The accuracy scores in the brackets are the accuracy without RFECV feature selection. For a more detailed illustration of individual features selected, see Table 8.2 in Appendix 8.1. One can see that the hyperparameter *step* that gives the best result is the 1. And the combination of features that gave the highest score is 1 as step size with 60 as minimum features. This setting selects 71 features, and gives a performance score of 0.827. It can be seen from the table that the smaller the step size, the higher the minimum number of features, the longer the computational time cost.

Table 5.13 RFECV with Extra Tree Classifier

Step	Minimum number of features	Number of Features selected	Accuracy Scores	Computational Time
1	1	56	0.826 (0.806)	2322.0s
3	1	64	0.824 (0.806)	778.9s
1	16	69	0.826 (0.806)	2314.4s
1	60	71	0.827 (0.806)	1473.7s

Figure 5.23 shows the cross validation scores for the settings with minimum number of features equal to 1, and step size as 1. One can see that as number of feature grows, the scores increase as well. Until a certain number of features (around 10 in this case), the scores maintain at a good level. This reflects that for a minimum number of 10 features from the whole feature set can provide a good level of accuracy score by the Extra Tree Classifier. However, the improvement of the performance might not be huge in this case. This could be related to the data itself and the classifier chosen. Other Figures for different minimum feature size are also given in Appendix 8.1 Figure 8.5. The figures give a closer examination of the optimum number of features to select, as the curve is in the bell shape with a smaller range of numbers of features for feature sets.

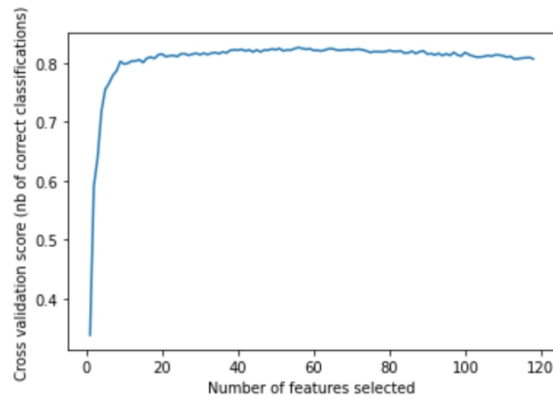


Figure 5.23 Cross validation scores with minimum number of features by RFECV with Extra Tree Classifier

Figure 5.24 illustrates the comparison of the confusion matrices by Extra Tree Classifier with the feature set of all features and the best set selected by the RFE method. The RFE selection method improves the classification accuracy in Stationary, Ascending, Biking, Descending, and Fast walk, and Slow walk while keeps the performance in Jogging. The class with the highest accuracy score predicted by RFE is still Slow walk.

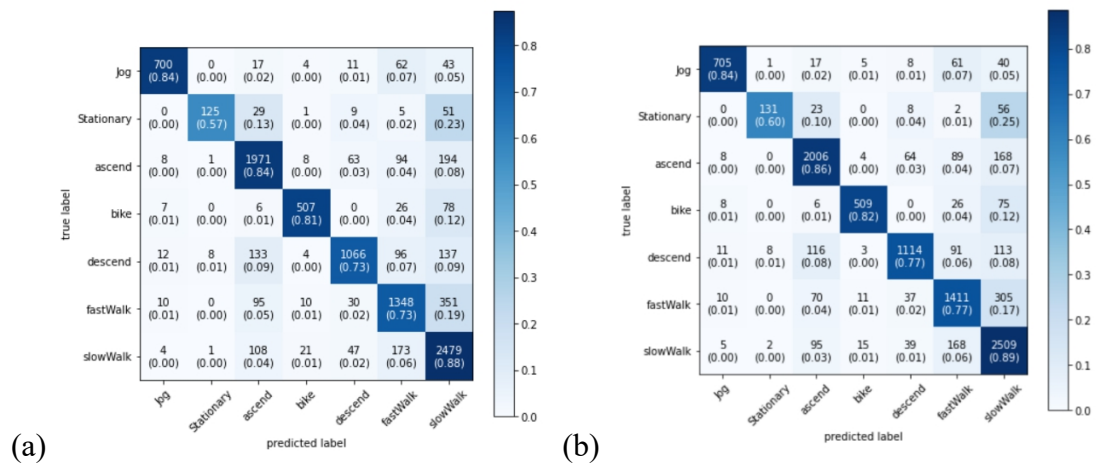


Figure 5.24 Confusion matrix by Extra Tree Classifier with the feature set selected by RFE (a) and the whole feature set (b).

Figure 5.25 shows the feature importance comparison of the Extra Tree Classifier trained by the whole feature set and RFE selected feature set. The top 20 most important features from the RFE feature set are: mean speed from the GPS sensor, maximum speed from the GPS sensor, total signal energy, total acceleration's autocorrelation, mean amplitude of the z axis peaks, vertical variance of altitude from the GPS sensor, total acceleration variance, z axis signal variance, mean amplitude of top three dominant frequencies of x axis, standard deviation of the total acceleration, interquartile range of the total acceleration, standard deviation of the speed, mean of Movement Intensity, variance of total acceleration, mean amplitude of the x axis peaks, standard deviation of the z axis signal, total amplitude of the total acceleration peaks, autocorrelation of the z axis, total amplitude of the z axis peaks, variance of the x axis, mean amplitude of the total acceleration peaks, the third dominant frequency's amplitude of the x axis.

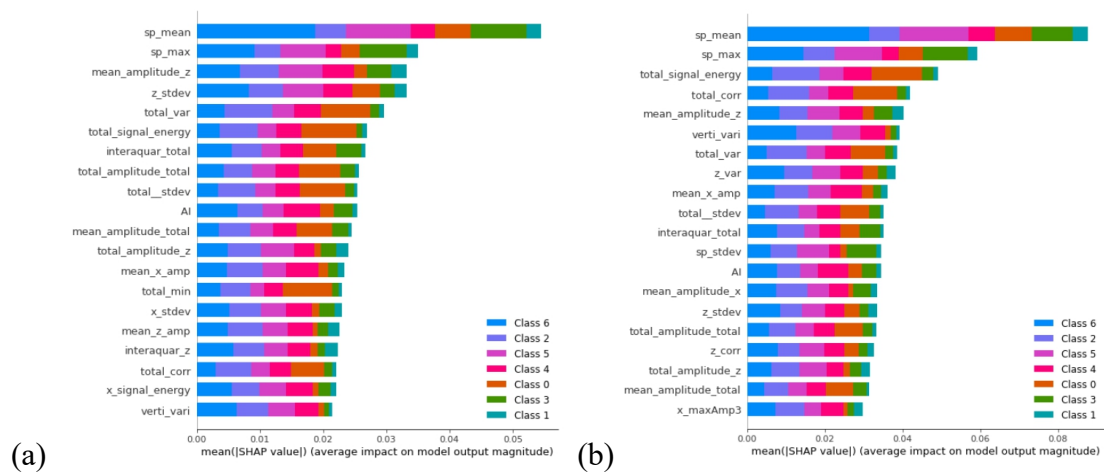


Figure 5.25 Feature importance comparison of the Extra Tree Classifier trained by the whole feature set (a) and RFE selected feature set (b)

Table 5.14 shows the top 20 RFE selected features in different in categories and the comparison of them between the top 20 features trained by the original data. Among the top 20 features trained by the RFE selected feature set, 4 of them are GPS features out of the 7 calculated. The rest 16 features contain 6 distinctive features out of the 16 summarized from the literature, 5 time domain features, and 5 frequency domain features. The time domain features are statistical descriptions for the z and total accelerations. In frequency domain features, 4 of them are peak features in x and z axes, and total acceleration signals, 1 of them is dominant frequencies in x axis signal. Overall speaking, after applying the RFE feature selection method, the important features in PA recognition are still speed features, statistical features in time domain, dominant frequencies, and peaks features in frequency domain.

