GIMA
Geographical Information Management and Applications

Using Land-use Data to Improve Automatic Classification Accuracy of Machine Learning Models for Classifying Outdoor Sport Activities in GNSS-Tracks

Gido Stoop (5751209)

G.H.E.Stoop@students.uu.nl

GIMA Msc. Thesis

February 2022

Supervisor: Drs. C.W. Quak

Responsible professor: Prof.dr.ir. P.J.M. van Oosterom

Abstract

Getting sufficient amounts of physical activity are widely understood to improve general health and wellbeing. Understanding the patterns in sport-behaviour and its connection to land-use elements are vital for promoting physical activity and meeting global health goals set by the World Health Organisation. Collecting and analysing data on physical activity can help this understanding. Nowadays, nearly everyone collects spatial data in the form of GNSS tracks through their smart-devices. This data can be used to detect physical activities. However, raw spatial data lacks context and requires analysis, which can be time-consuming. For this purpose, various machine learning models were trained in this research that can automatically classify sport activities performed in GNSS tracks. Pre-labelled GNSS-tracks were used to train and test the models. Land-use data that corresponded with the GNSS tracks was also used to find out to what extend it could influence the models' classification accuracy. The model trained with the support vector machines' algorithm achieved the highest classification accuracy with a classification accuracy of 82.6%. Adding land-use data to the model also significantly increased its classification accuracy (+5.6%). Using land-use data in other machine learning algorithms also significantly improved their models' performance. However, in these models, not all land-use features were found to have a positive influence on the models' performance.

Acknowledgement

This research was conducted during the first semester of the 2021-2022 study year. I want to express my gratitude to my supervisor Wilko Quak for his support, time and feedback during the thesis-process. I want to thank Peter van Oosterom for being my responsible professor and providing me with feedback during the first phase of my thesis. I also want to thank Edward Verbree in aiding me in finding a suitable topic for the thesis. Most of this thesis was written at home due to the covid-pandemic. Fortunately, in the final weeks of my thesis I was able to work on my thesis along with my fellow students in the computer-rooms of the Vening Meinesz-building on the Uithof. This gave me extra motivation to work on my thesis and therefore I also want to thank my fellow students who were there is the final weeks of the thesis period.

List of abbreviations

ANN – Artificial Neural Network

API – Application Programming Interface

BN – Bayesian Network

CSV – Comma Separated Value format

CV – Cross Validation

DT – Decision Tree

GAPPA – Global Action Plan on Physical Activity

GLONASS – Global Navigation Satellite System (Russian)

GML – Geographic Markup Language

GNSS – Global Navigation Satellite System

GPC – Gaussian Process Qualifiers

GPS – Global Positioning System (American)

GPX – GPS Exchange Format

HAR – Human Activity Recognition

KNN – K-Nearest Neighbour

MLP – Multilayer Perceptron

OSM – Open Street Map

OOB – Out-of-bag

RDP – Ramer-Douglas-Peucker algorithm

RF – Random Forest

SVM – Support Vector Machines

WHO – World Health Organisation

Table of content

1. Introduction

Using the public space as a place for sports and exercise has been growing in popularity in recent years, and many community policies are aimed at improving public space as a place for people to exercise (Chacón-Borrego, Corral-Perniá, Martínez- Martínez & Casteñeda-Vázquez, 2018). This trend can also be observed in the use of fitness related applications, that record mainly cardiovascular endurance-based sport activities in the public space, e.g., running and cycling (Janssen et al., 2017). Location data on sport activities in the public space can improve the understanding of human sports behaviour and the interaction with space. However, the problem with raw location data is the lack of context. The kind of activity that is performed in a certain location data log can be detected by manually analysing the data, but this is certainly a time-costly process. In this research, machine learning algorithms will be used to solve this problem and automatically detect and categorise the sport activity performed in certain GNSS tracks in combination with land-use data that corresponds with the GNSS tracks. The aim is to review the potential of using land-use data as a supplement to GNSS data in machine learning models that classify sport activities performed in the GNSS data.

1.1 Practical relevance

Getting sufficient amounts of physical activity are widely understood and acknowledged to significantly increase health benefits and mitigate health risks. On the contrary, inactive lifestyles can cause severe health problems (Mardini et al. 2021). According to the WHO, one out of four adults globally do not meet physical activity recommendations. There is, however, a plan called the Global Action Plan on Physical Activity 2018-2030 (GAPPA) by the WHO which aims to improve physical activity by 2030 (World Health Organization, 2019). Physical activity is not only linked to sport activities but can also be linked to travel behaviour. For example, people that walk, bike, or use public transportation accumulate more physical activity then people taking the car and are more likely to meet health recommendations (Ellis et al., 2014).

To meet the goals in the GAPPA, an accurate estimation of physical activity type, duration and intensity are needed. This can increase the understanding of the link between physical exercise and health (Mardini et al. 2021) and the link between physical activity behaviour and the built environment (Ellis et al., 2014).

The ability to detect the sport activity in GNSS-tracks is not only useful for addressing the goals of GAPPA. It could be used for identifying training patterns and preventing injury (Rossi et al., 2018). Smartphones could use the classification to automatically adjust mobile phone settings. The information could also be of value for effective advertisement and consumers could also be interested in learning about their sports patterns (Martin et al., 2017). The classification of activities could also be used in map-matching since different activities are performed on different network segments. Finally, the information could be used for managing infrastructure as well as to plan and design future facilities (Shafique & Hato, 2016).

1.2 Scientific relevance

Recently, machine learning algorithms in combination with Global Navigational Satellite System (GNSS) data have been used to identify physical mobility (Wu, Yang & Jing, 2016). However, most of this research focusses on automatically recognising modes of transport and very little research has been done on machine learning methods to identify sport activities. Besides, most of the research that is done uses data from sensors in the classification process (Wu, Yang & Jing, 2016). Little research has been done on using land-use data as supplementary data in classification problems. This master thesis will focus on building a machine learning model with GNSS and land-use features and assessing the

influence of the land-use features on the classification of specific sport activities exerted in GNSS track data.

1.3 Research objective

As stated before, the aim of the research is to review the potential of using land-use features for training machine learning models that can classify sport activities exerted in GNSS data. To give a good analysis on the potential of using land-use features in machine learning models, this research strives to find the machine learning algorithms and hyperparameters that produce the highest classification accuracy. The research also strives to extract as much relevant information as possible from the raw GNSS and land-use data to train and test the machine learning models.

1.3.1 Main question

- To what extent do land-use features improve the machine learning models' ability to correctly classify the kind of outdoor sport practices in GNSS-tracks?

1.3.2 Sub-questions

1) What is machine learning?
2) What is the state of the art in outdoor sports detection in GNSS-data based tracks using machine learning?
3) What features need to be extracted from the GNSS- and land-use data to detect the kind of outdoor sport practiced in the recorded tracks?
4) What machine learning algorithms would be suitable for outdoor sport activity detection in GNSS- and land-use data?
5) How can we validate the machine learning models for detecting the kind of outdoor sport practiced in recorded GNSS-data?
6) To what extent are machine learning models able to classify the kind of outdoor sport practices in pre-recorded GNSS-data based tracks using related land-use features?

1.3.3 Out of the scope for this research

It is also important to note what is out of the scope for this research. This research will not be about creating a model with the highest classification accuracy but is rather research on which methodology could be used to increase the classification accuracy using land-use data. It is also not the goal to be able to do real time or 'on-the-fly' predictions. It is not in the scope of this project to find the starting and ending point for each activity, as this will make the research drastically more difficult. Finally, it is not in the scope of this project to go in depth into the machine learning algorithms. Basic knowledge and techniques will be described, but R will be used to do the statistical analysis.

1.3.4 Outline of the research

This thesis consists of 7 chapters in total, of which this introduction is the first one. The other chapters are:

Chapter 2: The theoretical framework, which consist of an in-depth literature review to acquire an understanding of the state-of-the art research and methods in the field. General concepts that are reviewed is human activity recognition (HAR), fitness activity trackers, GNSS technology, research using GNSS, research using machine learning techniques and the machine learning techniques themselves.

Chapter 3: The methodology, which consist of a detailed description of how the research was conducted to ensure quality and reproducibility. The data is described, both GNSS- as land-use data, how the features for the machine learning models are calculated and finally which machine learning methods are used.

Chapter 4: Results, which consist of an evaluation of the data and features, an evaluation of the machine learning models and an evaluation on how (land-use) features influence the models' performances.

Chapter 5: Discussion, which consist of a deeper analysis of the results, that compares the results of the different models, tries to interpret performances, find general trends, and analyse the importance of (land-use) features.

Chapter 6: Conclusion, which answers the main question of this thesis.

Chapter 7: Reflection and recommendation: which reflects on the shortcomings of the research, and which provides recommendations and suggestions for future work.

2. Literature review

This literature review will give an insight into the key concepts and the state-of -the art research and technologies regarding sport activity classification using GNSS and land-use data. These key concepts are used for answering the questions 'What is machine learning?' and 'What is the state of the art in outdoor sports detection in GNSS-data based tracks using machine learning?'.

Firstly, the line of research regarding human activity recognition (HAR) will be discussed briefly in paragraph 2.1. Since the focus of this research is sports activities, paragraph 2.2 will go into technologies that can record these sport activities. The advances in- and popularity of wearable fitness activity trackers will be discussed, then how the location technology works (using GNSS sensors) and what advances are made in mobile phone GNSS technology. Paragraph 2.3 will review the advanced made in GNSS based classification research, firstly in general, and secondly specifically using machine learning technologies. Then, in the final paragraph (2.4) a general overview will be presented on state-of-the art machine learning technologies.

2.1 Human Activity Recognition (HAR)

Automatic classification of activities is nothing new. For years, researchers have been using various sensors to learn about Human Activity Recognition (HAR). HAR is a line of research that is focussed on automatically recognising human activities. Some of the main application of the information gained by HAR are crowd surveillance, healthcare support, population security and lifestyle and behaviour tracking. HAR often requires dedicated hardware, sophisticated engineering and statistical- and computational techniques. Another characteristic of HAR is that it usually consists of five steps, namely sensing, pre-processing, feature extraction, training, and classification (Ferrari, Mucucci, Mobilio & Nopoletano, 2021). Previous studies have attempted to survey the current status of the HAR field, however, these studies vary greatly in the type of activities categorised and experimental setups used. This makes comparing these studies hard to compare (Shoaib, Bosch, Incel & Scholten, 2015). Therefore, this thesis will focus more on the HAR studies that are closely related to this study, like research on sport-activities and research done using mobile device data.

2.2 Fitness activity trackers

Nowadays, smartphones are equipped with various sensors that can be used for sensing and data collection. Sensors included in smartphones that can sense activity are e.g., an accelerometer, a gyroscope, a magnetometer, a microphone, and a GPS/GNSS-chipset (Shoaib et al., 2015). In 2021, smartphone ownership is estimated at around 6.4 billion people worldwide, making up for about 82.5% of the global population (Statista, 2021a). These developments bring forth a large source of data which can be useful for analysis and innovations in different fields. According to a survey by Carto (2017), 94% of the participating medium and large organisations collects location data. Already 54% of the participating organisations used to collect location data through mobile devices and applications. This is the second most popular way of collecting location data after website and web-based applications (57%). Accurate estimations of physical activity can be achieved through mobility research. Rapid advancements have been made in tracking technology in recent years. (Ferri, 2016). Another development regarding smartphone data collection is the rising popularity of fitness applications and mobile fitness trackers.

In Europe and the US, the use of health-related applications and the use of electronic devices for monitoring health has grown extensively. These apps could be used to provide support or monitor those users that don't have access to any professional trainers or coaches (Janssen et al., 2017). Since most people have a smartphone, and apps are often free of charge or relatively cheap, the apps are accessible to nearly anyone. The use of sports apps is mainly related to individual recreational sports,

such as walking, running, cycling and fitness (Janssen et al., 2020). 26.5% of US adults used a health and fitness related application at least once a month in 2020 compared to 19.2% of all US adults in 2018 (Statista, 2021b). There is no significant relation between app users and their gender or education. The most popular sport that is monitored is running. Currently almost 10% of the EU-28 population and over 10% of the US population partake in recreational running. This is in line with the rise in popularity of other recreational, unorganized, and lighter forms of sports with a health-related focus (Janssen et al., 2017).

According to a survey conducted by Pew Research Center in 2019, about 21% of all adults in the US say they regularly wear and use smart watches or other wearable fitness trackers. People using fitness trackers are also found to be more willing to share data of the devices with health researchers. 53% of users find this acceptable, while among non-users this is only 38% (Vogel, 2020).

2.2.1 GNSS technologies

These fitness tracking apps for smartphones and other wearable devices use a variety of sensors to collect data. This study will only use the data acquired via the GPS/GNSS-chipset and will therefore be the focus in this theoretical framework.

The GPS/GNSS-chipset in a mobile device acquires data global navigational satellite system (GNSS). The first GNSS with global coverage was the Global Positioning System (GPS), that was launched in 1978 by the U.S. Department of Defense. It is the worlds most utilized satellite navigation system and the terms GPS and GNSS are often used interchangeably. Therefore, GPS has become a synecdoche for GNSS's (Frousiakis, 2018). The GPS system calculates the location of the GPS receiver based on the emission of synchronized radio signals by multiple satellites that orbit around the earth. The satellites are equipped with atomic clocks that emits the time and position of the satellite to the receiver with great precision. The time difference between the emitted signals of the various satellites allows the receiver to calculate its distance from the satellite using trilateration. To calculate it's 2D position, three satellite measurements are required. For a 3D position, four satellite signals are required (Schutz & Chambaz, 1997).

Other GNSS constellations work in a similar manner. There are four global GNSS constellations. Russia's GNSS is called GLONASS and was designed in the 1970's as Russia's military positioning system. China's GNSS is called BeiDou and has been operational since 2000 and is on the rise to overtake GPS in terms of commercial global usage. Finally, the EU's GNSS is called Galileo and has global coverage since 2020 (Frousiakis, 2018). The signals of satellites from multiple GNSS constellations can be combined to increase the accuracy. This is especially beneficial in areas where the satellite signals are blocked, like urban areas and parts close to the earth's magnetic equator (Li et al., 2015). In the past few decades, remarkable advances have been made in GNSS technology and the systems are changed to have a higher level of interoperability and consistency with other systems. This enables the use of multiple constellations, not only for high-grade receivers, but also for low-cost handheld receivers like smartphones (Paziewski, 2020).

2.2.2 GNSS systems in mobile devices

The GNSS constellations that are used for location services in the smartphone are dependent on the GNSS-chipsets that is used when manufacturing the phone. Most phones use multi-GNSS, but the chipsets used can vary from phone to phone (GalileoGNSS, 2017). Since not all devices are using the same GNSS constellations, and chip quality varies in smartphones, there can be significant performance differences in terms of accuracy per smartphone (Paziewski, Fortunato, Mazzoni &

Odolinski, 2021). For example, the first phone using the Galileo system was launched in late 2016 (GalileoGNSS, 2016) and in September 2017, the first phone was launched using a dual frequency GNSS chipset i.e., using multiple frequencies that are emitted by the GNSS satellites (Paziewski, 2020). Figure 2.1 shows an example of which satellites and constellations GNSS-chipsets can use. In the chipset of a OnePlus 3T from 2016.

There are a few problems when using mobile phones for HAR. Mobile phones are considered resource-limited devices, as they don't always possess enough battery resources to be able to continuously run activity recognition systems. However, advancements are being made in battery power and HAR capabilities. HAR can also be done 'offline', meaning that the pre-processing and classification is being done on a server instead of on the resources of the device (Shoaib et al., 2015). In this case, the data-collection still needs to be done locally on the smartphone. Another problem that occurs due to the battery limitations of smart devices is the duty cycling mode. Duty cycling is a mode which is active in most smart-devices and make for a discontinuous manner of location collection to prevent battery drainage. When the smart-devices location services are not being used, it will stop with continuous data collection and go into a battery friendlier mode. Only in the latest android versions, duty cycling can be switched off (Paziewski, 2020).