Table 5.14 Top 20 feature categories comparison before and after RFE algorithm

			Extra Tree Classifier without feature selection	Extra Tree Classifier with RFE feature selection
GPS features			mean speed, maximum speed, variance of the altitudes	mean speed, maximum speed, variance of the altitudes, standard deviation of the speed
Acceler ation features	Non- distinctiv e features	Time domain	standard deviation of the x axis, y axis, total acceleration and the z axis, interquartile range of the total acceleration, minimum of total acceleration, interquartile range of z axis	standard deviation of the total acceleration and the z axis signal, interquartile range of total acceleration, range of the total acceleration, autocorrelation of the z axis
		Freque ncy domain	mean amplitude of the z axis peaks, sum of amplitudes of the total acceleration peaks, mean amplitude of the total acceleration peaks, sum of amplitudes of the z axis peaks, mean amplitude of top three dominant frequencies of x axis, mean amplitude of top three dominant frequencies of z axis,	mean amplitudes of the z axis, total acceleration and x axis peaks, total amplitude z axis peaks, mean amplitude of top three dominant frequencies of x axis

			signal energy of the x axis	
	Distinctive features		Variance of Movement Intensity (VI), Averaged acceleration energy (AAE), variance of the total acceleration, Auto-correlation of total acceleration	Auto-correlation of total acceleration, variance of the total acceleration, x axis and z axis, mean of Movement Intensity Averaged acceleration energy (AAE)

In terms of the key features for different classes, for slow walk detection, important features are still the mean speed, maximum speed, and vertical distance variation. For the detection of ascending, total signal energy, total signal variance both have relatively higher impact. With regard to the detection of fast walk, mean and maximum speeds, mean amplitude of the z signal's peaks play a vital role. Vertical distance variation and mean of movement intensity are crucial features in the recognition of the descending class. Mean, standard deviation and maximum speeds are important for biking while autocorrelation of total acceleration, variance of total acceleration, total signal energy are key features of jogging detection. For stationary, the main features are still mean and maximum speed. Also the importance values for features for the stationary activity are still rather low.

Table 5.15 important features for different classes by Extra Tree Classifier with RFE feature selection

Activity types	Features
Slow walk	mean speed, maximum speed, and vertical distance variation
Ascending	total signal energy, total signal variance
Fast walk	mean and maximum speeds, mean amplitude of the z signal's peaks
Descending	Vertical distance variation and mean of movement intensity
Jogging	autocorrelation of total acceleration, variance of total acceleration, total signal energy
Biking	mean and maximum speeds
Stationary	mean, mean amplitudes of the z axis

Compared to the features selected from the original data (see Table 5.7), the top 20 features selected by the RFE method still have the similar numbers of features in different feature categories. Also, in both selected feature sets, the top two features are still the mean and maximum speed from the GPS sensor. Though a part of features has changed, the feature sets are still from the same types of features. Similar to the results from ReliefF and GA, one of the main differences of the feature sets is the change of feature importance ranking. For instance, in terms of the features from GPS sensor, RFE increases the importance of variance of altitude by 6 positions and decrease the importance of the maximum speed by 14 positions. Also, the GA increases the importance of the features from the distinctive feature category.

As aforementioned from the confusion matrices comparison, The RFE selection method improves the classification accuracy in Stationary, Ascending, Biking, Descending, and Fast walk, and Slow walk while keeps the performance in Jogging.

By a comparison of the important features for different classes in table 5.7 and 5.15, it can be seen that the adding of speed features and distinctive features by the RFE method is helpful in improving the model accuracy.

5.4.2.4 Feature selection results comparison

The above sub-chapters describe the implementation of three feature selection methods with the Extra Tree Classifier and the results. This sub-chapter gives a concise summary of performance of the methods, and the corresponding selected feature sets (see Table 5.16). It can be seen that all three feature selection methods are helpful in increasing accuracy score with smaller number of feature sets. Embedded method gives the best performance while ReliefF has the lowest time cost. This corresponds to the conclusions from literatures. However, the GA method's performance is lower than expected. Overall speaking, the improvement are not huge, this could due to the Extra Classifier already gives a good performance result and that the data quality itself influenced by the manual labelling process. Nevertheless, the improvement of performances are still significant considering multiple runs of the algorithm also with other random state settings when inspecting. For the future, the feature selection methods combined with other classifiers can be explored.

Table 5.16 Feature selection methods and results comparison (Stratified-10 fold validation)

Feature selection methods	Number of features selected	Performance	Time costs
ReliefF (filter)	60	0.825	35.14s
GA (wrapper)	58	0.824 (0.806)	2151.5s
RFE (embedded)	71	0.827 (0.806)	1473.7s

5.5 Comparison of GPS and Accelerometer sensors

As analysed above, both GPS and Accelerometer dimensions of data play an vital role in distinguishing features. More specially, the three speed related features from the GPS sensor are always the top features in importance ranking, especially the mean and maximum speeds. Also, the vertical variance of altitudes feature from GPS is always ranked among the top 20 important features by all feature selection methods.

The Accelerometer sensors consist larger number of features, and provides intensity characteristics that GPS does not provide. In this analysis, the Extra Tree Classifier training with individual GPS and Accelerometer data are given for a further inspection under the feature importance theme. Figure 5.26 shows the classification results from individual sensors data by different classifiers. Overall speaking, the performances trained by individual sensors data are lower than the data combination of both the sensors. However, individual data can still give good performance. Especially for GPS sensor, with only 7 features, it still gives an accuracy score of around 0.7.

Results

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)		Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.79	0.96	0.65	0.80	0.79	0.74	0.74	3.50	lightgbm	Light Gradient Boosting Machine	0.71	0.92	0.53	0.70	0.70	0.63	0.64	2.01
lightgbm	Light Gradient Boosting Machine	0.78	0.96	0.62	0.78	0.77	0.72	0.72	10.48	xgboost	Extreme Gradient Boosting	0.70	0.91	0.54	0.69	0.69	0.63	0.63	11.81
rf	Random Forest Classifier	0.77	0.95	0.61	0.78	0.76	0.71	0.71	3.84	catboost	CatBoost Classifier	0.70	0.92	0.54	0.69	0.69	0.63	0.63	28.82
knn	K Neighbors Classifier	0.62	0.85	0.51	0.62	0.61	0.53	0.53	1.12	rf	Random Forest Classifier	0.69	0.92	0.52	0.68	0.68	0.61	0.61	1.15
dt	Decision Tree Classifier	0.62	0.76	0.53	0.62	0.62	0.53	0.53	1.46	et	Extra Trees Classifier	0.67	0.91	0.49	0.66	0.66	0.59	0.59	1.56
qda	Quadratic Discriminant Analysis	0.59	0.86	0.45	0.61	0.59	0.50	0.51	0.13	gbc	Gradient Boosting Classifier	0.66	0.90	0.51	0.66	0.66	0.58	0.58	6.41
lda	Linear Discriminant Analysis	0.55	0.84	0.49	0.57	0.55	0.44	0.45	0.34	dt	Decision Tree Classifier	0.57	0.74	0.47	0.58	0.57	0.47	0.47	0.04
ada	Ada Boost Classifier	0.43	0.63	0.35	0.43	0.42	0.30	0.30	5.86	lda	Linear Discriminant Analysis	0.51	0.77	0.38	0.50	0.49	0.38	0.39	0.02
ridge	Ridge Classifier	0.42	0.00	0.28	0.44	0.41	0.27	0.28	0.09	ridge	Ridge Classifier	0.47	0.00	0.33	0.45	0.43	0.32	0.34	0.02
nb	Naive Bayes	0.34	0.73	0.38	0.46	0.34	0.25	0.27	0.06	ada	Ada Boost Classifier	0.40	0.62	0.34	0.44	0.38	0.28	0.30	0.31
svm	SVM - Linear Kernel	0.29	0.00	0.22	0.31	0.22	0.14	0.19	1.45	nb	Naive Bayes	0.40	0.72	0.26	0.33	0.33	0.23	0.26	0.02
lr	Logistic Regression	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.11	knn	K Neighbors Classifier	0.33	0.61	0.25	0.32	0.32	0.17	0.17	0.37
										lr	Logistic Regression	0.15	0.61	0.14	0.05	0.06	0.03	0.06	0.14
										qda	Quadratic Discriminant Analysis	0.11	0.58	0.21	0.24	0.11	0.05	0.06	0.03
										svm	SVM - Linear Kernel	0.12	0.00	0.11	0.02	0.04	0.00	0.00	0.14

Figure 5.26 classification results from GPS sensor (right) and Accelerometer (left) data by different classifiers.

6 Discussion

This Chapter discusses the analysis results under the framework of the proposed research questions. The first research question about the literature review of some popular papers in this theme is answered in Chapter 2. Therefore, this chapter mainly summarizes the conclusions to the last two research questions in each three subchapters.

6.1 RQ2 - Differences in performances by classifiers and dimension reduction methods

This research question is first answered by presenting individual performances of classifiers and dimension reduction methods. Later, a summary is given on the comparison of the proposed combinations.

6.1.1 Dimension reduction methods with K-means and KNN

6.1.1.1 K-mean with PCA

For the unsupervised K-means classifier, the performance is very low with an averaged accuracy around 0.115 of different parameter settings. This corresponds to the research conclusion from Peterek et al. (2014). The parameter for the number of times the k-means algorithm that is run with different centroid seeds is set as 3, 10, 20. The result does not show a significant performance difference by changing this parameter. Nevertheless, PCA helps in improving the classification accuracy, but the result is still poor. Therefore, K-means unsupervised learning is not considered as a proper classifier for PA classification in this thesis.

6.1.1.2 KNN with PCA

KNN method is not suitable for the original feature space with a really low accuracy score below 0.36. However, the application of PCA significantly improves the classification accuracy of KNN to an acceptable level (the highest score of 0.745 within the tested settings by the stratified 10-fold validation method). It is noteworthy that for the KNN classifier, observations are likely to be labelled as classes with larger amount of observations. In this case, most observations are detected as the slow walk, fast walk, and ascending. With the help of PCA, this situation is largely improved but still exists. Particularly there exist large proportions of misclassification of stationary, fast walk, biking to slow walk. This could be interpreted as the impact of the data quality itself.

In terms of the best parameters for the classifier and PCA method, the validation results from stratified 10-fold show that the best value for the *dimension* parameter for PCA in this combination might locates between 60 and 100 dimensions, as the both parameters result in similar performance score. For the same reason, the best value for the parameter *number of neighbors* for KNN might locate between 3 and 10. The 10-fold validation suggests that 100 is the best value for the *dimension* parameter, and the

best value for the parameter *number of neighbors* for KNN locate between 3 and 10 as well. This conclusion is different from the results from some researches, where smaller numbers of dimensions perform better accuracy score (Dehzangi et al., 2018). On the other hand, the different validation methods exert similar performance results but leads to different parameter selections. Additionally, the higher *the number of dimensions*, the higher the time cost, while *the number of neighbors* does not have a significant impact on the time costs in this analysis.

6.1.2 Feature selection methods with Extra Tree Classifier

Extra Tree Classifier is the best classifier considering both classification accuracy and time costs (see results from Chapter 5.3). It can provide an average accuracy around 0.8 with the default parameter settings for the classifier function. By this classifier, Slow walk has a highest accuracy score of 0.88 (stratified 10-fold), and Stationary activity has a lowest accuracy score of 0.57. And Stationary is most often misclassified as Slow walk (0.23). Extra Tree Classifier is also good at recognizing Jogging, Ascending, and Biking.

Three feature selection methods, namely ReliefF as the filter method, Genetic Algorithm as the wrapper method, and Recursive feature Elimination as the embedded method are implemented and compared in terms of their contributions in Extra Tree Classifier's performance improvement.

6.1.2.1 ReliefF

ReliefF is a common filter feature selection method which is applied and found useful in some PA studies (Dehzangi et al., 2018). In this study, the ReliefF method helps to slightly improve the performance of the Extra Tree Classifier to an accuracy score of 0.825 (stratified 10-fold). Regarding activity classes, ReliefF method slightly improves the classification results of the Extra Tree Classifier in Ascending, Descending, Biking, Stationary, and Fast walk and keeps the same accuracy for Jogging and Slow walk. Additionally, the time costs are low for this method (around 30 seconds).

In terms of the best parameters for the ReliefF method, the validation results show that the best value for the *dimension* parameter is 60 by stratified 10-fold validation and 100 by holdout validation. And the *number of neighbors* does not show a significant difference in performances. In terms of the comparison of validation methods, the differences in both accuracy score (highest for stratified 10-fold validation as 0.825 and for holdout validation for 0.85) and parameter selection exist as concluded in the last subchapter.

6.1.2.2 Genetic Algorithm

Genetic Algorithm is a common wrapper feature selection method (Baldominos et al. (2017). In this study, the Genetic Algorithm helps to slightly improve the performance of the Extra Tree Classifier to an accuracy score of 0.824 (stratified 10-fold). Regarding activity classes, Genetic Algorithm slightly improves the classification results of the Extra Tree Classifier in Stationary, Ascending, Biking, Descending, and Fast walk and keeps the same accuracy for Jogging and Slow walk. The class with the highest accuracy score predicted by GA is still Slow walk. Additionally, the time

costs are very high (above 2,000 seconds) for this method, this is due to this method runs all feature subsets under the maximum number of features defined.