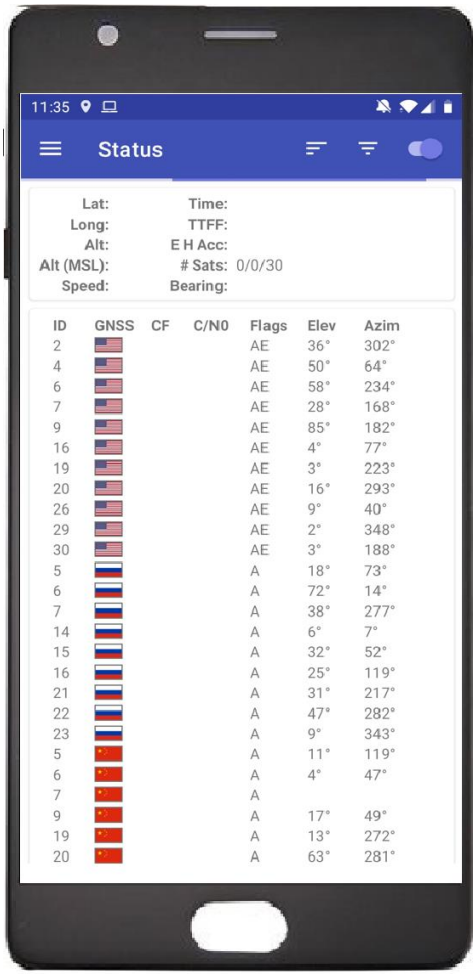


Figure 2.1 Satellites and Constellation snapshot by the OnePlus 3T from 2016

2.3 Research using GNSS

Traditionally, movement patterns of people were researched using surveys. However, self-reporting surveys had several shortcomings. Quite often, respondents overlooked short trips and activities. The trips were sometimes reported out of sequence and the departure and arrival times were mostly very approximate. Besides this, surveying was very time-consuming (Shafique & Hato, 2016). Another reason why GNSS has drawn the attention of researchers is the low-response and high incompletion rates of travel surveys (Lee & Kwan, 2018). The advantage of GNSS as opposed to surveys is the high accuracy.

GNSS devices generally produces data that includes ID, longitude, latitude, timestamp, horizontal dilution of precision, vertical dilution of precision, the number of satellites in view, altitude, heading and instantaneous speed (Allahbakhshi, Conrow, Naimi & Weibel, 2020). There are also disadvantages of GNSS-data as opposed to traditional surveying methods. First, GNSS data is raw data. Given the amount of data GNSS provides, it would be extremely time-consuming to manually interpret all the data. Additional data-analysis, mathematical models or surveying is needed to determine the mode of transport used or the type of activity done during the collection of the GNSS data (Shafique & Hato, 2016) and when modal or activity changes occur in the track. Besides this, GNSS-data is less accurate in places where signal detection is less, like indoors, tunnels or urban areas with a lot of high-rise buildings (Ellis et al., 2014).

2.3.1 Past research on Human Activity Recognition using machine learning

Past research on detecting sports activities have been using various methods, like PCA analysis (Ross, Dowling, Troje, Fischer & Graham, 2018), machine learning techniques (Ferri, 2016; Ouchi & Doi, 2012) and deep learning techniques (Clouthier, Ross & Graham, 2020) to classify sport activities.

Research on HAR is mostly focussed on finding methods that bring the highest accuracy in correctly detecting the travel mode or activity done in the GPS track. Several factors influence the detection accuracy. Firstly, the variables used in the application, secondly, the algorithm used in the application and thirdly, the training and testing data used in the application.

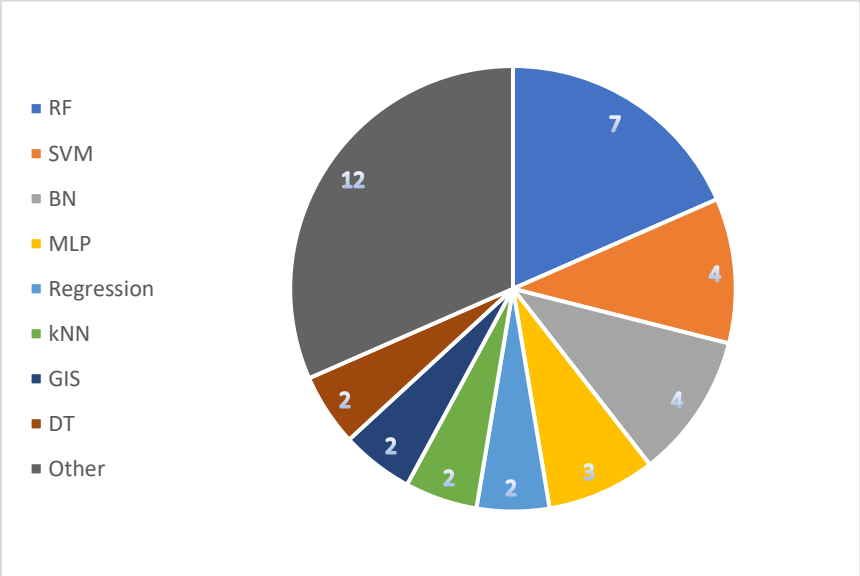
To achieve the optimal detection accuracy, studies have used various variables that can be derived from GPS data. These variables mostly include average speed, average acceleration, trip distance and duration (Wu, Yang & Jing, 2016). Another common variable additionally used is average heading change (Shafique & Hato, 2016; Fang et al., 2018; Wu, Yang & Jing, 2016). Some studies will not use the average speed, but the 95^{th} percentile or 75^{th} percentile of speed and acceleration instead to exclude random errors. (Xiao, Cheng & Zhang, 2019; Xiao, Juan & Zhang, 2015; Zong et al., 2017; Zong et al., 2015). These studies also include the standard deviation of speed and the standard deviation of acceleration as an extra variable as well as the low-speed point rate (the total time that the GPS is stationary or very slowly moving). Lee & Kwan (2018) use intervals two different kinds of speed variables, instantaneous level, and interval level. Finally, Ellis et al. (2014) uses more variables like average number of satellites used and average signal-to-noise ratio.

Besides variables that are constructed from GPS-data, there can also be variables that are constructed from other data sources. Accelerometers are found in most modern smartphones and can detect acceleration along three axes with respect to the gravitational force (Shafique & Hato, 2016). Lee & Kwan (2018) use accelerometer data for identifying indoor activities, where GPS signal is less accurate. This increased their accuracy from 75% with only GPS-data to about 95% with GPS- and accelerometer data. Furthermore, accelerometer data is often used as complementary data to GPS data and leads to higher classification accuracy (Feng & Timmerman, 2013; Martin et al., 2017; Tamura et al., 2018; Allahbakhshi, Conrow, Naimi & Weibel, 2020; Mardini et al., 2021; Ellis et al, 2014). In some cases, spatial context data is used to further improve results. Allahbakhsi et al. (2020) and Zong et al. (2017) use map matching techniques to match GPS-tracks to networks and improve spatial accuracy. Shafique & Hato (2016) used spatial data to add variables to the data like proximity to bus stations, proximity to rail line trajectories and zip codes. Finally, Fang et al., (2018) use network data to process the data and assign missing values to points that deviate too much from the road network.

Most papers will use a few different machine learning algorithms, to find the one that will give the most accurate classifications. Commonly used algorithms include k-nearest neighbour (kNN), support-vector machines (SVM), Bayesian networks (BN), decision trees (DT), Gaussian process qualifiers (GPC), random forest (RF) and artificial neural networks (ANN). Figure 2.2 shows the frequency of methods used in HAR of 13 different papers. The methods that used the RF algorithm were in general the most successful and could be considered the superior algorithm for this purpose (Ferri, 2016; Ellis et al., 2014; Lee & Kwan, 2018; Martin et al., 2017; Feng & Timmerman, 2015). Xiao, Cheng and Zhang (2019) found that GPC was the most suitable for their case, closely followed by ANN. While another paper by Xiao, Juan & Zhang (2015) found BN to be the most suitable for their research. Finally, Li et al. (2020) found that ANN was the most accurate algorithm. All studies give a classification accuracy per algorithm, but since the studies vary in focus and methods, the classification accuracies are not suitable for comparison. All papers in this literature review, its variables used and its classification accuracies per algorithm can be found in appendix A. In this literature review most of these papers focus on transportation mode classification and not sport-activity classification. Sport-classification using GNSS

data is less extensively researched. Classifying sport-activities might find different success rates with different algorithms.

Figure 2.2 Frequency of methods used for classification from 13 studies



2.4 Machine learning techniques

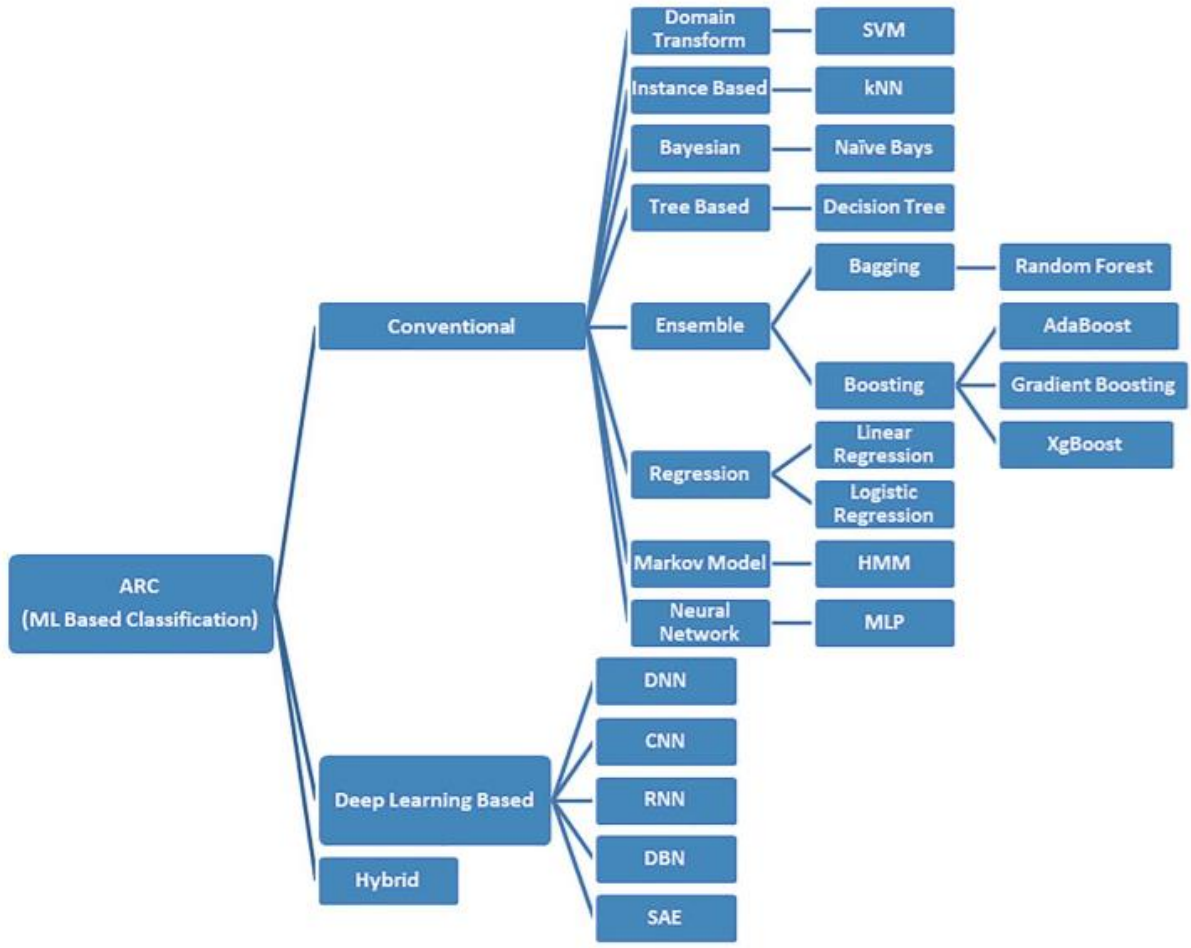
Research in HAR and travel mode detection often uses machine learning techniques to automatically categorize its activity with high accuracy. To use machine learning and understand the outcomes of this research, it is important to understand what machine learning is, how it works, and how a high accuracy of classification can be achieved.

The field of machine learning is always trying to make computer programs that can improve with experience. The name 'machine learning' refers to the idea that combining computer algorithms and large amounts of data can make the machine (the computer) learn. By using iterative processes and training data, the machine can improve itself in the given task. For example, the goal is to automatically separate spam-mail from regular e-mail. It is hard to precisely tell the computer when something is considered spam, but it is easy to collect thousands of examples of spam messages and regular e-mails and let the computer learn itself. The machine can be trained by using data that has already been classified, so it can recognise patterns and attributes to classify upon. The model can then be used to classify new data, based on what it learned from the training data (Alpaydin, 2014a). Other examples of machine learning applications are detecting fraudulent credit card transactions, autonomous vehicle driving and information filtering systems that learn users' reading preferences (Mitchell, 1997). Machine learning can also be applied to visual recognition, speech recognition and robotics. Recognizing faces is an everyday task for people, but it is an unconscious process causing people to be unable to explain the process to a computer. The machine learning model can recognize patterns in data and is therefore able to classify the data accordingly (Alpaydin, 2014a). There are two ways of classifying data using machine learning. One is supervised learning, in which the input and the output are provided by the supervisor. The other one is unsupervised learning, where no output is given for the input data and the machines job is to find data-clusters themselves (Alpaydin, 2014a).

There is a broad selection of different machine learning algorithms to choose from (Figure 2.3) that can be classified in three main categories (Chaurasia & Reddy, 2021). In conventional methods (or traditional machine learning) the features are hand-crafted by the user and fed to the algorithm, which

in turn does the classification and delivers the output. In deep learning the features are not crafted by hand, but rather, the algorithm learns to identify features (Ferrari et al., 2021). The last category are hybrid algorithms, which make use of a combination of both hand-crafted and computer identified features (Chaurasia & Reddy, 2021).

Figure 2.3 Activity Recognition Classification based on machine learning (Chaurasia & Reddy, 2021)



The advantage of using conventional methods and using hand-crafted features is the features calculation simplicity and low computational complexity. Disadvantages are the high dependency on the knowledge of the person selecting the features. Data needs to be investigated beforehand, and even then, it is still not always clear what features are likely to work best (Ferrari et al., 2021). However, after the model is trained, conventional models can also give quite a good indication of what the importance of each feature is in the classification process. This is different for deep learning models, as the computer crafted parameters are hard to interpret. Deep learning eliminates this problem, but deep learning also has its inherent disadvantages. Deep learning, on the contrary, requires high-end machines with large computing power. Deep learning algorithms take a large amount of data and many parameters to train. It can take up to weeks to train an algorithm, compared to a few hours at maximum for traditional machine learning algorithm (Mahapatra, 2018). Due to the amount of time, processing power and data required the deep learning algorithms will not be useful for this research. Besides deep learning there are many conventional machine learning techniques, in this paper there will be a focus on a few of the most popular and successful algorithms in HAR. These are, SVM, Random Forest and MLP.

The support vector machines algorithm (SVM) makes use of the quality of lines and hyperplanes to separate data classes. In many cases, data is not easily separable by a line or a hyperplane. The SVM therefore uses 'kernels' to transform the data-dimensions without affecting the features. In essence the SVM algorithm only handles binary classifications, but multiclass classifications can be achieved through a variety of methods (Noble, 2006).

The random forest algorithm (RF) is based on the principle of decision trees. However, decision trees are not very suited for classifying complex datasets. The random forest is a tree-based method, that builds multiple trees with random features and randomly selected feature values. The outcome of the multiple random trees will be used to determine the most likely class for each instance (Ho, 1995).

The multilayer perceptron algorithm (MLP) is a part of the broader tree of artificial neural networks (ANN). MLP's can be defined as a network in which input, and output is connected through nodes with high computational powers. The idea is that the network simulates the processing patterns in the brain. ANN's are 'trained' to produce a usable input-output relationship. In the training phase, parameters in the network are finetuned to realise the best input-output relation (Liao & Wen, 2007). This is called 'supervised' learning.

Above was a very brief description of the algorithms used in this research. A more detailed description of the machine learning algorithms and the application of the algorithms for this research can be found in the methodology chapter.