In terms of the best parameter for the Genetic Algorithm, the validation results show that the best value for the *maximum number of features* parameter is 60, which in the end selects 58 features. In this feature selection method, only stratified 10 fold is applied, as the GeneticSelectionCV function does not support the non-cross-validation method.

6.1.2.3 Recursive feature Elimination

Recursive feature Elimination combined with SVM is a common embedded feature selection method. However, in the implementation process, this combination takes too long running to which does not considered as an effective method for this dataset. Therefore, in this analysis, RFE is combined with Extra Tree to inspect its influence on this classifier. In this study, the RFE helps to slightly improve the performance of the Extra Tree Classifier to an accuracy score of 0.827 (stratified 10-fold). Regarding activity classes, RFE slightly improves the classification results of the Extra Tree Classifier in Stationary, Ascending, Biking, Descending, and Fast walk, and Slow walk and keeps the same accuracy for Jogging. The class with the highest accuracy score predicted by RFE is still Slow walk. Additionally, the time costs are relatively high (above 1,000 seconds) for this method, this is due to this method runs all feature subsets above the minimum number of features defined. And for this method, the smaller the step size, the higher the minimum number of features, the longer the computational time cost.

In terms of the best parameter for the RFE, the validation results show that the best value for the *minimum number of features* parameter is 60, and the best value for the *step* parameter is 1, which in the end selects 71 features. One can see that as number of feature grows, the scores increase as well. Also from the plot of the performance curve with all feature subsets (See Figure 5.23), it can be seen that until a certain number of features (around 10 in this analysis), the scores maintain at a good level. This reflects that for a minimum number of 10 features from the whole feature set can provide a good level of accuracy score by the RFE - Extra Tree method. In this feature selection method, only stratified 10 fold is applied, as the RFECV function does not support the non-cross-validation method.

6.1.3 Comparison of feature selection methods

Looking at the feature selection methods individually, the filter method has the lowest computational costs, while the embedded method provides the best results with a lower computational time than the wrapper method. General speaking, all feature selection method helps in improving the performance of the Extra Tree Classifier to some extent. However, the improvement of the performance might not be huge in this case. This could relate to the data influenced by the manual labelling process and the classifier chosen. For example, the classification accuracy for the Stationary class is always much lower (round 0.6) by this Classifier in comparison to other classes though the feature selection methods are helpful in improving it slightly. In this case, a manual examination of misclassified segments can be done in the next step to find out the reason for the low performance in this class. Nevertheless, the improvement of performances are still significant considering multiple runs of the feature selection

algorithm with other random state settings when inspecting. In the future, the feature selection methods combined with other classifiers can be explored.

6.1.4 Comparison of validation methods

This analysis tried an extra holdout validation methods besides the stratified 10-fold methods on the combination of KNN Classifier with PCA, and ReliefF with Extra Tree Classifier to answer the research question about the different validation methods' impact (the last two feature selection methods are not applied the holdout validation due to the functions in Python do not support non-cross-validation methods). Theoretically speaking, the differences of the two validation methods are from the number of folds tested and the stratified method applied. For the holdout validation method, only one fold is fitted for testing and training, while for stratified k-fold method, k folds are fitted for testing and training in total to reduce the bias exist. The application of the stratified k-fold method helps in dealing with the unbalanced classes problems which is suitable for the case of this thesis as well. Since the stratified 10-fold is more reliable in theory compared to the holdout validation method, this analysis aims for testing if the holdout validation method can provide a similar level of result as stratified 10-fold, and thus be an alternative for model training. As the stratified k-fold spends k-1 times more time than the holdout validation method, holdout validation method can largely reduce the time for some cases with large dataset. Moreover, the comparison of the score values from the holdout method and k-fold method also indicates model stability. This is because that if the model performances among different folds are very different, the model is not considered stable.

The results show that there indeed exist certain levels of performance differences between the two validation methods but the accuracy score differences of those two validation methods are not significant. And as expected, the time costs for holdout method is much lower compared to stratified-k-fold method. This indicates that the holdout validation method can help in balancing the classification performance improvement and computational cost in feature transformation process for large datasets. However, different validation methods might result in the wrong judgement of the hyper-parameters to tune and thus influence the model training process.

6.2 RQ-3 Features from the two sensors and distinctive features

6.2.1 Feature comparison from two sensors

In this analysis, a total of 118 features are calculated, containing 7 GPS features and 111 accelerometer features. And the accelerometer features are further categorized as distinctive features summarized from some existing literature, other statistical time-domain features, and other statistical frequency domain features. After training the Extra Tree Classifier with only GPS features, only Accelerometer features, and both sensor features, one can see that the training with features from both sensors gives the best performance results, followed by only Accelerometer features. Nevertheless, it is noteworthy that though GPS features still show a good level of accuracy score (round

0.7) with a rather small number of features (7 compared to 111 of accelerometer features).

The distinctive features are ranked by the Extra Tree Classifier measured in Shapley values. Among the original 118 feature sets, the top 10 important features are: mean speed from the GPS sensor, maximum speed from the GPS sensor, mean amplitude of the z axis peaks, standard deviation of z axis signal, variance of the total acceleration, signal energy of the total acceleration, interquartile range of the total acceleration, sum of amplitudes of the total acceleration peaks, standard deviation of the total acceleration, variance of Movement Intensity. It can be seen that the top two important features are from GPS, and the rest 8 are from Accelerometer. Looking at features individually, the mean speed has a relatively large importance values than the rest features.

When looking at the top 20 features, 3 of them are GPS features out of the 7 calculated. The rest 17 features contain 4 distinctive features out of the 16 summarized from the literature, 6 time domain features, and 7 frequency domain features. After the application of the three feature selection methods, the top 20 features changes in ranking and individual features but still have the similar numbers of features in different feature categories (see Chapter 5 for more detail). Additionally, the weight scores of all the top 20 features are also higher and more balanced after feature selection, which is also due to the smaller number of total features trained.

6.2.2 Features for different activity types

This part examines distinctive features for different activity types. For Extra Tree Classifier without feature selection methods applied, the top important feature, mean speed, has a great impact on the distinguishing slow walk, fast walk and biking. For slow walk detection, other important features are maximum speed, standard deviation of the z axis, mean value of the movement intensity. For the detection of ascending, total acceleration variation, standard deviation of total acceleration, auto-correlation of total acceleration all have relatively higher impact. With regard to the detection of fast walk, maximum speed and amplitudes of the z axis peaks play an vital role. Mean value of the movement intensity, vertical variance are crucial features in the recognition of the descending class. Mean and maximum speeds are important for biking while total acceleration variation, total signal energy, minimum of total acceleration are key features of jogging detection. For stationary, the main features are also mean and maximum speeds. However, the importance values for features for the stationary activity are rather low. This could be a reflection that the classifier is not good at detecting the stationary class.

With the implementation of the ReliefF method, the adding of vertical variance weight in ascending, total acceleration features' weights in jogging and descending help in improving the model accuracy. Besides, ReliefF also increases the importance of variance of altitude by 8 positions as well as the importance of the y axis features, which are not included in before the feature selection.

With the implementation of the GA method, the adding of total acceleration and the z axis features' weights help in improving the model accuracy. Besides, GA also increases the importance of variance of altitude by 6 positions as well as the

distinctive feature Eigenvalue of the horizontal direction, which is also a feature constructed by sensor fusion, and is not included in the top 20 features before the feature selection.

.
With the implementation of the RFE method, the adding of speed features and distinctive features help in improving the model accuracy. Besides, GA also increases the importance of variance of altitude by 6 positions as well as the the importance of the features from the distinctive feature category.

7 Outlook

For the first and second research questions, a closer examination of the misclassified segments can be conducted to find out the reasons for the misclassification. Also, the data for this thesis is unbalanced, which could also be the reason for the highly unbalanced classification scores for different classes. The way to treat the unbalanced data in this analysis is to apply the stratified validation method. In the future, more efforts can be spare in dealing with unbalanced dataset.

In terms of the classifiers and validation methods, efforts on other classifiers that are not tested in this study can be further explored. Additionally, the tested methods in this thesis can still be inspected by adjusting the hyperparameters to find the optimum models. More specifically, in this thesis, for all dimension reduction methods, more fine grid searches can be conducted to find a better balance in both the performance score and the time cost. For the model evaluation, more metrics, such as model stability can be taken into consideration.

For the validation method comparison, more researches can be done in the future as well. For example, between the comparison of the k-fold and holdout method, this thesis only inspects the 10-fold and holdout method with a train/test split as 90%/10%, given the reason to investigate the influence of the bias range and stratified method. The next step could be comparing 10-fold and holdout method with a train/test split as 70%/30%, since these two methods are most common ones applied in literature, and most studies only selected one method to test the model. Besides the two methods, other validation methods can also be compared. For example, the comparison of leave-one-subject-out validation and k-fold validation is meaningful for this thesis. This comparison can validate if the model trained can be generalizable on other subjects.

In terms of the third research question about distinctive features from different sensor dimensions, there are still different aspects can be explored in the future. First, more GPS features can be extracted, such as the distance ratio in Wu et al.'s (2010) study. Second, as mentioned above in Chapter 3, features from the GPS such as the number of satellite can be helpful in detecting indoor and outdoor walking. For future research, these features can be explored for additional walking types. Third, other sensor dimensions, such as gyroscope and magnetometer, are also meaningful in activity detection. Especially for gyroscope, it is useful to detect directional walking, such as walking right and walking left. However, this is not suitable for large unlabelled dataset given the difficulty in labelling those activity types.

For this dataset, visualizations of different participants' PA types classified can be conducted to get an more intuitive overview of the older adults' daily exercising life. Different activity types' relations with geographic locations can be explored.

Literature

Allahbakhshi, H., Hinrichs, T., Huang, H., & Weibel, R. (2019). The key factors in physical activity type detection using real-life data: A systematic review. *Frontiers in Physiology*, 10, 75.

Ainsworth, B. E., Haskell, W. L., Whitt, M. C., Irwin, M. L., Swartz, A. M., Strath, S. J., ... & Jacobs, D. R. (2000). Compendium of physical activities: an update of activity codes and MET intensities. *Medicine & science in sports & exercise*, 32(9), S498-S516.

Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.L.: Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine. In: Bravo, J., Hervás, R., Rodríguez, M. (eds.) *IWAAL 2012*. LNCS, vol. 7657, pp. 216–223. Springer, Heidelberg (2012)

Baldominos, A., Saez, Y., & Isasi, P. (2015, July). Feature set optimization for physical activity recognition using genetic algorithms. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation* (pp. 1311-1318).

Baldominos, A., Isasi, P., Saez, Y., & Manderick, B. (2015, September). Monte carlo schemata searching for physical activity recognition. In *2015 International Conference on Intelligent Networking and Collaborative Systems* (pp. 176-183). IEEE.

Baldominos, A., Isasi, P., & Saez, Y. (2017, June). Feature selection for physical activity recognition using genetic algorithms. In *2017 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2185-2192). IEEE.

Bhati, B. S., & Rai, C. S. (2020). Ensemble Based Approach for Intrusion Detection Using Extra Tree Classifier. In *Intelligent Computing in Engineering* (pp. 213-220). Springer, Singapore.

Bayat, A., Pomplun, M., & Tran, D. A. (2014). A study on human activity recognition using accelerometer data from smartphones. *Procedia Computer Science*, 34, 450-457.

Boswell, D. (2002). Introduction to support vector machines. Department of Computer Science and Engineering University of California San Diego.

Bui, V., Le, N. T., Vu, T. L., Nguyen, V. H., & Jang, Y. M. (2020). GPS-Based Indoor/Outdoor Detection Scheme Using Machine Learning Techniques. *Applied Sciences*, 10(2), 500.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.

Chapelle, O., & Keerthi, S. S. (2008, August). Multi-class feature selection with support vector machines. In *Proceedings of the American statistical association* (Vol. 58).

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1), 16-28.

Cilla, R., Patricio, M. A., García, J., Berlanga, A., & Molina, J. M. (2009). Recognizing human activities from sensors using hidden markov models constructed by feature selection techniques. *Algorithms*, 2(1), 282-300.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273-297.

- Deng, Z., & Ji, M. (2010). Deriving rules for trip purpose identification from GPS travel survey data and land use data: A machine learning approach. In *Traffic and Transportation Studies 2010* (pp. 768-777).
- Dehzangi, O., & Sahu, V. (2018, August). IMU-Based Robust Human Activity Recognition using Feature Analysis, Extraction, and Reduction. In *2018 24th International Conference on Pattern Recognition (ICPR)* (pp. 1402-1407). IEEE.
- El Moudden, I., Ouzir, M., Benyacoub, B., & ElBernoussi, S. (2016). Mining human activity using dimensionality reduction and pattern recognition. *Contemporary Engineering Sciences*, 9(21), 1031-1041.
- Fujiki, Yuichi. (2010). iPhone as a physical activity measurement platform, CHI'10 Extended Abstracts on Human Factors in Computing Systems, 4315.
- Gong, L., Morikawa, T., Yamamoto, T., & Sato, H. (2014). Deriving personal trip data from GPS data: A literature review on the existing methodologies. *Procedia-Social and Behavioral Sciences*, 138, 557-565.
- Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1), 389-422.
- Gonzalez, P., Weinstein, J., Barbeau, S., Labrador, M., Winters, P., Georggi, N. L., & Perez, R. (2008, November). Automating mode detection using neural networks and assisted GPS data collected using GPS-enabled mobile phones. In *15th World congress on intelligent transportation systems* (pp. 16-20).
- Haskell, W. L., Lee, I. M., Pate, R. R., Powell, K. E., Blair, S. N., Franklin, B. A., ... & Bauman, A. (2007). Physical activity and public health: updated recommendation for adults from the American College of Sports Medicine and the American Heart Association. *Medicine & science in sports & exercise*, 39(8), 1423-1434.
- Huynh, T., & Schiele, B. (2005, October). Analyzing features for activity recognition. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies* (pp. 159-163).
- Ignatov, A. D., & Strijov, V. V. (2016). Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer. *Multimedia tools and applications*, 75(12), 7257-7270.
- I. Fatima, M. Fahim, Y.-K. Lee, and S. Lee, "A genetic algorithm-based classifier ensemble optimization for activity recognition in smart homes," *Appl Soft Comput*, vol. In Press, 2015.
- Jiang, W., & Yin, Z. (2015, October). Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proceedings of the 23rd ACM international conference on Multimedia* (pp. 1307-1310).
- Jones, P., Mirkes, E. M., Yates, T., Edwardson, C. L., Catt, M., Davies, M. J., ... & Rowlands, A. V. (2019). Towards a Portable Model to Discriminate Activity Clusters from Accelerometer Data. *Sensors*, 19(20), 4504.
- Kalai, E., & Samet, D. (1987). On weighted Shapley values. *International journal of game theory*, 16(3), 205-222.
- Kozina, S., Lustrek, M., & Gams, M. (2011, July). Dynamic signal segmentation for activity recognition. In *Proceedings of international joint conference on artificial intelligence* (Vol. 1622, p. 1522).
- Khan, A. M., Lee, Y. K., Lee, S., & Kim, T. S. (2010). Accelerometer's position independent physical activity recognition system for long-term activity monitoring in the elderly. *Medical & biological engineering & computing*, 48(12), 1271-1279.