3. Methodology

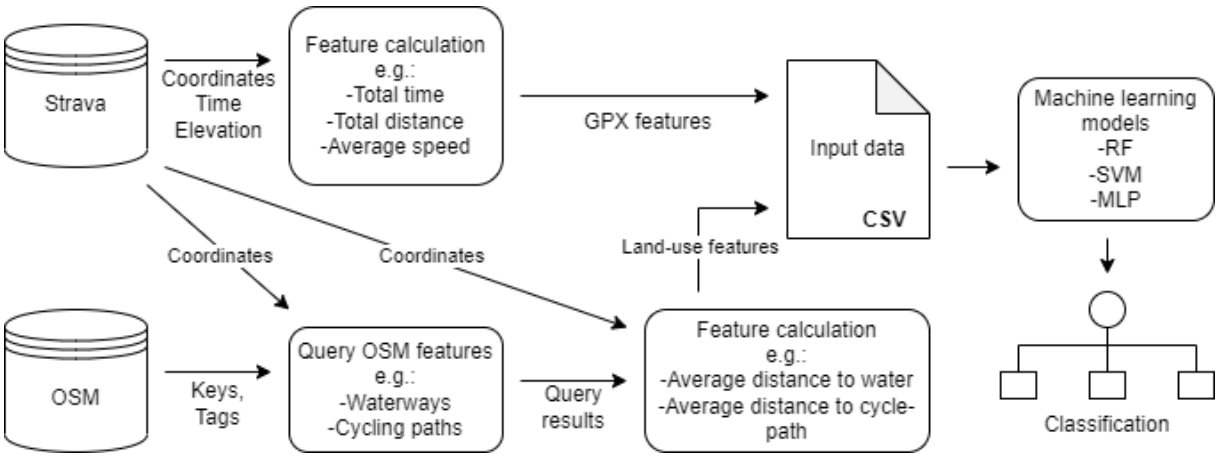
To answer the main question of this research, all sub-questions must be answered as well. In the theoretical framework, some sub-questions have been answered regarding the state-of-the-art research in sport activity recognition (sub-question 1) and machine learning techniques (sub-question 2). Other sub-questions, like sub-question 3, 4 and 5, have been discussed in the theoretic framework, but are not yet answered. This is done in this methodology chapter.

Sub-question 3 is about what features need to be constructed from the raw-GNSS data and land-use data to classify the sports activities performed. To know what variables are extracted from the GNSS data, it is required to discuss the data and data structure that are used in this research and how it is obtained. This is discussed in the paragraph 3.1. Then it is discussed how the extracted data is to calculate features that are used in the machine learning models. This is discussed in paragraph 3.2.

Sub-question 4 is about what machine learning methods are most suitable for this research. This is discussed in paragraph 3.3. This paragraph also goes in depth on how the machine learning method is executed and with which hyperparameters. The kind of statistical output the various machine learning techniques produces are also discussed. Later in the same paragraph sub-question 5, which is about how the machine learning model is validated and tweaked in this research, is also answered.

Figure 3.1 displays a schematic overview of the process that will be described in this chapter. First, data from Strava and OpenStreetMap (OSM) is required. Then, features will be calculated using the Strava data. Land-use data is queried from the OpenStreetMap database using the coordinates of the Strava tracks. Additional features are calculated based on the Strava data and the queried data. This data is saved in a tabular format and used for training and testing the machine learning models. These models will be able to classify the data that is given to the model.

Figure 3.1 Schematic overview of the classification process



3.1 Data overview

To create the features that are used in the machine learning models, two types of data are collected. Firstly, the GNSS tracks forms the main supplier of the data for the creation of features. Secondly, land-use data is used as additional data related to the GNSS data to create more location specific features.

3.1.1 GPX data

As discussed in the theoretical framework, machine learning models often require large amounts of data to deliver reliable results. Large datasets exist with GNSS data, however, these datasets are not publicly available (Carto, 2017). Another requirement for the dataset is that the data is already labelled with its corresponding sports activity. Each track should also contain only one type of activity from the

beginning until the end. Finally, the data-point interval must be consistent, so there are no large gaps in data recordings. To eliminate the problem of phones in 'duty cycle' mode (Paziewski, 2020), the activity must be actively recorded by some application on the smartphone or smart device.

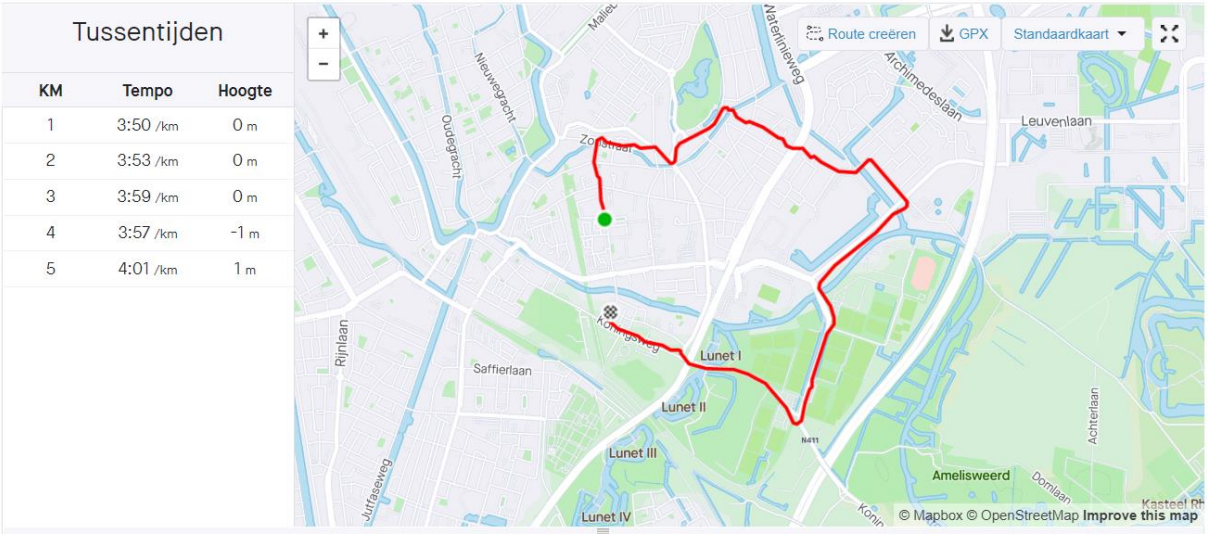
The data that is used for this research is self-recorded and labelled by athletes of all skill levels in a fitness application called Strava. As of March 2021, Strava has over 76 million users and over 1.1 billion registered GPS-segments (Curry, 2021). In the app, it is mandatory to select the type of activity that is about to be performed. All the activities are saved to a personal profile. All users can extract their activities in GPS Exchange Format (GPX) at once in a comprehensive ZIP archive format. The activities can also be downloaded from other users' profiles using an integrated download button from Strava or additional browser plug-ins.

The GPX file contains data about the date, track name and activity type. Per observation it contains data about the longitude, latitude, elevation, and a timestamp. This data is used as testing and training data for the algorithms. The data is suitable for machine learning, as all GPS-segments contain labels on the activity that is performed, and because the segments have a clear beginning and ending of the activity. In Figure 3.2, a small fragment of a downloaded GPX track is shown. 'name' refers to the name given to the activity by the user. 'type' is the type of activity selected by the user. 'trkpt', 'ele' and 'time' are the location, elevation, and timestamp respectively. In Strava, number 9 refers to running. In figure 3.3, the Strava interface is shown of the GPX track from figure 3.2.

Figure 3.2 Raw data of a Strava GPX track in GML format

```
<metadata>
 <time>2021-03-03T09:02:18Z</time>
</metadata>
<trk>
 <name>5k</name>
 <type>9</type>
 <trkseg>
  <trkpt lat="52.0817960" lon="5.1336370">
   <ele>4.3</ele>
   <time>2021-03-03T09:02:18Z</time>
  </trkpt>
  <trkpt lat="52.0818170" lon="5.1336760">
   <ele>4.3</ele>
   <time>2021-03-03T09:02:19Z</time>
  </trkpt>
  <trkpt lat="52.0818240" lon="5.1336820">
   <ele>4.3</ele>
   <time>2021-03-03T09:02:20Z</time>
  </trkpt>
```

Figure 3.3 Strava interface of an activity



The data that is collected features a collection of 8 sports that can be recorded on Strava. These will be biking, running, walking, hiking, ice-skating, inline-skating, kayaking/canoeing, and swimming. The data is collected from a random sample in order to get a diverse set of training data, including various athletes of all skill levels.

Figure 3.4 Strava scrapers' data collection

```
Running:  251
Cycling:  251
Walking:  119
Hiking:  17
Ice-Skating:  0
Inline-Skating:  1
Kayaking:  0
Swimming:  5
Total downloads:  644
```

To collect all the data, a script is written to automatically extract random GPX tracks from Strava. This script 'scrapes' tracks from the website. After running the scraper, some activities still require manual searching, as these activity types are uncommon and are hardly found using the scripts random search approach. Figure 3.4 shows the data collection using the random search approach of the Strava scraper. Strava does however have groups for athletes to join in which specific activities are recorded. For this research, data was collected from these groups for ice-skating, inline-skating, kayaking/canoeing, and swimming. This does however mean that not all data is collected completely random, as the Strava groups are often geographically centred around one place and athletes might have personal connections. Still, widespread coverage is reached. Figure 3.5 shows the geographical distribution of the activities.

Figure 3.5 Geographic distribution of the Strava tracks

While scraping and downloading public personal information is not illegal, it is sometimes considered unethical to do so. Richards & King (2014) theorize that while people (voluntarily) create data throughout the day and companies, governments or other institutions and legally allowed to use this data, it can be done in an unethical way. Richards & King argue that people are unaware of the sensitivity, magnitude, and characteristics of the data they create, and the user's big data should therefore also take responsibility in protecting the privacy of people. The data should therefore be pre-processed in a way that makes reverse engineering the data near impossible. In this research, the same is done. The data is made anonymous in the data processing phase. This is done by removing the title and date from the track. The Strava-id is replaced by a newly generated primary key. After pre-processing, the tabular data also doesn't contain any spatial data anymore. The tracks used in this research are deleted after the research is performed, as due to the characteristics of GNSS-data, anonymity can otherwise not be guaranteed.

In total, 1397 tracks were collected and used for this research. The aim is to collect around 250 tracks for each sport activity. However, some activity types were very time costly to find and due to time restrictions, the research is conducted with less than 250 tracks for some activity types. Table 3.1 shows the number of tracks used per activity type.

Table 3.1 Tracks used per activity type

Activity type	**Total tracks**
Running	242
Cycling	252
Walking	249
Hiking	157
Ice-Skating	85
Inline-Skating	158
Kayaking/Canoeing	113
Swimming	141

The Strava raw data is written in the Geographic Markup Language (GML) and saved in a GPX file format, this is converted to tabular data in order to be used in the machine learning algorithms. To extract the variables, a Python library is utilized called GpxPy. This is a GPX file parser that can parse information from GPX and XML documents (Krajina, 2021). These values are saved using a Python script. This script is also used to calculate the features and save them as tabular data in a Comma Separated Value (CSV) file data format. After feature extraction phase, each track is a row in the tabular data file containing continuous values for all features.

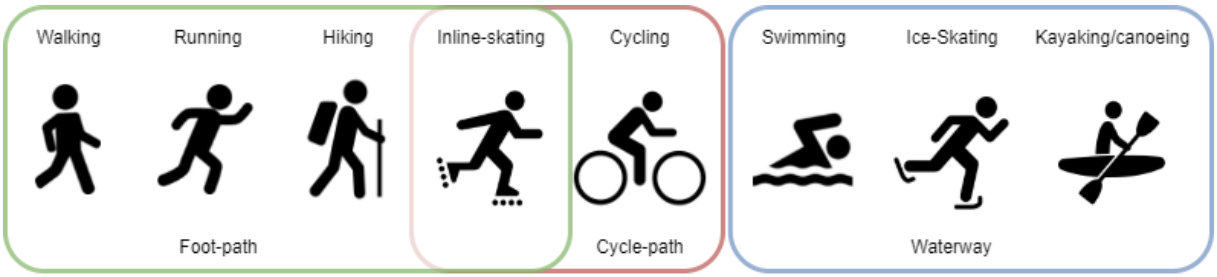
Usually, raw GPS data requires data smoothing techniques to decrease the impact of random errors. Popular statistical smoothing techniques include, among others, the Kalman filter, least squares spline approximation and kernel-based smoothing (Jun, Guensler & Ogle, 2005). It is very likely that Strava incorporates some smoothing technique in their application. Unfortunately, information on what techniques are used in the application is not publicly available. Considering the data used, no additional smoothing of the data deemed necessary.

3.1.2 Land use data

In this research, land use data is used to improve the machine learning model's accuracy by providing information on the relationship between the GPX tracks and the nearby spatial elements. To provide these spatial elements, Open Street Map (OSM) is used. OSM is an open collaborative mapping project, which offers free editable geographical data of the world (OpenStreetMap, n.d.).

The land-use data used in this research is the proximity of the GPX track to waterways, foot-paths and cycle-paths. Waterways include all rivers, canals, streams, and such, as well as bays, oceans, and seas. It also includes man made water features like swimming pools and ice-rinks. Foot-paths include all features intended exclusively for walking or running, like sidewalks, pedestrian zones, tartan surfaces or unpaved paths. Cycle paths include all features that allow cycling, like smaller roads, painted bicycle infrastructure on the road or sidewalk and exclusive bike paths (Appendix C). The land-use categories are expected to correspond with certain sport activities that will be categorized in this research (Figure 3.6). Where walking, running, and hiking are done in the proximity of foot-paths, cycling is done in the proximity of cycling-paths and swimming, ice-skating, and kayaking/canoeing is done in the proximity of waterways. Inline skating can be performed in the proximity of both foot- and cycling-paths.

Figure 3.6 Expected corresponding land-use types per activity

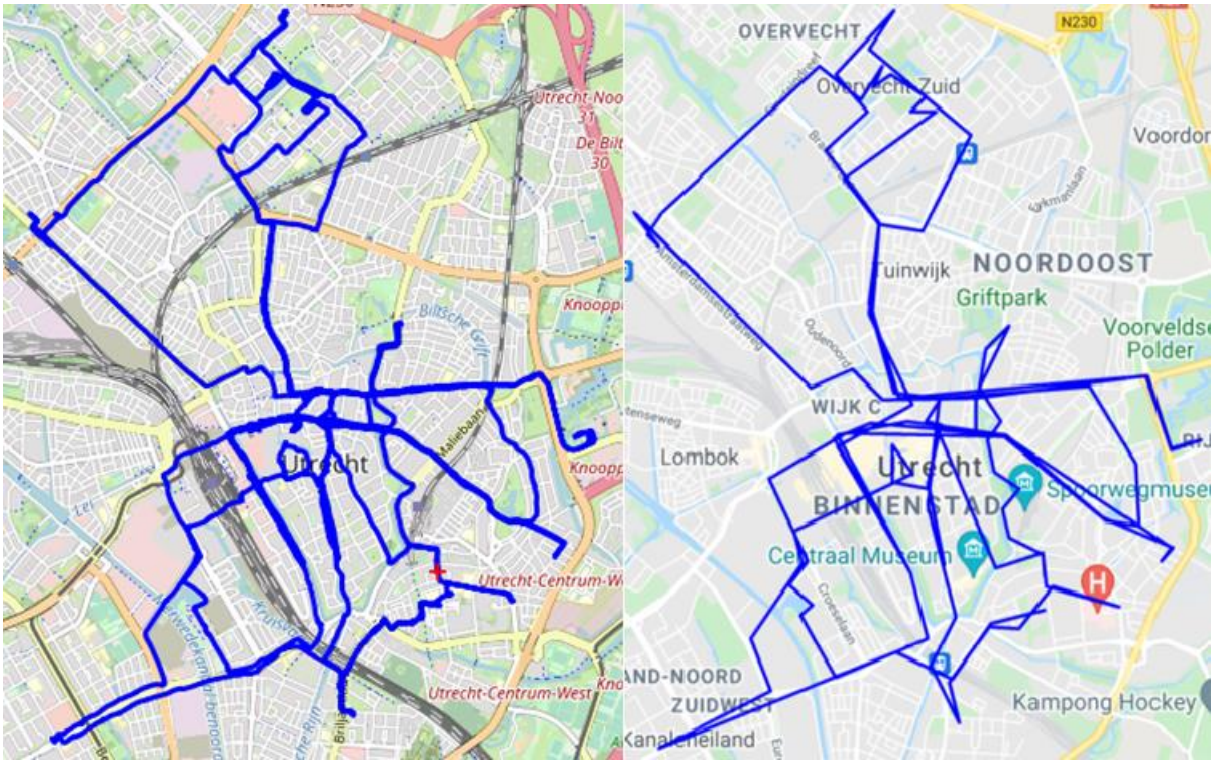


For querying the data, the Overpass API is used. This is a read-only application programming interface (API) that acts as a database over the web. Queries can be sent to the client via the API and returns the queried results. It can deliver roughly 10 million elements in minutes by selecting them based on criteria like object type and proximity (OSM Wiki, 2021). A query is written that will obtain all nodes of waterways, cycle-paths and foot-paths in the proximity of 50 meters from the track's GPS points. The keys and values used in the query can be found in Appendix C. The query is written in the OverpassQL and is tested in the Overpass Turbo web tool. The land-use data is saved using the Overpass API. This

is integrated in a python script using the OverPy library and sends queries to the overpass server in order to obtain data points from OSM (PhiBo, 2021). In Overpass API, a linestring geometry can be queried by providing the code with the longitude and latitude of the all the points in the sequence. However, providing the API with all datapoints for all tracks puts a large burden on the server and can take a long time to process. Using all points from the tracks even causes server overloads. To mitigate this problem, the number of points in the track is reduced.