Lara, O.D.; Labrador, M.A. A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Commun. Surv. Tutor.* 2013, 15, 1192–1209.

L. Bao and S. Intille. Activity Recognition from User-Annotated Acceleration Data. In *Pervasive Computing*, pages 1–17, Linz/Vienna, Austria, 2004.

Lu, Y., Zhu, S., & Zhang, L. (2012, May). A machine learning approach to trip purpose imputation in GPS-based travel surveys. In *4th Conference on Innovations in Travel Modeling, Tampa, Fla.*

Lu, Y., Wei, Y., Liu, L., Zhong, J., Sun, L., & Liu, Y. (2017). Towards unsupervised physical activity recognition using smartphone accelerometers. *Multimedia Tools and Applications*, 76(8), 10701-10719.

Kranz, M.; Möller, A.; Hammerla, N.; Diewald, S.; Plötz, T.; Olivier, P.; Roalter, L. The mobile fitness coach: Towards individualized skill assessment using personalized mobile devices. *Perv. Mob. Comput.* 2013, 9, 203–215.

Ordóñez, F. J., & Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1), 115.

Lal, T. N., Chapelle, O., Weston, J., & Elisseeff, A. (2006). Embedded methods. In *Feature extraction* (pp. 137-165). Springer, Berlin, Heidelberg.

Liu, H., & Motoda, H. (1998). Feature transformation and subset selection. *IEEE Intell Syst Their Appl*, 13(2), 26-28.

Likas, A., Vlassis, N., & Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern recognition*, 36(2), 451-461.

Patel, H. H., & Prajapati, P. (2018). Study and analysis of decision tree based classification algorithms. *International Journal of Computer Sciences and Engineering*, 6(10), 74-78.

Park, J., Ishikawa-Takata, K., Tanaka, S., Mekata, Y., & Tabata, I. (2011). Effects of walking speed and step frequency on estimation of physical activity using accelerometers. *Journal of physiological anthropology*, 30(3), 119-127.

Pham, D. T., Dimov, S. S., & Nguyen, C. D. (2005). Selection of K in K-means clustering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 219(1), 103-119.

Peterek, T., Penhaker, M., Gajdoš, P., & Dohnálek, P. (2014). Comparison of classification algorithms for physical activity recognition. In *Innovations in bio-inspired computing and applications* (pp. 123-131). Springer, Cham.

Scikit learning. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html. Accessed on 18 Feb 2019

Grootendorst, M. (2019). Validating your Machine Learning Model. <https://towardsdatascience.com/validating-your-machine-learning-model-25b4c8643fb7> Retrieved on April 24, 2021.

Manuel Calzolari. (2021, April 3). manuel-calzolari/sklearn-genetic: sklearn-genetic 0.4.0 (Version 0.4.0). Zenodo. <http://doi.org/10.5281/zenodo.4661178>
IEEE. <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

Masaeli, M., Fung, G., & Dy, J. G. (2010, January). From transformation-based dimensionality reduction to feature selection. In *ICML*.

Mattila, J., Ding, H., Mattila, E., & Sarela, A. (2009, September). Mobile tools for home-based cardiac rehabilitation based on heart rate and movement activity analysis. In 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (pp. 6448-6452). IEEE.

Mannini, A., & Sabatini, A. M. (2010). Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, 10(2), 1154-1175.

McLachlan, Geoffrey J.; Do, Kim-Anh; Ambroise, Christophe (2004). *Analyzing microarray gene expression data*. Wiley.

Miller, Brad; Goldberg, David (1995). "Genetic Algorithms, Tournament Selection, and the Effects of Noise". *Complex Systems*. 9: 193–212. S2CID 6491320.

Montoye, A. H., Westgate, B. S., Fonley, M. R., & Pfeiffer, K. A. (2018). Cross-validation and out-of-sample testing of physical activity intensity predictions with a wrist-worn accelerometer. *Journal of Applied Physiology*, 124(5), 1284-1293.

Munther, A., Razif, R., AbuAlhaj, M., Anbar, M., & Nizam, S. (2016). A preliminary performance evaluation of K-means, KNN and EM unsupervised machine learning methods for network flow classification. *Int. J. Electr. Comput. Eng*, 6(2), 778.

Reiss & Stricker (2011). Towards global aerobic activity monitoring. In *Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments* (pp. 1-8).

Robnik-Šikonja, M., & Kononenko, I. (2003). Theoretical and empirical analysis of ReliefF and RReliefF. *Machine learning*, 53 (1), 23-69.

"Sampling Bias". *Medical Dictionary*. Archived from the original on 10 March 2016. Retrieved 21 April 2021.

Saez, Y., Baldominos, A., & Isasi, P. (2017). A comparison study of classifier algorithms for cross-person physical activity recognition. *Sensors*, 17(1), 66.

Sukor, A. A., Zakaria, A., & Rahim, N. A. (2018, March). Activity recognition using accelerometer sensor and machine learning classifiers. In *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)* (pp. 233-238).

Sunkad, Z. A. (2016, November). Feature selection and hyperparameter optimization of SVM for human activity recognition. In *2016 3rd International Conference on Soft Computing & Machine Intelligence (ISCMI)* (pp. 104-109). IEEE.

T. M. Mitchell, "Machine learning. 1997", Burr Ridge, IL: McGraw Hill, vol. 45, 1997.

Skotte, J., Korshøj, M., Kristiansen, J., Hanisch, C., & Holtermann, A. (2014). Detection of physical activity types using triaxial accelerometers. *Journal of physical activity and health*, 11(1), 76-84.

Sprint, G., Cook, D. J., & Schmitter-Edgecombe, M. (2016). Unsupervised detection and analysis of changes in everyday physical activity data. *Journal of biomedical informatics*, 63, 54-65.

Swain, P. H., & Hauska, H. (1977). The decision tree classifier: Design and potential. *IEEE Transactions on Geoscience Electronics*, 15(3), 142-147.