To compose a line with a similar curve, but with fewer points, the Ramer-Douglas-Peucker (RDP) algorithms is used. The algorithm constructs lines between sets of points. If the intermediate points between these points are within a given tolerance (ε) from the line, the intermediate points are removed, and the constructed line becomes the new line segment (Karthaus, 2012). The epsilon is a distance defined in degree, minute, and second. In Figure 3.7 an example is given of a simplified track. This track is a recording of a bicycle delivery service employee. Even though the track has a high curvature, the RPD algorithm successfully decreases the number of points with 98.6% from 8023 to 109 (epsilon: 0.001). The higher the tolerance, the more simplified the track will become. The tolerance used in the example is for demonstration purposes. The tolerance that is used in this research is obtained through trial and error and is found to be 0.0001. It is important to note that the ideal tolerance was obtained by using a track that is close to the latitude of Greenwich. The tolerance unit is in degree, minute, and seconds, so the tolerance in meters will be bigger at higher latitudes and smaller at lower latitudes. The coordinates of the RDP simplified track are used to query all waterways, foot-paths and cycle-paths in a proximity of 50 meters from the track. To not overload the Overpass server and to keep the runtime manageable, not every single point in the GPX track is used. Instead, 20 evenly distributed points are taken from each track and used to calculate the distances between them and the queried nodes from OSM. The queried results are used to calculate land use features. This process is described in the next paragraph.

Figure 3.7 GPX track before (left) and after (right) applicating the RDP algorithm



3.2 Feature calculation

The selection of the features that are used in the machine learning algorithms highly influence the success of the model. In the theoretical framework, a survey is done on past HAR research and the features that are used (appendix A). Based on popular features in past research, features used for this research are selected. Table 3.1 gives an overview of the features used in this research and how they are constructed and using what data. To calculate the values of the features, temporary lists (or dynamic arrays) are made in a python script. The section below describes all the temporary lists that are made and how they are calculated.

Using gpxpy, certain lists of variables are extracted from the raw GPX data files. These values include $\vec{t}$ for datetime, which is a list of times of all observations in a file in the format 'year, month, day, hour, minute, second'. It includes the variables $\overrightarrow{lat}$ and $\overrightarrow{lon}$, which is a list of latitude and longitude values respectively for every observation in a file in radians. And lastly it includes the variable $\vec{e}$ for elevation, which is a list of elevation values of all observations in a file in meters. From these lists of variables, all other variables are calculated using the formulas described below. In these formulas, subscript 'i' refers to the ith observation from the corresponding list. 'n' refers to the number of observations in the list.

3.2.1 Time-delta

$\overrightarrow{\Delta t}$ is the time-delta variable, which is a list of time between one observation and the observation before. It is calculated in seconds

$$(1)\ \Delta t_i = t_i - t_{i-1}$$

Where $\vec{t}$ is the datetime list.

3.2.2 Distance

$\vec{d}$ is the distance variable, which is a list of distance between one observation and the observation before in meters. In Strava, only the horizontal distance is calculated, not considering elevation gain (Strava Support, 2021). Therefore, this research does the same. There are many ways to calculate the distance between coordinates. The Geopy module in python uses the geodesic distance to calculate the distance (Geopy, 2018) and is also used in this research. It makes use of the Haversine formula, which works as follows (MTL, n.d.):

$$(2)\ d_i = R * c_i$$

Where $\vec{d}$ is the distance in m, R is the radius of the earth in m, $\vec{c}$ is the central angle between two points on a sphere in radials.

$$(3)\ R\ = 6{,}373 * 10^6$$

$$(4)\ c_i = 2 *\ atan2(\sqrt{a_i}, \sqrt{1 - a_i})$$

Where atan2 is a function that takes arguments y and x and computes the arc tangent of the ratio x/y. Haversine's formula uses $\sqrt{a}$ and $\sqrt{1-a}$ as arguments.

$$(5)\ a_i = \sin(\frac{lat_i - lat_{i-1}}{2})^2 + \cos(lat_{i-1}) * \cos(lat_i) * \sin(\frac{lon_i - lon_{i-1}}{2})^2$$

Where $\overrightarrow{lat}$ and $\overrightarrow{lon}$ are the latitude and longitude in radians.

3.2.3 Elevation change

$\overrightarrow{\Delta e}$ is the elevation change, which is a list of elevation changes in the GPX at each observation and the observations before in meters. For calculating this value, 25 observations before are used instead of the last observation. Then the change between these observations is divided by 25 to estimate the elevation difference between one observation and the last. This is done to improve estimation accuracy, as the elevation change is only measured with increments of 0.1 meters. Due to these large increments, the total elevation change can appear much larger than is the case. The results from the calculations done in this research are compared to the estimations of Strava and were found to be sufficiently comparable as opposed to calculating one observation at a time. Note that due to this, $\Delta e_1, \Delta e_2 \ldots \Delta e_{25} = 0$.

$$(6)\Delta e = \frac{e_i - e_{i-25}}{25}$$

Where $\vec{e}$ is the elevation in m.

3.2.4 Velocity

$\vec{v}$ is the velocity variable, which is a list of velocities at each observation in the GPX file in meter per second. Note that for the first observation $v_1 = 0$.

$$(7)\ v_i = \frac{d_i}{\Delta t_i}$$

Where $\vec{d}$ is the distance in m, and $\overrightarrow{\Delta t}$ is the time-delta in s.

3.2.5 Acceleration

$\vec{a}$ is the acceleration variable, which is a list of accelerations at each observation in the GPX file in meter per second squared. Note that for the first and second observation $a_1 = 0,\ a_2 = 0$.

$$(8)\ a_i = \frac{v_i - v_{i-1}}{\Delta t_i}$$

Where $\vec{v}$ is the velocity in m/s, and $\overrightarrow{\Delta t}$ is the time-delta in s.

3.2.6 Relative bearing

$\vec{\theta}$ is the relative bearing variable, which is a list of relative bearings between each observation and the observation before in degrees. The average bearing change is calculated regardless of the direction of the bearing change. Therefore, the absolute number of bearing change will be saved to the bearing change list. Note that for the first observation $\theta_1 = 0$.

$$(9)\ \theta_i = |atan2(y_i, x_i) * \frac{180}{\pi}|$$

Where atan2 is a function that takes arguments $\vec{y}$ and $\vec{x}$ and computes the arc tangent of the ratio x/y.

$$(10)\ y_i = \sin(lon_i - lon_{i-1}) * \cos(lat_i)$$

$$(11)\ x_i = \cos(lat_{i-1}) * \sin(lat_i) - \sin(lat_{i-1}) * \cos(lat_i * \cos(lon_i - lon_{i-1})$$

Where $\overrightarrow{lat}$ and $\overrightarrow{lon}$ are the latitude list and longitude list of observations in radians.

3.2.7 Proximity to water, cycle-path, and footpath

$\overrightarrow{Pwater}, \overrightarrow{Pcycle}$ and $\overrightarrow{Pfoot}$ are the variables for the distance of every 100th observation in the GPX track to the closest waterway (for $\overrightarrow{Pwater}$), cycle-path (for $\overrightarrow{Pcycle}$) and footpath (for $\overrightarrow{Pfoot}$) in meters. Proximity is chosen as a variable, as it describes the relationship between the points and the proximity best. It provides more detailed information then making proximity classes or even deciding on a binary classification for being close to the land-use feature or not. Only 20 points are selected for each track, so the influence of each point on the track has a large influence on the feature values. Only 20 evenly distributed points are taken from the track to decrease the processing power needed as opposed to taking every single point of the track. This does however mean that the lists for these variables contain far less values than all the other lists mentioned above. Therefore, instead of using 'i' to refer to the elements in the list, 'j' will be used. Similarly, the number of spatial elements found have no relation to the number of values in the original track, therefore, 'k' is used to refer to the list of spatial elements. For notational convenience, calculating the distance is written as a function distance (). This function is identical to how the variable $\vec{d}$ is calculated.

$$(12)\ Pwater_j = \min_k(distance(SimpTrack_j, waterway_k))$$

$$(13)\ Pcycle_j = \min_k(distance(SimpTrack_j, cyclepath_k))$$

$$(14)\ Pfoot_j = \min_k(distance(SimpTrack_j, footpath_k))$$

Where $\overrightarrow{SimpTrack}$ refers to the lists of longitudes and latitudes of 20 evenly distributed points in radians on the original track and $\overrightarrow{waterway}$, $\overrightarrow{cyclepath}$, and $\overrightarrow{footpath}$ refer to the lists of longitudes and latitudes of the spatial elements that are within a 10 meter buffer of the RDP simplified track.

A drawback of only selecting features that are within a buffer, is that the outcome might not reflect the real-life situation. For example, a waterway could be 100m from the track, but never be identified, as it is outside of the buffer range. If there are no waterways, foot paths or cycle paths in the proximity in the entire track, a missing data value is assigned that can be easily identified. In this case, 10,000 is assigned as the identifier for all missing values as this will be more than all other values that are within the buffer range. Later in the pre-processing phase, this value is changed to suit the range of the data values per feature (Machine Learning Knowledge, 2018).

3.2.8 Feature list

Table 3.1 contains a list of the features used in the machine learning model, a description of the feature and the formula used to calculate the feature for each track. The features are based on the features used in past HAR research which can be found in appendix A. Most of the features selected for this research are based on features selected by Ferri (2016). This research has similarities in the data used and classification methods used. Besides these, some more features are added, including the 95th percentile speed, 95th percentile acceleration and heading change as in Xiao, Cheng & Zhang (2019). Using only the 95th percentile of speed and acceleration helps remove the biggest outliers in the data, assuming some GPS measurement inaccuracies. The low-speed point is set at everything lower or equal to 0.5 km/h.

All these features are calculated using a python script and added to a CSV file. Besides the features in table 3.1 that are used to train and test the model, the CSV file also contains a column 'type of activity performed' in the corresponding track, so that supervised learning can be applied to the machine learning model. Other columns that can be found in the CSV file but are not used for the machine learning models are a unique ID for every track and an average Δt and are therefore not mentioned in table 3.1.

Table 3.2 Features used for the machine learning model

Features	Description	Formula
Total distance	Total distance of the route (m)	$L = \sum_{i=1}^{n} d_i$
Total time	Total time of the track excluding the low-speed point rate time (seconds)	$t_{tot=} \Delta t_n - \Delta t_1$
Low-speed point rate per second	Total time spent stationary or without any significant movement per second (points/second). Stationary of without any significant movement is 0.5 km/h or 0.14 m/s or slower.	$Update\ \overrightarrow{\Delta t}\ such\ that\ \Delta t_i := 0\ where\ v_i > 0.14$ $v_{low-speed} = \sum_{i=1}^{n} \Delta t_i$ $v_{low-speed/s} = \frac{v_{low-speed}}{t_{tot}}$
Average speed	Average speed of the track (m/s)	$\bar{v} = \frac{1}{n}\sum_{i=1}^{n} v_i$
Average acceleration	Average acceleration of the track (m/s^2)	$\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i$
95th percentile of speed	The average speed in the 95th percentile (km/h)	$Sort(\vec{v})\ such\ that\ v_1 \le v_2 \le v_3$ $v_{95th} = v_i \rightarrow [i] = n * 0.95$
95th percentile of acceleration	The average acceleration in the 95th percentile (m/s^2)	$Sort(\vec{a})such\ that\ a_1 \le a_2 \le a_3$ $a_{95th} = a_i \rightarrow [i] = n * 0.95$
Elevation gain	Total elevation gain (m)	$Update(\overrightarrow{\Delta e})\ such\ that\ \Delta e_i < 0 := 0$ $E_{gain} = \sum_{i=1}^{n} \Delta e_i$
Elevation loss	Total elevation loss (m)	$Update(\overrightarrow{\Delta e})\ such\ that\ \Delta e_i > 0 := 0$ $E_{loss} = \sum_{i=1}^{n} \Delta e_i$
Elevation gains relative	Average elevation gains per km travelled (m/km)	$Update(\overrightarrow{\Delta e})\ such\ that\ \Delta e_i < 0 := 0$ $E_{gain} = \frac{1}{1000L}\sum_{i=1}^{n} \Delta e_i$
Elevation loss relative	Average elevation loss per km travelled (m/km)	$Update(\overrightarrow{\Delta e})\ such\ that\ \Delta e_i > 0 := 0$ $E_{loss} = \frac{1}{1000L}\sum_{i=1}^{n} \Delta e_i$

Maximum Elevation	Maximum elevation of the track (m)	$e_{max} = \max \vec{e}$
Minimum Elevation	Minimum elevation of the track (m)	$e_{min} = \min \vec{e}$
Heading change relative	Average heading change per km travelled (degrees/km)	$\bar{\theta} = \frac{1}{n}\sum_{i=1}^{n} \theta_i$
Average distance to water	The average distance from a point in the route to the closest waterway	$\overline{Pwater} = \frac{1}{n}\sum_{j=1}^{n} Pwater_j$
Average distance to cycle path	The average distance from a point in the route to the closest cycle path	$\overline{Pcycle} = \frac{1}{n}\sum_{j=1}^{n} Pcycle_j$
Average distance to foot path	The average distance from a point in the route to the closest foot path	$\overline{Pfoot} = \frac{1}{n}\sum_{j=1}^{n} Pfoot_j$

Data filtering, editing and normalization

When all features are calculated, some data filtering, editing and normalization is needed to deliver the best results from the machine learning model. Not all data is suitable for building the model. First, the track should be at least 2 minutes in length, as this is suggested by Martin et al. (2017). If a track is only 2 minutes long, the data will only have about 120 datapoints and is therefore sensitive to outliers. Often tracks of 2 minutes or shorter are recorded by mistake or are short because of GPS problems. Second, duplicate tracks will be removed. Duplicate tracks can occur when athletes are tagged in the segments of other athletes, when the Strava scraper happened to randomly load the same track twice, or when one athlete shared their activity in multiple groups.

After filtering the data, some features are edited. In the last paragraph, the calculation of average proximity to water, foot-paths and cycle-paths are covered. When the proximity of water, foot-path or cycle-paths are not inside the buffer range, a value of 10,000 is gives to the track as an identifier of a missing value. Since the machine learning algorithm only handles features with continuous data, the missing value is required to be a number (Machine Learning Knowledge, 2018). However, the value of 10,000 is large compared to most other values and causes some problems when the data is normalized. Therefore, the assigned value should be closer to the maximum values of the proximities within the buffer range. A boxplot maximum of these values is used to replace the assigned values of 10,000. For this research, this is 1346.74 meters for waterways, 386.44 meters for footpaths and 1340.35 meters for cycle-paths.

When using machine learning models, it is important to normalise the data. Otherwise, the dimensions of the data can vary greatly, affecting the algorithms capabilities to efficiently classify data. A usual range for normalisation is considered to be either 0 to 1 or -1 to 1 (Kumar, 2024). Since the data in this research does not have any negative values, all values are normalised to be between 0 and 1.

3.3 Machine learning algorithms

In this paragraph, the machine learning algorithms that are used in this research are discussed. Then the usage of the algorithms in R are discussed. Finally, the calibration of the models is discussed and what output it delivers.

There are many different machine learning algorithms that can be used for the application, literature on the mode and activity detection showed that not all algorithms are evenly suitable for the situation. Therefore, only the most suitable algorithms are used. From the literature review it became apparent that the random forest (RF) algorithm was the superior one for multi-class classification problems (Ferri, 2016; Ellis et al., 2014; Lee & Kwan, 2018; Martin et al., 2017; Feng & Timmerman, 2015). Other algorithms used in past research that yielded promising results were Support Vector Machines (SVM) and Artificial Neural Networks (ANN) in the form of Multilayer Perceptron (MLP) (Xiao, Cheng and Zhang, 2019; Li et al., 2020). For that reason, this research focusses on comparing the results of these three algorithms.