Troped, P. J., Oliveira, M. S., Matthews, C. E., Cromley, E. K., Melly, S. J., & Craig, B. A. (2008). Prediction of activity mode with global positioning system and accelerometer data. *Medicine & Science in Sports & Exercise*, 40(5), 972-978.

Vanwinckelen, Gitte (2 October 2019). *On Estimating Model Accuracy with Repeated Cross-Validation*. *lirias.kuleuven*. pp. 39–44. ISBN 9789461970442.

Van Der Maaten, L., Postma, E., & Van den Herik, J. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71), 13.

Literature

Van Kuppevelt, D., Heywood, J., Hamer, M., Sabia, S., Fitzsimons, E., & van Hees, V. (2019). Segmenting accelerometer data from daily life with unsupervised machine learning. *PloS one*, 14(1).

Wu, J., Jiang, C., Houston, D., Baker, D., & Delfino, R. (2011). Automated time activity classification based on global positioning system (GPS) tracking data. *Environmental Health*, 10(1), 1-13.

Wu, J., Feng, Y., & Sun, P. (2018). Sensor fusion for recognition of activities of daily living. *Sensors*, 18(11), 4029.

Yang, J. (2009, October). Toward physical activity diary: motion recognition using simple acceleration features with mobile phones. In *Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics* (pp. 1-10).

Zhao, S., Li, W., & Cao, J. (2018). A user-adaptive algorithm for activity recognition based on k-means clustering, local outlier factor, and multivariate gaussian distribution. *Sensors*, 18(6), 1850.

Zhang, M., & Sawchuk, A. A. (2011, November). A feature selection-based framework for human activity recognition using wearable multimodal sensors. In *BodyNets* (pp. 92-98).

8 Appendices

8.1 Appendix 1: Tables and Figures

Tables:

Table 8.1 GeneticSelection with chosen features by Extra Tree Classifier

Maximum number of features	Number of Features	Chosen Features	Accuracy Scores	Chosen Estimator	Computational Time
16	16	'sp_max', 'hori_dis', 'verti_vari', 'x_kurto', 'y_max', 'rang_y', 'z_stdev', 'mean_amplitude_y', 'mean_amplitude_z', 'steps_total', 'y_corr', 'x_signal_energy', 'x_maxAmp3', 'z_maxAmp', 'mean_total_amp', 'edd_1'	0.807 (0.806)	ExtraTreesClassifier()	1426.8s
30	30	'sp_mean', 'sp_max', 'verti_vari', 'distance', 'x_var', 'x_skew', 'y_mean', 'y_var', 'y_kurto', 'z_max', 'z_mean', 'interaquar_z', 'total_min', 'interaquar_total', 'time_interval_y', 'peak_width_z', 'steps_total', 'mean_amplitude_total', 'time_interval_total', 'x_corr', 'z_corr', 'zero_cros_x', 'zero_cros_y', 'aver_cros_z', 'zero_cros_total', 'power_z_maxAmp2', 'power_z_maxAmp3', 'total_maxAmp', 'power_total_maxAmp', 'edd_2'	0.816 (0.806)	ExtraTreesClassifier()	1607.8s
60	58	'sp_stdev', 'sp_mean', 'sp_max', 'verti_vari', 'distance', 'x_stdev', 'x_min', 'x_mean', 'x_median', 'x_var', 'x_skew', 'y_min',	0.824 (0.806)	ExtraTreesClassifier()	2151.5s

		'rang_y', 'y_var', 'y_skew', 'y_kurto', 'z_stdev', 'z_mean', 'rang_z', 'interaquar_z', 'total_max', 'total_median', 'total_var', 'interaquar_total', 'steps_x', 'mean_amplitude_x', 'time_interval_x', 'total_amplitude_y', 'peak_width_y', 'mean_amplitude_z', 'total_amplitude_z', 'time_interval_z', 'peak_width_total', 'x_corr', 'y_corr', 'z_corr', 'total_corr', 'zero_cros_x', 'aver_cros_x', 'zero_cros_y', 'zero_cros_z', 'aver_cros_z', 'aver_cros_total', 'x_maxAmp', 'power_x_maxAmp2', 'power_x_maxAmp3', 'mean_x_amp', 'y_maxAmp2', 'power_y_maxAmp', 'power_y_maxAmp2', 'z_maxAmp', 'z_maxAmp2', 'power_z_maxAmp', 'mean_total_amp', 'edd_1', 'edd_2', 'AI', 'VI'			
118	70	'sp_stdev', 'sp_mean', 'hori_dis', 'verti_vari', 'distance', 'x_stdev', 'x_min', 'x_max', 'rang_x', 'x_median', 'x_var', 'interaquar_x', 'x_skew', 'x_kurto', 'y_stdev', 'y_min', 'rang_y', 'interaquar_y', 'y_skew', 'z_stdev', 'z_mean', 'rang_z', 'z_var', 'z_skew', 'total_stdev', 'total_min', 'total_mean', 'total_median', 'interaquar_total', 'total_amplitude_x', 'peak_width_x', 'steps_y', 'mean_amplitude_y', 'time_interval_y', 'steps_z',	0.817 (0.806)	ExtraTreesClassifier()	2220.9s

		'mean_amplitude_z', 'total_amplitude_z', 'peak_width_z', 'steps_total', 'x_corr', 'y_corr', 'zero_cros_x', 'aver_cros_x', 'aver_cros_y', 'zero_cros_z', 'zero_cros_total', 'x_spectral_density', 'x_maxAmp2', ' x_maxAmp3', 'power_x_maxAmp', 'power_x_maxAmp2', 'mean_x_amp', 'y_spectral_density', 'y_signal_energy', 'y_maxAmp3', 'power_y_maxAmp', 'power_y_maxAmp2', 'z_spectral_density', 'z_signal_energy', 'z_maxAmp2', ' z_maxAmp3', 'power_z_maxAmp2', 'mean_z_amp', 'total_maxAmp2', 'power_total_maxAmp', 'power_total_maxAmp2', , 'mean_total_amp', 'edd_2', 'entropy_total', 'VI'			
--	--	--	--	--	--

Table 8.2 RFECV with chosen features by Extra Tree Classifier

Step	Minimum number of features	Number of Features selected	Chosen features	Accuracy Scores	Computational Time
1	1	56	'sp_stdev','sp_mean','sp_max','hori_dis', 'verti_vari', 'distance','x_stdev', 'x_max', 'x_median','x_var','interaquar_x','x_skew', 'y_stdev','y_var','interaquar_y','y_skew', 'z_stdev','z_min','z_var','interaquar_z', 'total_stdev', 'total_min', 'rang_total', 'total_var', 'interaquar_total', 'steps_x', 'mean_amplitude_x','total_amplitude_x', 'mean_amplitude_y','total_amplitude_y', 'peak_width_y','mean_amplitude_z', 'total_amplitude_z', 'peak_width_z','mean_amplitude_total', 'total_amplitude_total','x_corr', 'z_corr','total_corr','zero_cros_y', 'aver_cros_y', 'x_signal_energy', 'x_maxAmp','x_maxAmp2','x_maxAmp3', ',mean_x_amp','y_signal_energy', 'mean_y_amp','z_signal_energy', 'z_maxAmp2', 'z_maxAmp3', 'mean z amp', 'total signal energy',	0.826 (0.806)	2322.0s