The machine learning models are programmed, trained, validated, tested, and calibrated using the R programming language. R is a software environment that is used for statistical computing. R has various packages dedicated to specific machine learning techniques that can be downloaded.

3.3. 1 Testing, training, and validation

The ideal ratio of training data to testing data is 90% to 10%. However, the overall increase of accuracy between a 50% to 50% ratio and a 90% to 10% ratio is only about 0,5% (Shafique & Hato, 2016). But besides accuracy, there are other reasons to consider which ratio to use. The ratio also determines method of cross validation (CV). Using a 50% to 50% ratio, only a 2-fold cross validation is possible, where a 90% to 10% ratio allows for 10-fold cross validation. Cross validation splits the data in training- and testing-data, and changes the samples that are used for training and testing each fold, until all data is used for testing the model. The accuracy of the model is then calculated by averaging each iteration's accuracy measurement (Yang et al., 2021; Wu, Yang & Jing, 2016). A higher number of folds generally leads to a lower prediction error (Olsen, 2021). Therefore, in this research, a ratio of 90% training data and 10% testing data is used.

To evaluate the models, the F1-score is calculated for each class. The F1 score measures the fit of a model on a particular dataset and is calculated using the precision and recall of the model. The precision is the number of true positives, divided by the total number of positives. In other words, the number of times the prediction was right, divided by all predictions done on data belonging to that class. The recall is the number of true positives, divided by the amount true positives and false negatives. In other words, the number of times the prediction was right, divided by all data that was predicted to be in the corresponding class. F1-scores are typically used for binary classifications but can also be used for multi-class classification. In this case, a one-vs.-all approach is taken to find out the F-scores for each activity type in the models. The F1-score is a value between 0 and 1, in which 1 is a perfect fit. The F-score gives is more robust, compared to accuracy, as it also considers the unbalanced distribution of class sizes (Wood, n.d.).

$$(15)\ P = \frac{TP}{TP + FP}$$

Where P is the precision, TP is the number of true positives and FP is the number of false positives.

$$(16)\ R = \frac{TP}{TP + FN}$$

Where R is the recall, TP is the number of true positives and FN is the amount of false negatives

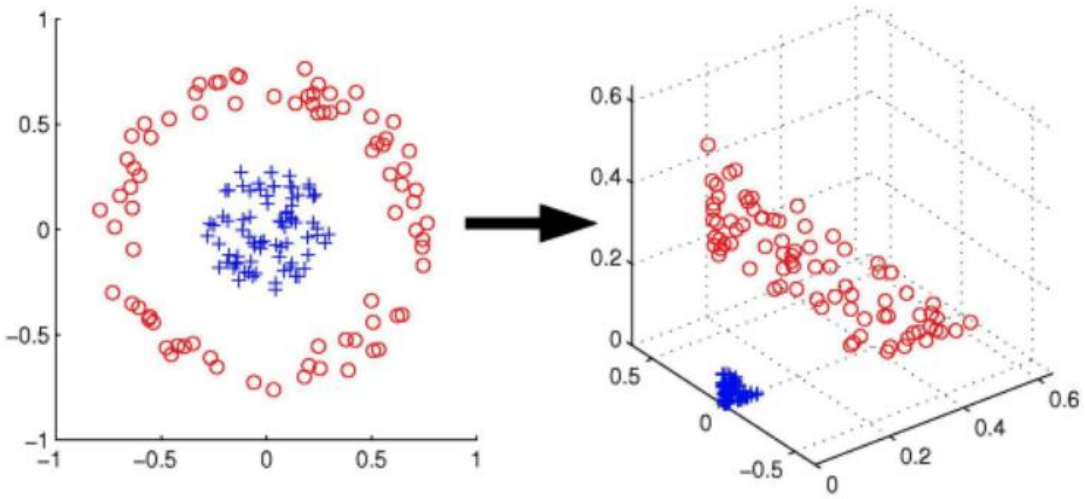
$$(17)\ F = \frac{2 * P * R}{P + R}$$

Where F is the F1-score, P is the precision and R is the Recall

3.3.2 Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a supervised machine learning algorithm that uses a line, plane, or hyperplane through a dataset to separate it into clusters. The algorithms make use of support vectors, these are points of data that influence the position and orientation of the line or hyperplane. The support vectors that are used are the data points on the edge or within the soft margin. The soft margin is the margin from the line or hyperplane in which the number of observations compared to the number of misclassifications is the highest. Of course, not all datasets can be easily separated using a line, plane, or hyperplane. To be able to do this, the SVM uses so called 'kernel' functions. These functions can add additional dimensions to the data, to be able to separate the clusters. In essence, using the kernel functions in SVM is a mathematical trick that allows SVM to perform 'multi-dimensional' classification on a set of one-dimensional data (Noble, 2006). Figure 3.8 shows data that cannot be easily separated by a line, but after transforming it using a kernel function, the clusters are separable by hyperplanes (Bedell, 2018). SVM's are often used in HAR (Ferrari et al., 2021). Support Vector Machines are used for classifying two classes. In the case of multiclass classification, a one against one or a one against rest approach needs to be taken. This means that the data is partitioned into subsets with each containing two classes. In one against one this is one class versus another class, while in the one against rest approach this is one class versus the rest of the classes. The binary classification is solved for each subset and using majority voting, the classification is assigned to each data point (Baeldung, 2021).

Figure 3.8 Using a kernel operation to make clusters separable (Bedell, 2018)



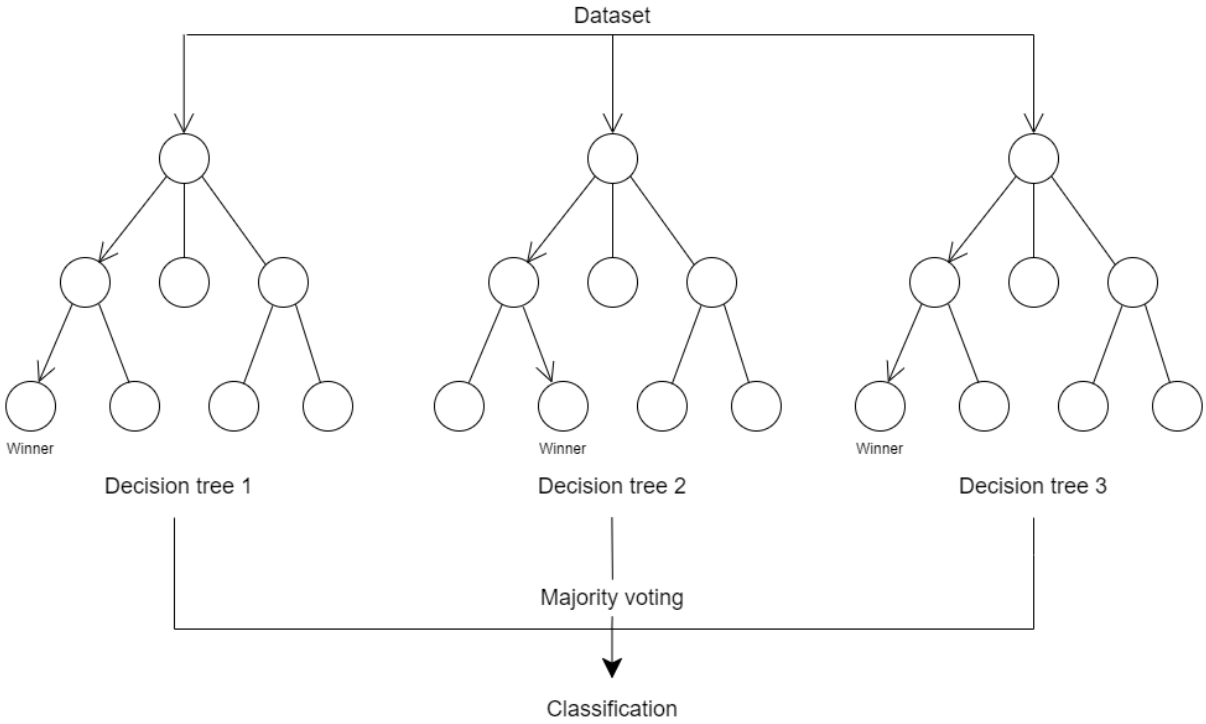
In R, the SVM model can be built using a library called 'e1071'. In this library, the SVM model can make use of four different kernel type. These are: (1) 'linear', (2) 'polynomial', (3) 'radial basis' and (4) 'sigmoid' kernels. In the training phase, the different types of kernels are tested to see which one is best suited to this specific problem (RDocumentation, 2021a). Cost and gamma hyperparameters are used to finetune the model. In which cost is the cost of a constraint validation, and the gamma parameter influences the influence of the selected support vectors. A high cost usually means less support vectors are selected, and a low gamma makes the influence of the selected support vectors high (Scikit-Learn, n.d.). The library also has an argument for cross validation. As stated before, 10-fold cross validation is used for this model with 10% testing- and 90% training-data.

3.3.3 Random Forest (RF)

The Random Forest is based on the tree-based algorithm of Decision Trees (DT's) (Martin et al., 2017). A decision tree has a lot of resemblance to flow charts. DT's split data into groups, based on certain variable thresholds into groups. These groups can then be split into more groups based on other thresholds of different variables until a leaf node is reached in which the classification is decided. DTs are simple to implement, interpret and require low computational power. This simplicity also causes the algorithm to be less suited for more complex datasets and highly susceptible to change when the thresholds are changed (Kingsford & Salzberg, 2008).

The Random Forest (RF) is a machine learning algorithm that uses an ensemble learning method. This means that the algorithm generates many classifications and then aggregates them to come with a final classification. To get these classifications, multiple DTs are generated. These DTs are generated by randomly selecting data samples and a feature subset on which a 'random' DT is generated. This process is called 'Bagging' (a combination of bootstrapping and aggregating). The data that was is not sampled for building the DTs are called the out-of-bag samples. All of these 'random' DTs generate their own classification based on the training data. The test data is classified based upon the most common classification among all the random DT's (Figure 3.9) (Ali, Khan, Ahmad & Maqsood, 2012). The RF is generally able to achieve high performance with high-dimensional data by increasing the number of DT's (Ferrari et al., 2021).

Figure 3.9 Schematic representation of RF algorithm



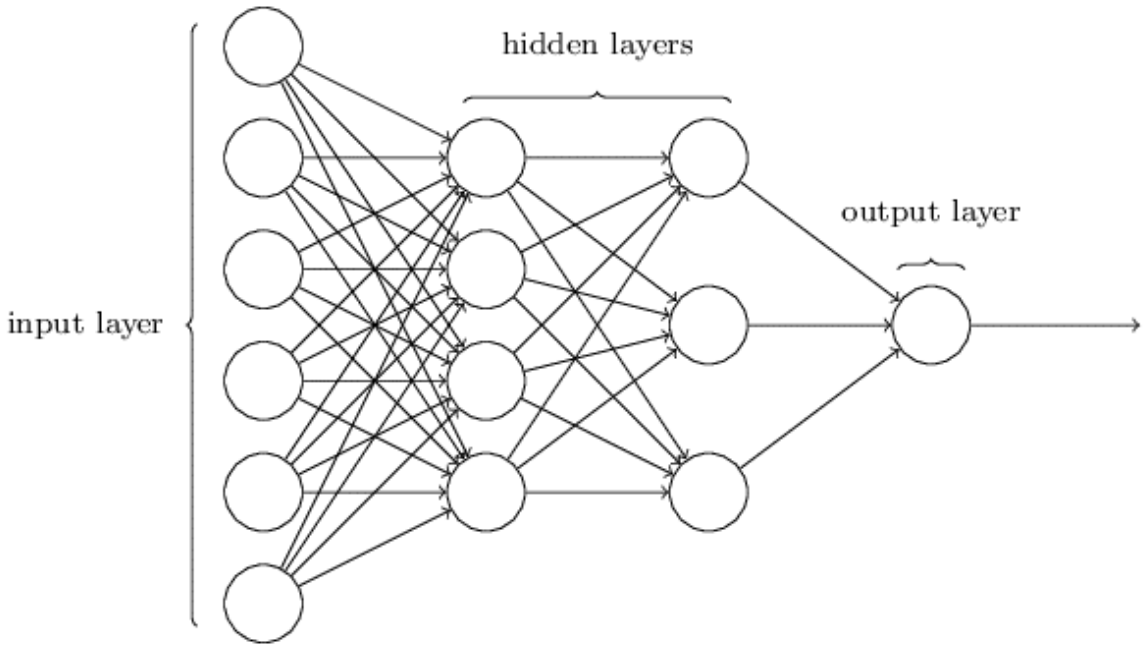
In R, the RF model is built using a library called 'randomForest'. It uses Breiman's implementation of the RF algorithm (Breiman, 2001). There are various hyperparameters that can be tweaked that will influence the accuracy of the model. Three of these hyperparameters will be tweaked in this research. First of which is the number of trees. A higher number of trees usually makes for a better classification. This can be in the range of thousands of trees. However, using to many trees can also cause overfitting of the model. Second, the number of random features used for each decision tree (Towards Data Science, 2020). The number of variables that will be randomly sampled is said to yield the best result when it is around the square-root of the number of features used in classification problems (Breiman,

2002). To find the best value, a grid analysis can be performed. A value that is too small will reduce the variance, but increase the bias, while a value that is too high will do the opposite. A high value also decreases the training speed (Towards Data Science, 2020). Finally, the maximum number of terminal nodes. This parameter is not prone to overfitting, but a higher number does not necessarily make a better model. Testing various values can decide the best value to use (Scornet, 2017). The RF model output includes an out-of-bag error rate, the confusion matrix, and the classification accuracy. The RF model's is validated using out-of-bag samples, therefore, the model does not split its data into subsamples of test- and training data. The validation of the accuracy is done by running the model 10-fold and taking the average accuracy, instead of doing a 10-fold cross validation as is done in the SVM and MLP models.

3.3.4 Multilayer Perceptron (MLP)

Artificial Neural Networks (ANN) often simply called Neural Networks are a collection of machine learning algorithms that are based on the biological neural networks of brains. Brains have neurons, a lot of them, that interact with each other. In real life, neurons transmit chemicals (neurotransmitters) to the next group of neurons to send messages to each other. In the same way, the ANN's uses 'neurons' that can influence one another by sending messages that are influenced by set weights to find patterns in data on which it can classify (Alpaydin, 2014b; Noriega, 2005). There are many ANN's, but the most classic example of ANN is the Multilayer Perceptron (MLP) algorithm. The input data is given in the form of hand-crafted features, the data will feed through one or more sets of neurons called the 'hidden layers' and will produce an output classification (Figure 3.10). The data feeds forward through the neurons and are influenced by the weights (parameters) given to the connections. The parameters are assigned to minimize the cost function, which means finding the right parameters to minimize the error of a line or hyperplane that categorize the data. After feeding the data forward, the model evaluates the predicted values to the actual values. The difference between this is displayed in an error term, and this error term will be backwards propagated from the output node, through the hidden layer to the input node. This is done so the model knows how to adjust the weights to decrease the error term, and thus achieve higher accuracy (Noriega, 2005).

Figure 3.10 Neurons and layers of a MLP (Rodriguez, 2020)



Other popular algorithms are deep learning based Convolutional Neural Network (CNN) and Recurrent Neural Networks (RNN). CNNs were designed for automatic image recognition and RNN's are mostly designed to work with sequence prediction problems (Brownlee, 2019). Although very popular ANN's, they will not be discussed here, as they are not as relevant for the field of HAR.