			'edd 1', 'edd 2', 'AI'		
3	1	64	'sp_stdev', 'sp_mean', 'sp_max', 'hori_dis', 'verti_dis', 'verti_vari', 'distance', 'x_stdev', 'x_median', 'x_var', 'interaquar_x', 'x_skew', 'y_stdev', 'y_var', 'interaquar_y', 'y_skew', 'z_stdev', 'z_min', 'rang_z', 'z_var', 'interaquar_z', 'total_stdev', 'total_min', 'rang_total', 'total_var', 'interaquar_total', 'steps_x', 'mean_amplitude_x', 'total_amplitude_x', 'peak_width_x', 'mean_amplitude_y', 'total_amplitude_y', 'peak_width_y', 'steps_z', 'mean_amplitude_z', 'total_amplitude_z', 'peak_width_z', 'mean_amplitude_total', 'total_amplitude_total', 'x_corr', 'y_corr', 'z_corr', 'total_corr', 'aver_cros_x', 'zero_cros_y', 'aver_cros_y', 'x_signal_energy', 'x_maxAmp', 'x_maxAmp2', 'x_maxAmp3', 'power_x_maxAmp', 'mean_x_amp', 'y_signal_energy', 'mean_y_amp', 'z_signal_energy', 'z_maxAmp2', 'z_maxAmp3', 'mean_z_amp', 'total_spectral_density', 'total_signal_energy', 'mean_total_amp', 'edd 1', 'edd 2', 'AI'	0.824 (0.806)	778.9s

Figures:

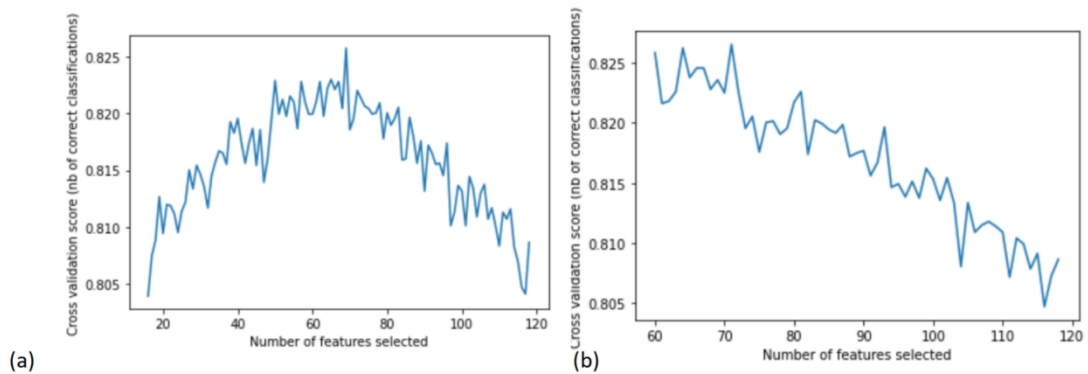


Figure 8.1 RFECV feature selection scores with the the number of features selected
 Figure (a) shows the results from the hyperparameter minimum number of features set as 16, step size set as 1. Figure (b) shows the results from the hyperparameter minimum number of features set as 60, step size set as 1.

8.2 Appendix 2: Code

Table 8.3 Overview of scripts containing important steps for the PA classification

Function	Description
Low pass filter	Compute the coefficients based on cutoff freq (Hz), sampling freq (Hz), and the order of the filter (amount of past values filters uses). This returns the gravity y and ignores the rest.

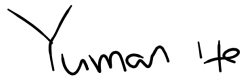
	<pre>def butter_lowpass (cutoff, fs, order=1): nyq = 0.5 * fs normal_cutoff = cutoff / nyq b, a = butter(order, normal_cutoff, btype='low', analog=False) return b, a def butter_lowpass_filter (data, cutoff, fs, order=1): b, a = butter_lowpass(cutoff, fs, order=order) y = lfilter(b, a, data) return y</pre>
GPS speed recalculation	<p>Recalculate the GPS speed by taken the mean values of the adjacent observations.</p> <pre>data['speed'] = data['speed'].rolling(3).mean()</pre>
Peak finding	<p>Calculate the signal's peak-related features in the time domain. The threshold parameter represents the vertical distance to its neighboring samples. The distance parameter represents the required minimal horizontal distance (≥ 1) in samples between neighbouring peaks.</p> <pre>peaks, _ = scipy.signal.find_peaks (signal, threshold = 20, distance = 2) prominences = scipy.signal.peak_prominences (signal, peaks, wlen=None)[0] steps = sensormotion.gait.step_count (peaks)</pre>
Zero/Mean crossing rate	<p>Calculate the times signal goes cross the zero and mean values in one time window</p> <pre>def getZeroCrossingRate(arr): my_array = np.array(arr) return float("{0:.2f}".format((((my_array[:-1] * my_array[1:]) < 0).sum())))) def getMeanCrossingRate(arr): return getZeroCrossingRate(np.array(arr) - np.mean(arr))</pre>
Power spectral density	<p>Calculate power spectral density by integration over spectral bandwidth in frequency domain</p> <pre>f_welch, S_xx_welch = scipy.signal.welch(signal, fs=fs)</pre>
Dominant frequencies of signals	<p>Calculate the top 3 dominate frequencies of signals in frequency domain</p> <pre>xdft = np.fft.fft(signal) xdf = pd.DataFrame((np.abs(xdft))) maxAmp, maxAmp2, maxAmp3 = xdf[0].nlargest(3)</pre>
Eigenvalues of dominant directions (EDD)	<p>Calculate the distinctive feature Eigenvalues of dominant directions that measures the corresponding relative motion magnitude along the vertical direction and the heading direction respectively.</p> <pre>def EDD(x, y, z): edd_mat = np.array([x, y, z]) cov_mat = np.cov(edd_mat.T) eigen_vals, eigen_vecs = np.linalg.eig(cov_mat) return eigen_vals[0], eigen_vals[1]</pre>
Signal Entropy	<p>Calculate the entropy of signals</p> <pre>antropy.entropy.spectral_entropy (ac, fs)</pre>
ReliefF	<pre>ReliefF (n_neighbors = 20, features_to_keep = 16)</pre>

Genetic selection	<i>selector = GeneticSelectionCV (estimator, cv = cv, verbose = 0,scoring = "accuracy", max_features = i, n_population = 50, crossover_proba = 0.35, mutation_proba = 0.015, n_generations = 10, tournament_size = 3)</i>
Recursive feature elimination	<i>selector = RFECV (estimator, min_features_to_select= min_features_to_select, step = step, cv = cv, scoring = 'accuracy')</i>

Personal Declaration

I hereby declare that the submitted thesis is the result of my own, independent work. All external sources are explicitly acknowledged in the thesis.

Chengdu, 30 April 2021

A handwritten signature in black ink that reads "Yuman He". The signature is written in a cursive style with a large 'Y' and a stylized 'H'.

Yuman He