In R, the MLP model is built using a library called 'RSNNS'. RSNNS is an R library based on the Stuttgart Neural Network Simulator, which used to be a neural networks simulator for the University of Stuttgart. By default, it uses error backpropagation to train the model (Zell, 1998). Hyperparameters are set for the MLP are the number of hidden layers and the number of neurons per hidden layer and the number of iterations for learning (RDocumentation, 2021b). This model uses two hidden layers, as two layers are also capable of solving non-linear categorization problems. The number of neurons per layer are an important hyperparameter, too few neurons causes underfitting, while too many causes overfitting (Gad, 2018). The ideal number of neurons per hidden layer can be found using a formula defined by Stathakis (2009). The formula is:

$$(18)\ h_n = 2\sqrt{(m+2)N}$$

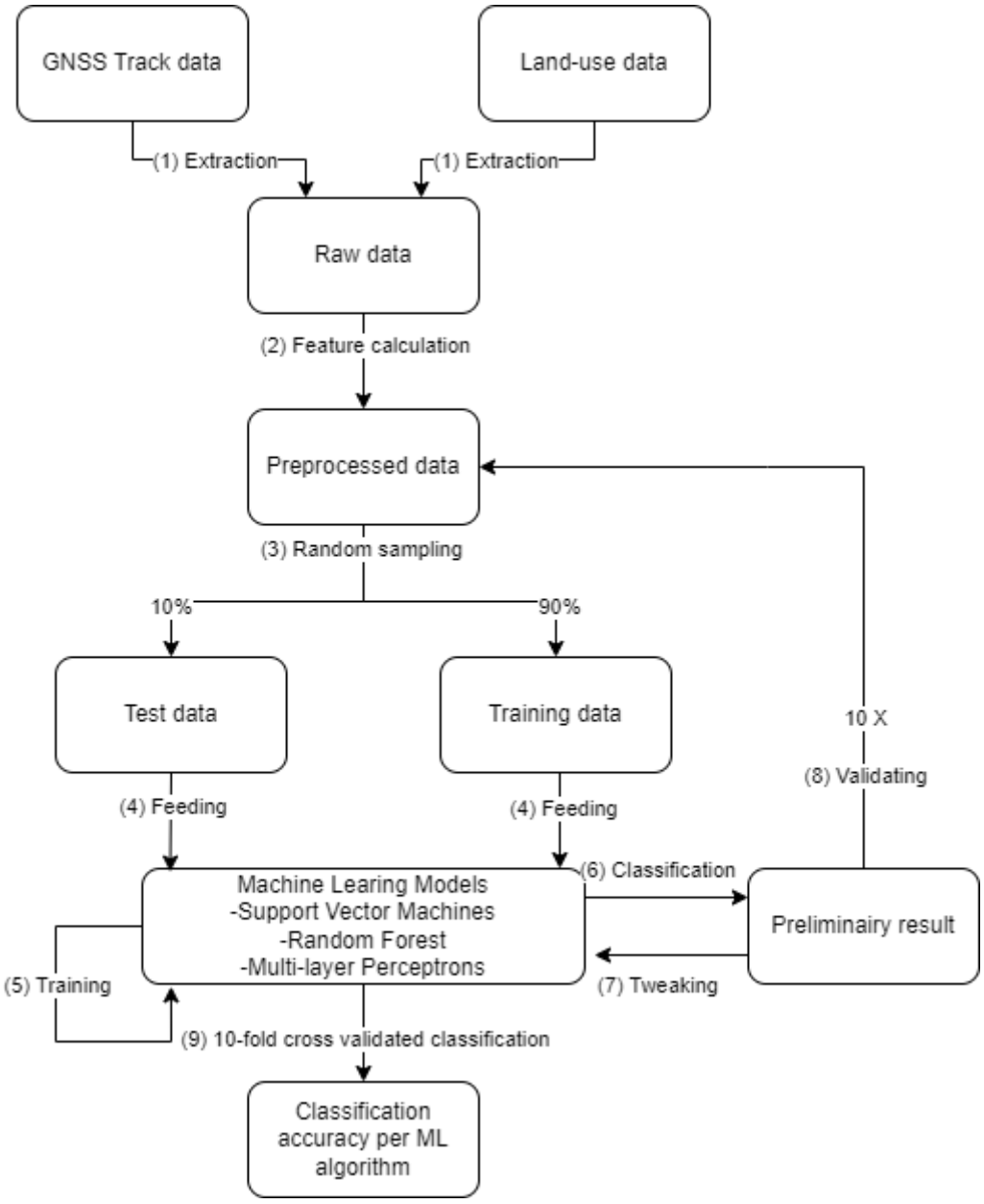
Where $h_n$is the number of neurons per hidden layer, m is the number of output neurons and N is the number of samples.

The number of iterations for learning is another important hyperparameter. Too little will cause a poor model performance, while too many will cause overfitting. Trial and error will be used to find the value that trains the best model. The starting weights of the MLP model are randomized, and the update function follows the topological order. The model is validated using 10-fold cross validation with 10% testing- and 90% training-data.

3.5 Workflow

Figure 3.11 gives a conceptual overview of the steps taken to build the various machine learning sport activity classification models with the GNSS and land-use data.

Figure 3.11 Conceptual model for machine learning model building



4. Results

Three machine learning methods are tested in this research. Each model is trained using specific parameter values. These values are obtained by a combination of trial and error, hyperparameter analyses and recommendations from previous research. For each machine learning method, two models were trained and tested. One with all features that are discussed in the methodology chapter, and one without the land-use specific features. Each model yields at least a confusion matrix, a classification accuracy for each model, and a precision, recall and F1-score for each class in the models. The upcoming chapter discusses the results from the models, as well as some specific hyperparameters for each machine learning method. The strong points, weak points, and overall suitability for all models regarding the classification task at hand will be discussed. Finally, a leave-one-out accuracy analysis is performed for each machine learning method to gain a deeper understanding on the influenced of the features that are used.

4.1 Data overview

After filtering and editing, 1397 GNSS tracks are left. Each track has 17 different features, of which 14 are calculated using the GNSS data and 3 are calculated using OSM land-use data. Table 4.1 shows the values of the 'average distance to water' for all the different sport activities. In Figure 3.6, the expected relationship between land-use type and sport activity is displayed. Comparing this to the output it becomes apparent that sports associated with foot-paths, like walking, running, hiking, and inline-skating, are nearer to this land-use type than average, while all others, except kayaking, are further from the average. Sports related to water, like Ice-skating, kayaking, and swimming are also closer to water-features than average. All other sports are further from the average. Finally, sports expected to be related to cycle-paths, like cycling or inline-skating were further away from the average, while sports like kayaking, ice-skating, and swimming were closer to the cycle-paths than average. While this is not in line with the expectations, the outcome can still positively influence the accuracy of the machine learning algorithm. Therefore, it is still used as a feature for some of the models. An overview of all the histograms, related to the features can be found in Appendix D.

Table 4.1 Overview of the average values for the land-use features per activity type

Track type	*Average distance to foot-path (m)*	*Average distance to cycle-path (m)*	*Average distance to water (m)*
Running	145	638	619
Cycling	185	712	818
Walking	156	716	701
Hiking	116	529	598
Ice-Skating	188	548	574
Inline-Skating	119	552	606
Kayaking	153	132	214
Swimming	211	191	285
All tracks	158	552	596

4.2 Support Vector Machines

The machine learning models for support vector machines (SVM) are trained and tested using the statistical computing programming language 'R'. Multiple kernels are tested to find the kernel that yields the highest classification accuracy. The possible kernels are 'linear', 'polynomial', 'radial basis' and 'sigmoid'. The radial basis kernel function yielded the best results in the 10-fold cross validated models with an accuracy of 0.75 and thus is used for these models. The linear function was a little less accurate with 0.73. The polynomial- and sigmoid-functions were not suitable for the model as they yielded an accuracy of 0.54 and 0.53 respectively. For the final models, the radial basis function has been used. Since this research deals with a classification problem in which the classes are factors, a c-classification type is used.

The models' cost and gamma hyperparameters are established by using a tuning function in R, in which every gamma value between 0 and 0.5 is tested with increments of 0.01 and every cost value is tested from 1 to 10 with increments of 1. The best hyperparameters for this model are a gamma value of 0.05 and a cost value of 7.

The support vector model was 10-fold cross validated and has a classification accuracy of 0.826. Which means that it was able to predict the right class in 82.6% of the cases. It was built using 1013 support vectors. The confusion matrix of the model can be found in table 4.2. The highest F1-score is achieved in the swimming category with 0.92. The lowest F1-score is found in the inline-skate category with 0.76. GNSS tracks of inline-skating were most often confused with cycling (in 9% of the cases), while at the same time running (8%) and cycling tracks (8%) were mostly misclassified as inline-skating. Other notable misclassifications are kayaking being classified as walking in 11% of the cases. Walking is confused with hiking in 17% of the cases and in 8% of the cases vice versa.

Table 4.2 Confusion matrix of the SVM model

Prediction / *Class*	*Running*	*Cycling*	*Walking*	*Hiking*	*Ice-skating*	*Inline-skating*	*Kayaking*	*Swimming*	*Precision*	*Recall*	*F1-score*
Running	207	10	6	3	9	13	9	1	0.80	0.86	0.83
Cycling	2	225	2	0	4	13	0	0	0.91	0.89	0.90
Walking	19	2	207	27	3	6	12	8	0.73	0.83	0.78
Hiking	2	1	20	122	0	1	0	0	0.84	0.78	0.81
Ice-skating	0	2	1	0	60	3	0	0	0.91	0.71	0.79
Inline-skating	5	12	1	1	8	114	0	0	0.81	0.72	0.76
Kayaking	6	0	4	4	1	7	89	2	0.79	0.79	0.79
Swimming	1	0	8	0	0	1	3	130	0.91	0.92	0.92
Total	242	252	249	157	85	158	113	141			

Another model was ran without the land use features. The gamma and cost values were tuned for the model and were 0.05 and 6 respectively. It was built using 1024 support vectors. The 10-fold cross validated model achieved an accuracy of 77.0%. The confusion matrix of the model can be found in table 4.3. Once again, the highest F-1 score is achieved in the swimming category. However, the lowest

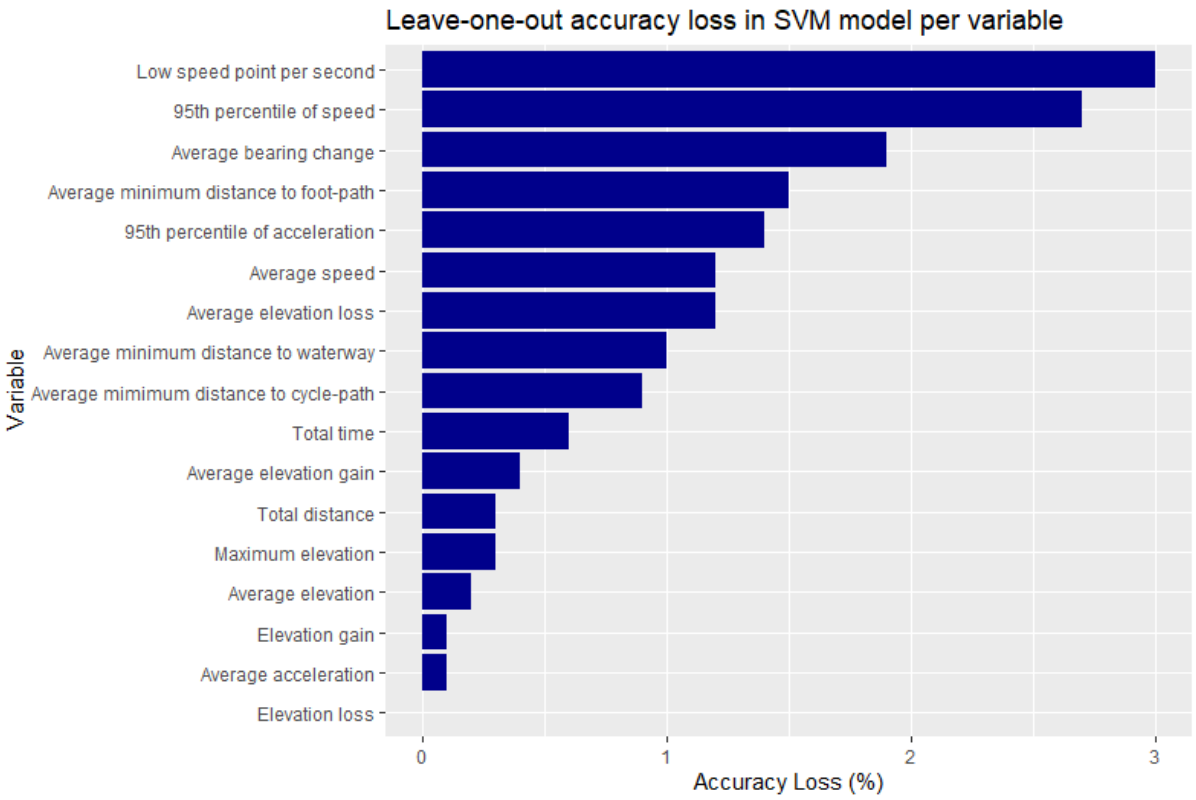
F-1 score in this model is in the category of ice-skating, as opposed to inline-skating in the model using land-use features. The F-1 score dropped from 0.79 to 0.50. Ice-skating tracks are mostly confused with inline-skating tracks. In 27% of the cases, ice-skating tracks are predicted to be inline-skating tracks when land-use data is left out, as opposed to 9% when land-use data is used.

Table 4.3 Confusion matrix of the SVM model without land use features

Prediction / *Class*	*Running*	*Cycling*	*Walking*	*Hiking*	*Ice-skating*	*Inline-skating*	*Kayaking*	*Swimming*	*Precision*	*Recall*	*F1-score*
Running	197	9	6	2	9	13	15	0	0.78	0.81	0.80
Cycling	3	215	1	1	15	17	0	0	0.85	0.85	0.85
Walking	23	2	211	33	5	10	15	10	0.68	0.85	0.76
Hiking	1	0	14	115	0	2	3	1	0.85	0.73	0.78
Ice-skating	0	1	0	0	31	7	0	0	0.79	0.36	0.50
Inline-skating	6	22	2	0	23	105	2	0	0.66	0.66	0.66
Kayaking	11	1	3	6	2	4	76	4	0.71	0.67	0.69
Swimming	1	2	12	0	0	0	2	126	0.88	0.89	0.89
Total	242	252	249	157	85	158	113	141			

To gain insight in the effect of each feature for the accuracy of the model, a 10-fold cross-validated leave-one-out accuracy analysis was done. In which each time, one feature was left out and the accuracy loss was calculated. Figure 4.1 shows that the low-speed point per second was the most influential variable to the overall accuracy. Leaving out elevation loss didn't influence the accuracy of the model up to three decimal places. The features of interest for this research are the land-use features. The 'average distance to foot-path' was the most influential for the SVM model with an accuracy loss of 1.5%. Followed by 'average distance to waterway' with 1.0% and 'average distance to cycle-path' with 0.9%.

Figure 4.1 Leave-one-out-accuracy loss in SVM model per variable



4.3 Random Forest

The random forest (RF) models were built using the 'random forest' and 'caret' packages is R. Random Forest works different for different purposes, but for this research, the model will be built for the purpose of classification. To build the most accurate model, the hyperparameters used for building the random forest need to be tuned. One of the most important hyperparameters is the number of decision trees built per random forest. The more trees, the more robust the model will be. However, more trees cause larger computational time and can cause overfitting. By comparing the number of trees and the out-of-bag (OOB) error rate, the optimal number of trees can be found. After about 200 trees, the OOB error rate stabilises, and the model doesn't improve much (Appendix E). Therefore 200 trees will be used, to prevent overfitting. Another important feature to use is the number of random features to consider at each split. By testing and cross-validating the accuracy of each model using a grid search method, it was determined that using 7 random features is the best parameter value to use for the model (Appendix F). Finally, the maximum number of terminal nodes in the decision trees need to be determined. Testing and cross-validating determined that 70 is the best parameter value to use for this model (Appendix G).

The random forest model is verified by using the out-of-bag error rate instead of splitting the data for a 10-fold cross-validation, like how it is done in SVM- and MLP-models. The model is trained 10-fold using different bootstrapped data and the accuracy is calculated by averaging the 10 models. The average accuracy was 74.6%. A confusion matrix of one of the models can be found in table 4.4. Swimming is the class with the highest F1-score, with a value of 0.91. Inline-skating has the lowest F1-score, with a value of 0.60. Some notable outcomes are the difference in precision and recall in ice-skating, where it yields the second highest overall precision (0.92), but the lowest overall recall (0.53). This means that in 47% of the cases ice-skating tracks were predicted to be some other sport activity, while only in 8% of the cases, another sport activity was incorrectly classified as ice-skating. Another

notable outcome is that hiking and walking are often confused, where about 12% of walking tracks are identified as hiking tracks, and about 27% of hiking tracks are identified as walking tracks. Also 20% of ice-skating tracks were identified as inline-skating tracks but was not misclassified the other way around. Instead, inline-skating is often confused with running (18%) and cycling (28%). Finally, the last combination of categories with more than 10% misclassification is kayaking with walking with 19%.

Table 4.4 Confusion matrix of the RF model

Prediction \ Class	Running	Cycling	Walking	Hiking	Ice-Skating	Inline-Skating	Kayaking	Swimming	Precision	Recall	F1-score
Running	194	6	10	4	9	19	12	1	0.76	0.80	0.78
Cycling	3	205	0	2	6	28	0	0	0.84	0.81	0.83
Walking	18	2	200	42	3	4	22	10	0.66	0.80	0.73
Hiking	5	1	29	107	0	3	4	3	0.70	0.68	0.69
Ice-Skating	0	3	0	0	45	0	1	0	0.92	0.53	0.67
Inline-Skating	16	35	7	0	17	101	0	1	0.57	0.64	0.60
Kayaking	5	0	0	2	5	1	73	2	0.83	0.65	0.73
Swimming	1	0	3	0	0	2	1	124	0.95	0.88	0.91
Total	242	252	249	157	85	158	113	141			

Other models were ran in 10-fold without the land-use features, this average accuracy of these models was 73.9%. This is 0.7% lower than the model that uses all the features. The confusion matrix of this model can be found in table 4.5. The notable outcomes described in the model containing all features remain similar in this model. The most notable change is, while all the values of precision, recall and F1-score are dropping, the precision of walking increased by a value of 0.01.

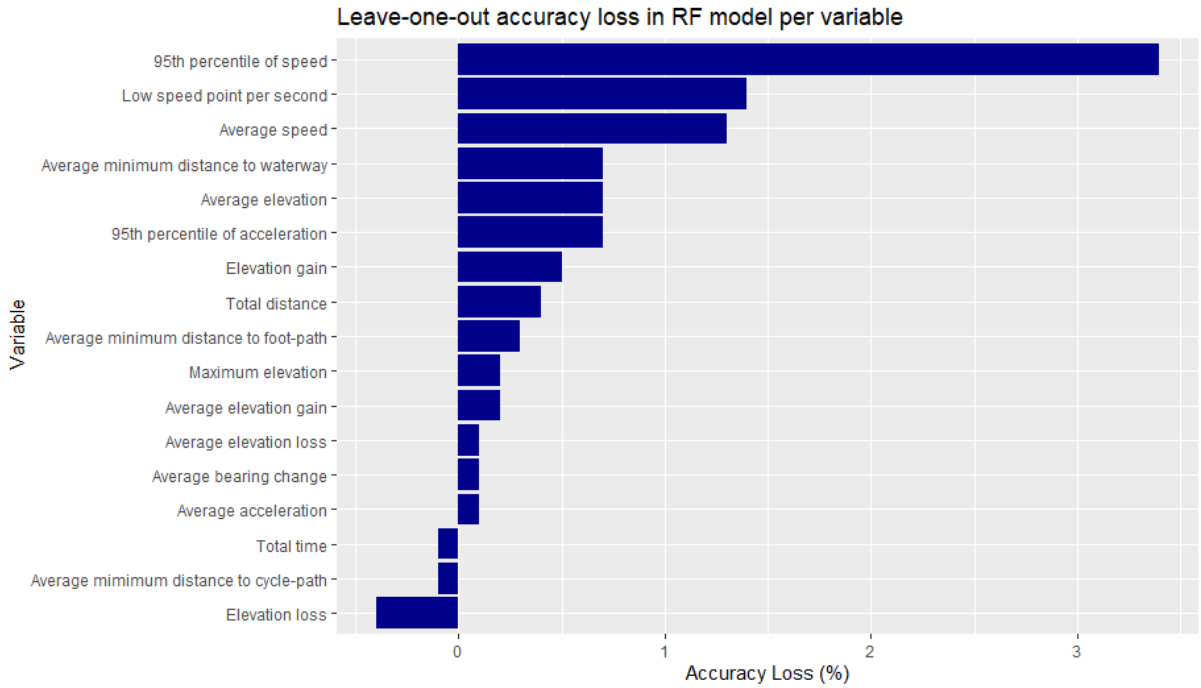
Table 4.5 Confusion matrix of the RF model without land-use features

Prediction \ Class	Running	Cycling	Walking	Hiking	Ice-Skating	Inline-Skating	Kayaking	Swimming	Precision	Recall	F1-score
Running	193	5	10	3	7	14	18	0	0.77	0.80	0.78
Cycling	5	206	0	2	7	29	1	0	0.82	0.82	0.82
Walking	21	1	198	38	5	7	24	10	0.65	0.80	0.72
Hiking	4	0	30	111	0	2	3	3	0.73	0.71	0.72
Ice-Skating	0	4	0	0	45	1	0	0	0.90	0.53	0.67
Inline-Skating	12	35	6	0	19	97	7	1	0.55	0.61	0.58
Kayaking	6	0	1	3	2	6	59	3	0.74	0.52	0.61
Swimming	1	1	4	0	0	2	1	124	0.93	0.88	0.91

Total	242	252	249	157	85	158	113	141

Finally, a leave-one-out accuracy analysis was performed for the model. Figure 4.2 shows that leaving out the '95th percentile of speed' causes the biggest loss in accuracy by far with a 3.4% loss. Leaving out the features 'total time', 'average distance to cycle-path', and 'elevation loss' even caused the model to increase in accuracy. For the last of which it caused an accuracy increase of 0.4%. Leaving out the other land-use features did cause a loss in accuracy. 'Average minimum distance to waterway' caused a 0.7% loss, and 'average minimum distance to foot-path' caused a 0.3% loss.

Figure 4.2 Leave-one-out accuracy loss in RF model per variable



4.4 Multilayer Perceptron

The multilayer perceptron (MLP) models were built using the RSNSS package in R. The learning function used to train the model is the standard backpropagation, as discussed in the methodology chapter. The hyperparameters of the model were tuned to achieve the highest accuracy. The maximum number of iterations was tested through trial-and-error and was set to be 1000 iterations, as this value trained the model with the highest accuracy. The number of nodes in the hidden layer was set to 13. This value was calculated using the formula of Stathakis (2008) as described in the methodology chapter.

The model was 10-fold cross validated and has an accuracy of 69.4%. The confusion matrix can be found in table 4.6 and contains all the test data that was used during the 10-fold cross validation process. Swimming is the category with the highest F1-score with 0.79, while inline-skating is the one with the lowest with 0.49. The outcomes are similar to those of the SVM and random forest, where walking and hiking are often confused. Walking was misclassified as hiking in 13% of the time and 22% vice versa. Ice-skating is often classified as inline-skating, while inline-skating is confused with running (18%) and cycling (16%). Also, kayaking is misclassified as walking in 10% of the cases. In the MLP, as opposed to the other models, swimming is more often confused with walking (in 11% of cases). This was 7% in SVM and 6% in RF.

Table 4.6 Confusion matrix of the MLP model

Prediction / Class	*Running*	*Cycling*	*Walking*	*Hiking*	*Ice-Skating*	*Inline-Skating*	*Kayaking*	*Swimming*	*Precision*	*Recall*	*F1-score*
Running	165	1	21	6	2	12	8	2	0.75	0.75	0.75
Cycling	8	178	3	4	7	19	2	7	0.77	0.78	0.78
Walking	8	1	159	34	1	0	5	17	0.71	0.66	0.68
Hiking	3	1	27	102	0	4	0	6	0.69	0.65	0.67
Ice-Skating	5	5	1	0	44	9	7	3	0.61	0.93	0.74
Inline-Skating	10	43	6	3	13	61	4	1	0.45	0.54	0.49
Kayaking	17	1	17	4	4	5	49	4	0.47	0.62	0.54
Swimming	1	0	6	2	0	1	3	113	0.87	0.73	0.79
Total	217	230	240	155	71	111	78	153			

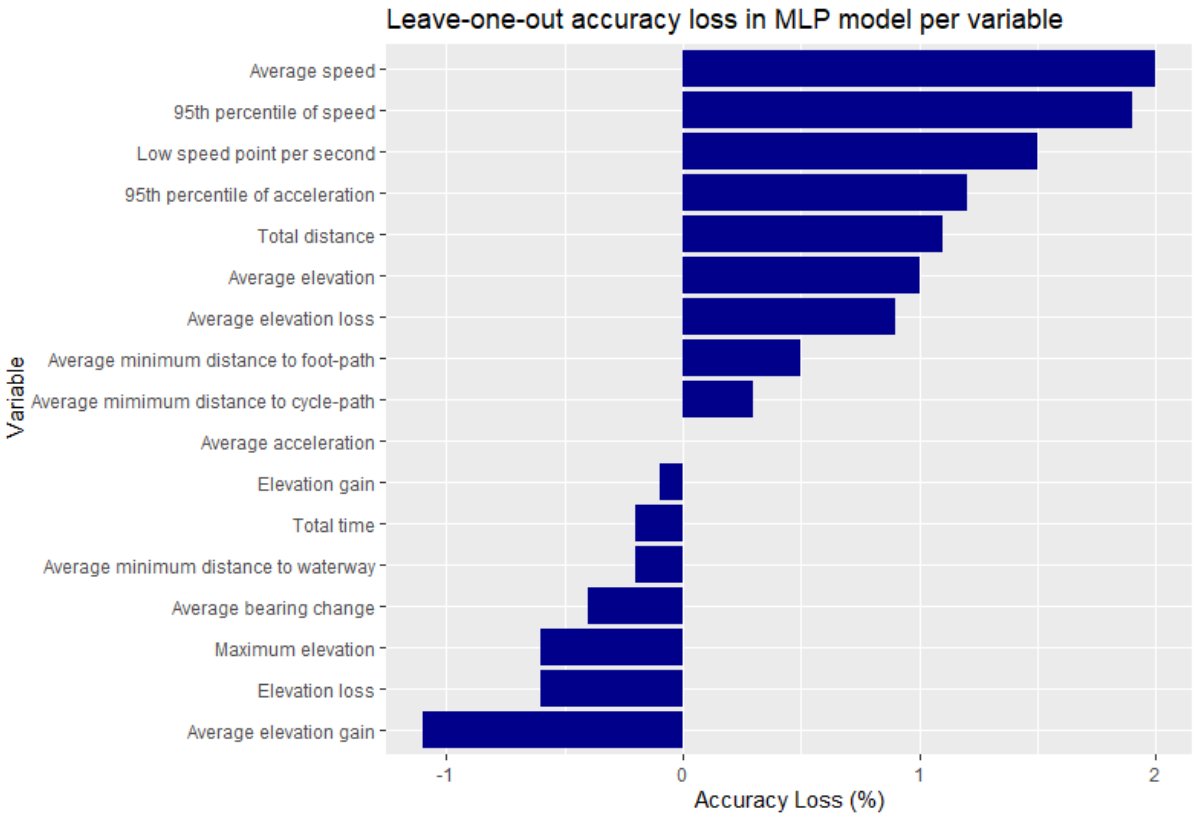
The model was also ran without using the land-use features. The model reached an accuracy of 67.2%, which is 2.2% lower than the MLP model using all features. The confusion matrix can be found in table 4.7. Unlike in the SVM and RF models without land-use features, not all F1-scores were lower compared to the same model with all features. Running and hiking improved the F1-score with 0.01. Most other F1-scores did not change or dropped 0.02, while the F1-score of ice-skating dropped with 0.32 from 0.74 to 0.42. Both the precision and recall dropped significantly. Ice-skating was more often confused with cycling and inline-skating, while at the same time, inline-skating was classified as ice-skating more often.

Table 4.7 Confusion matrix of the MLP model without land-use features

Prediction / *Class*	*Running*	*Cycling*	*Walking*	*Hiking*	*Ice-Skating*	*Inline-Skating*	*Kayaking*	*Swimming*	*Precision*	*Recall*	*F1-score*
Running	185	4	6	1	9	14	24	0	0.76	0.76	0.76
Cycling	5	204	1	0	21	37	2	0	0.81	0.76	0.78
Walking	18	5	183	40	3	8	17	18	0.73	0.63	0.68
Hiking	7	1	29	110	1	2	5	8	0.69	0.67	0.68
Ice-Skating	1	8	0	0	30	18	1	0	0.35	0.52	0.42
Inline-Skating	11	25	3	3	18	65	7	0	0.41	0.49	0.44
Kayaking	15	2	7	4	3	8	55	4	0.48	0.56	0.52
Swimming	1	3	21	1	1	6	3	110	0.79	0.75	0.77
Total	243	252	250	159	86	158	114	140			

A 10-fold cross-validated leave-one-out accuracy analysis was performed for the model. Figure 4.3 shows that leaving out average speed will cause the biggest loss in accuracy with 2.0%. Just like in the SVM and RF models, the '95th percentile of speed' and the 'low speed point per second' are amongst the features that result in the biggest loss of accuracy when left out. Leaving out the land-use features 'average minimum distance to foot-path' and 'average minimum distance to cycle-path' will cause a loss of accuracy of 0.5% and 0.3% respectively. 'Average minimum distance to waterway' caused an increase in accuracy of 0.2%, while this is not the case in the other models.

Figure 4.3 Leave-one-out accuracy loss in MLP model per variable



5. Discussion

In the discussion, the results of all the individual models are placed side by side to reflect on the performances, general trends, and the influence of land use features.

5.1 Model's performance

For this research 6 machine learning models were trained and validated to gain an insight in what the influence is of incorporating land-use data in automatic activity type classification models based on machine learning algorithms. Differences were found in the classification accuracy between the different machine learning algorithms. The models with the highest classification accuracy were the support vector machine models, in which both the model with all features and the model with no land-use features outperformed the other models (Table 5.1). The random forest was the second-best machine learning algorithms for building the classification models. The multilayer perceptron models had the lowest classification accuracy. This is not in line with past research on human activity recognition, in which random forest models often had the highest accuracy (Ellis et al., 2014; Martin et al., 2017; Feng & Timmerman, 2015, Ferri, 2016). This can be due to other research using more- and/or different features.

Table 5.1 Classification accuracies per machine learning model

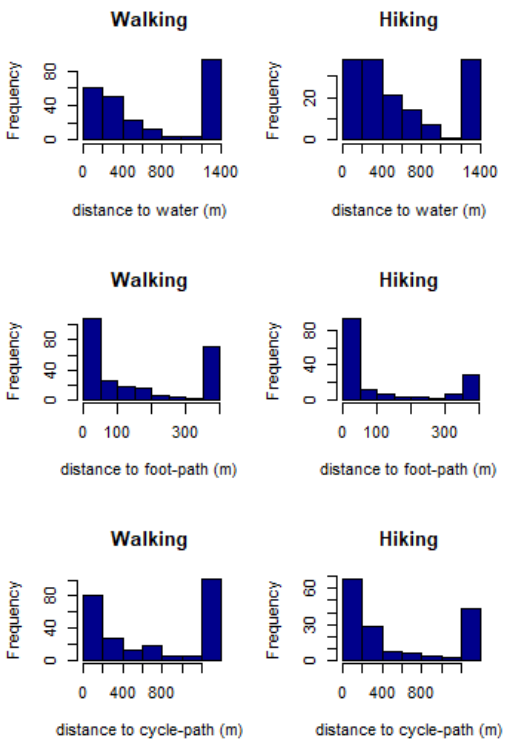
Machine learning model	**Classification accuracy**	
	All features	No land-use features
Support Vector Machines	82.6%	77.0%
Random Forest	74.6%	73.9%
Multilayer Perceptron	69.4%	67.2%

5.2 General trends

A trend among all three of the machine learning algorithms is that leaving out the land-use features from the model caused the models to have a lower overall accuracy. This drop in accuracy was generally reflected throughout every category. The SVM algorithm improved the most when adding land-use features. Its accuracy increased with 5.6%. For the MLP algorithm it increased with 2.2% and for the RF this was only 0.7%.

General trends among all models were the misclassification of specific classes. One of these trends is the confusion between hiking and walking. This can be caused the similarity of the activity. Generally, hikes are considered to be longer and harder walks, but the distinction between the two is somewhat subjectivity. One might consider the performed activity a walk, and labels it accordingly, while someone else considers the same activity a hike. However, since walking and hiking achieved an F1-score between 0.67 and 0.81 in all models, people do tend to make a distinction between the two activity types based on some characteristics. Unfortunately, due to nature of the machine learning algorithms, it is hard to tell what these distinctive characteristics are. Adding land-use features to the models caused the confusion between hiking and walking to go down in all cases, even though it was expected that both activities would be performed on the same land-use type, namely foot-paths. Figure 5.1 shows the distribution of values for all land-use features for walking and hiking. When comparing walking with hiking, it becomes clear that there are differences in the distribution of values between walking and hiking. This is probably what caused the model to get a higher classification accuracy regarding walking and hiking.

Figure 5.1 Histograms of walking and hiking for the land-use features



Another general trend is that in all models except for one, the inline-skating was the category with the lowest F1-score. Going as low as a score of 0.44. Unlike the case with hiking and walking, inline-skating was often confused with more activities than just one. It was often confused with running, cycling, walking, and ice-skating. Adding land-use features to the models increased the F1-score of inline-skating in all three cases. In the support vector machines model it even increased the inline-skating's F1-score from 0.66 to 0.76. In this case, the precision increased from 0.66 to 0.91 as cycling and ice-skating tracks' classification accuracy increased significantly. It seems like the land-use features gives more context to the activities, but again, it is hard to know exactly why this is the case.

The improvements in F1-scores per machine learning model when adding land-use features seem to be very dependent on the algorithm used (table 5.2). Kayaking's F1-score improved greatly in the SVM and RF models, while in the MLP it only improved with 0.02. Ice-skating improved greatly in the SVM and MLP models, while it did not improve in the RF model.

Table 5.2 F1 scores for all classes per model

	Support Vector Machines			**Random Forest**			**Multilayer Perceptron**		
Class	With land-use features	Without land-use features	**Improvement**	With land-use features	Without land-use features	**Improvement**	With land-use features	Without land-use features	**Improvement**
Running	0.83	0.80	**0.03**	0.78	0.78	**0.00**	0.75	0.76	**-0.01**
Cycling	0.90	0.85	**0.05**	0.83	0.82	**0.01**	0.78	0.78	**0.00**

Walking	0.78	0.76	**0.02**	0.73	0.72	**0.01**	0.68	0.68	**0.00**
Hiking	0.81	0.78	**0.03**	0.69	0.72	**-0.03**	0.67	0.68	**-0.01**
Ice-skating	0.79	0.66	**0.13**	0.67	0.67	**0.00**	0.74	0.42	**0.32**
Inline-skating	0.76	0.66	**0.10**	0.60	0.58	**0.02**	0.49	0.44	**0.05**
Kayaking	0.79	0.69	**0.10**	0.73	0.61	**0.12**	0.54	0.52	**0.02**
Swimming	0.92	0.81	**0.03**	0.91	0.91	**0.00**	0.79	0.77	**0.02**

5.3 Leave-one-out analysis

The leave-one-out analysis for the different models gave mixed results. Two features were always among the top 3 features that caused the biggest loss in accuracy. These were the 'low speed point per second' and the '95th percentile of speed'. 'Elevation loss' was the only feature that never caused any loss of accuracy when left out. The land use features performed well in the SVM model, with a 0.9%, 1.0%, and 1.4% increase (table 5.3). When leaving out all land-use features, the model's accuracy drops by 5.6%, meaning that the land-use features also interact with- and positively influence each other in the SVM model. In the RF model and MLP model, not all land-use features caused a loss of accuracy when left out. Leaving out the 'average minimum distance to water' caused the RF model to increase it's accuracy by 0.1%. Leaving out the 'average minimum distance to waterway' caused the MLP model to increase it's accuracy by 0.2%. This shows that the land-use features picked for this research don't work well for all models and it can vary per algorithm that is used.

Table 5.3 Accuracy loss when leaving out land-use features

	Accuracy loss		
	Average distance to waterway	Average distance to foot-path	Average distance to cycle-path
Support Vector Machines	1%	1.5%	0.9%
Random Forest	0.7%	0.3%	-0.1%
Multilayer-Perceptron	-0.2%	0.5%	0.3%

6. Conclusion

The research questions and their related sub-questions are answered in this research. This last chapter go over all of the questions and the answers that were found, starting with all of the sub-questions. The first sub-question was:

What is machine learning?

Machine learning is a method of data analysis that combines computer algorithms and large amounts of data to make models that can train itself and learn through iterative processes. It can be used for various tasks like classification tasks, visual recognition, speech recognition and robotics. There is a broad selection of algorithms that can be used, which can be split into conventional methods and in deep learning based methods. Besides this, it can also be split in supervised- and unsupervised learning algorithms. For this research, conventional and supervised machine learning algorithms are further explored.

What is the state of the art in outdoor sports detection in GNSS-data based tracks using machine learning?

Automatic classification of human activity (HAR) is a widely researched field. In this field, a wide variety of methods are being used, utilizing various sensors to collect data. However, data collection using GNSS-sensors has drawn attention of researchers in recent years. HAR studies using data collected by GNSS-data are often accompanied by data acquired by other sensors, like accelerometers, gyroscopes, magnetometers and microphones. Most papers that used machine learning algorithms used multiple machine learning algorithms to find which algorithms yielded the best results. The 'support vector machines'-, 'random forest'-, and 'multilayer perceptron'-algorithms yielded the best results in the papers discussed in this research.

What features need to be extracted from the GNSS- and land-use data to detect the kind of outdoor sport practiced in the recorded tracks?

Nearly all studies extracted features like 'average speed', 'average acceleration', 'trip distance' and 'duration' from the GNSS tracks. Other commonly used variables were 'average heading change' and 'low-speed points'. Some studies also used additional features like '95th percentile of speed' or '95th percentile of acceleration'. All of the variables above were used in this study in addition to variables concerning elevation. Finally, studies used variables like the 'average number of satellites used' and 'average signal-to-noise ratio', however, the data used in this research did not contain this information and these features were therefore not used.

What machine learning algorithms would be suitable for outdoor sport activity detection in GNSS- and land-use data?

Three algorithms were selected as being the most suitable algorithms for this purpose, based on the classification accuracies that were reached in previous research. First, the support vector machines-algorithms makes use of lines and hyperplanes to separate multidimensional data in order to classify them. Second, the random forest-algorithm uses multiple decision trees and bootstrapped data to classify the data. The outcomes of the individual decision trees will be used for a classification through a majority voting. Finally, the multilayer perceptron-algorithm creates a network in which input and output is connected through layers of so called 'hidden-layers'. These nodes are trained to produce usable input-output relationships which can be used to classify the inputs.

How can we validate the machine learning models for detecting the kind of outdoor sport practiced in recorded GNSS-data?

The support vector machines- and multilayer perceptron-models are validated through 10-fold cross validation. 90% of data is used for training and 10% for testing. Each iteration uses different data for training and testing until the models are run in 10-fold and all data has been used for both testing and training. The average accuracy of the 10 models is then taken. The random forest-models bootstrap data for building decision trees. All data that is not bootstrapped is used for testing and will result in an out-of-bag (OOB) accuracy. Validation is done by running the model in 10-fold with different bootstrapped data and averaging their accuracies.

To what extent are machine learning models able to classify the kind of outdoor sport practices in pre-recorded GNSS-data based tracks using related land-use features?

The best models for classifying activity types in pre-recorded GNSS-data tracks were built using the support vector machines-algorithm. The model with land-used features reached an accuracy of 82.6%, while the model without land-use features reached an accuracy of 77.0%. The random forest-algorithm with land-use features reached an accuracy of 74.6% and the multilayer perceptron-algorithm with land-use features reached an accuracy of 69.4%.

The main question for this thesis was the following:

To what extent do land-use features improve the machine learning models' ability to correctly classify the kind of outdoor sport practices in GNSS-tracks?

This research has shown that using land-use features related to GNSS tracks as additional data for machine learning models improves the machine learning models' classification accuracy. Validating the models has verified that, in the cases of support vector machines-, random forest-, and multilayer perceptron algorithms, the chosen land-use features significantly improve the models' accuracy. The largest improvements were found in the SVM model, followed by the MLP and RF models. The SVM model was found to be the most accurate model and the model with the most potential for improving its accuracy using land-use features. Even though the land-use features data values did not always reflect the hypothesis, the various algorithms have still found useful patterns from which different classes could be distinguished. However, in some cases, individual land-use features caused a loss in classification accuracy. Not all land-use features are providing the same effect to the models' accuracy and careful evaluation is still required to achieve a higher classification accuracy. This thesis has proven that it is possible to significantly increase a machine learning models' classification accuracy for classifying sport activities using land-use features related to GNSS-tracks. Machine learning algorithms can find patterns in data that are hard to find and therefore there is a lot of potential for experimenting with land-use features related to GNSS-tracks to achieve higher classification accuracies.

7. Reflection and recommendations

This thesis has explored the opportunities of using land-use data for activity type classification in GNSS-tracks. As the author was new to a lot of the subjects discussed and the methods used, some unforeseen obstacles have occurred which might have influenced the outcome. The following paragraphs will discuss some of these shortcomings and will provide some recommendations and suggestions for future work related to this research.

The data collection proved to be more difficult than expected, especially in specific classes, like ice-skating or kayaking. Classes like cycling and running were rather easy to find, however, the Strava website's rate limit made the collection of these tracks a slow and tedious process. Due to time-constraints, the research was conducted with less tracks than originally anticipated. Using more tracks would have been beneficial to training and testing the model. Another limitation of the data that was collected is that it was self-labelled. Some tracks have been removed as they were identified as being misclassified by the user, but identifying misclassification can be hard and the chances are high that there were still misclassified tracks among the data. Machine learning algorithms are not able to identify these misclassified tracks and will therefore train the model to with an incorrect input-output relationship. Besides obvious misclassifications, there is also the problem of subjectivity, as discussed earlier in the case of walking versus hiking. Another problem with the data is that it all comes from one source. This source already pre-processed the data in some way. The models that were built during this research only validated the classification of this specific pre-processed version of GNSS-tracks, so it is not possible to say how well the model will perform when GNSS-tracks from different sources are being classified.

Then there are some shortcomings regarding the land-use features. OpenStreetMap land-use data was used to calculate these features. However, when querying these features, it is not possible to query the full line-segment or polygon from OSM. Instead, the nodes are queried. These nodes occur at every bend of a line-segment or at every corner of a polygon. In practice, this means that even if someone was inside the polygon or on the line-segment, the closest node to that point is queried, giving a

distorted image of the reality. For example, if a user is swimming in the sea, the distance of the water will be calculated as the closest node separating the mainland and the sea. Another limitation is that only nodes within 50 meters from the GNSS-tracks were queried, due to server limitations on the side of OSM. If this research were to be applied to a bigger scale, it is recommended to make a mirror server of the OSM server.

There are also some shortcomings in the tracks that were used to query the nodes. First, a python module was used for applying the Ramer-Douglas-Peucker algorithm. This algorithm uses the epsilon parameter as a parameter for the simplification threshold. The epsilon is in degrees minutes seconds. Consequentially, the actual threshold in meters varies with the latitude that the track is recorded on and can also vary based on cardinal directions. Second, due to limited processing power, only 20 points per track were used, which means that the average distance from a land-use element is based on only a small portion of the whole track.

Future researchers on this subject are recommended to take this reflection into consideration to work on these shortcomings in future work. Besides working on these shortcomings, future research is also needed to get a better insight into what land-use features could be used for the purpose of reaching a greater classification accuracy. More GNSS features can be extracted in combination with land-use features to research the potential classification accuracy when optimizing both types of features. Finally, future research is needed to see if these models can be combined with automatic detection of activity type change, so the model can be deployed on raw-GPS tracks with no clear-beginning and endpoint.

8. References

Ali, J., Khan, R., Ahmad, N., & Maqsood, I. (2012). Random forests and decision trees. *International Journal of Computer Science Issues (IJCSI)*, *9*(5), 272.

Allahbakhshi, H., Conrow, L., Naimi, B., & Weibel, R. (2020). Using accelerometer and GPS data for real-life physical activity type detection. *Sensors*, *20*(3), 588.

Alpaydin, E. (2014a). Introduction. *Introduction to machine learning* (pp. 1-20). MIT press.

Alpaydin, E. (2014b). Multilayer Perceptrons. *Introduction to machine learning* (pp. 267-313). MIT press.

Bhandari, A. (2020, 3 April). Feature Scaling for Machine Learning: Understanding the Difference Between Normalization vs. Standardization. Retrieved from https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/ on 27 January 2022.

Baeldung (2021, 25 August). Multiclass Classification Using Support Vector Machines. Retrieved from https://www.baeldung.com/cs/svm-multiclass-classification on 1 December 2021.

Bedell, Z. (2018, 7 December). Support Vector Machines Explained. Retrieved from https://medium.com/@zachary.bedell/support-vector-machines-explained-73f4ec363f13 on 16 November 2021.

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

Breiman, L. (2002). Manual on Setting Up, Using, and Understanding Random Forests V3.1. Retrieved from https://www.stat.berkeley.edu/~breiman/Using_random_forests_V3.1.pdf on 1 December 2021.

Brownlee, J (2019, 19 August). When to Use MLP, CNN, and RNN Neural Networks. Retrieved from https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/ on 17 November 2021.

Carto (2017). *The State of Location Intelligence 2018*. New York: Carto.

Chacón-Borrego, F., Corral-Pernía, J. A., Martínez-Martínez, A., & Castañeda-Vázquez, C. (2018). Usage behaviour of public spaces associated with sport and recreational activities*. Sustainability, 10(7)*, 2377.

Chaurasia, S. K., & Reddy, S. R. N. (2021). State-of-the-art survey on activity recognition and classification using smartphones and wearable sensors. *Multimedia Tools and Applications*, 1-32.

Clouthier, A. L., Ross, G. B., & Graham, R. B. (2020). Sensor data required for automatic recognition of athletic tasks using deep neural networks. *Frontiers in bioengineering and biotechnology*, *7*, 473.

Curry, D. (2021, 24 March). Strava Revenue and Usage Statistics (2021). Retrieved from: https://www.businessofapps.com/data/strava-statistics/ on 11 October 2021

Ellis, K., Godbole, S., Marshall, S., Lanckriet, G., Staudenmayer, J., & Kerr, J. (2014). Identifying active travel behaviors in challenging environments using GPS, accelerometers, and machine learning algorithms. *Frontiers in public health*, *2*, 36.

Fang, Z., Jian-yu, L., Jin-jun, T., Xiao, W., & Fei, G. (2018). Identifying activities and trips with GPS data. *IET Intelligent Transport Systems*, *12*(8), 884-890.

Feng, T., & Timmermans, H. J. (2013). Transportation mode recognition using GPS and accelerometer data. *Transportation Research Part C: Emerging Technologies*, *37*, 118-130.

Feng, T., & Timmermans, H. J. (2015). Detecting activity type from GPS traces using spatial and temporal information. *European Journal of Transport and Infrastructure Research*, *15*(4).

Ferrari, A., Micucci, D., Mobilio, M., & Napoletano, P. (2021). Trends in human activity recognition using smartphones. *Journal of Reliable Intelligent Environments*, 1-25.

Ferri, C. (2016). Identifying the sport activity of GPS tracks. *Procedia Computer Science*, *80*, 301-312.

Frousiakis, G. (2018, March 6). The Business Advantages of a Multi GNSS Set-Up. Retrieved from https://www.telit.com/blog/multi-gnss-business-advantages/ on 12 November 2021.

Gad, A. (2018, June 27). Beginners Ask “How Many Hidden Layers/Neurons to Use in Artificial Neural Networks?”. Retrieved from https://towardsdatascience.com/beginners-ask-how-many-hidden-layers-neurons-to-use-in-artificial-neural-networks-51466afa0d3e on 1 December 2021.

GalileoGNSS (2016, 4 September). BQ Aquaris X5 Plus. First European Galileo-ready smartphone. Retrieved from https://galileognss.eu/bq-aquaris-x5-plus-first-european-galileo-ready-smartphone/ on 12 November 2021.

GalileoGNSS (2017, 28 November). Is your phone using Galileo? Retrieved from https://galileognss.eu/is-your-phone-using-galileo/ on 12 November 2021.

Gautier, L. (2021, 4 June). Rpy2 3.4.5. Retrieved from https://pypi.org/project/rpy2/ on 30 November 2021.

Geopy (2018). Welcome to GeoPy’s documentation! Retrieved from https://geopy.readthedocs.io/en/stable/ on 29 November 2021.

Ho, T. (1995). Random decision forest. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition, 1995* (pp. 278-282).

Janssen, M., Goudsmit, J., Lauwerijssen, C., Brombacher, A., Lallemand, C., & Vos, S. (2020). How Do Runners Experience Personalization of Their Training Scheme: The Inspirun E-Coach?. *Sensors*, *20*(16), 4590.

Janssen, M., Scheerder, J., Thibaut, E., Brombacher, A., & Vos, S. (2017). Who uses running apps and sports watches? Determinants and consumer profiles of event runners’ usage of running-related smartphone applications and sports watches. *PloS one*, *12*(7), e0181167.

Jun, J., Guensler, R., & Ogle, J. (2005). Smoothing methods designed to minimize the impact of GPS random error on travel distance, speed, and acceleration profile estimates. *Transportation Research Record*.

Karthaus, K. (2012). Javascript implementation of the Ramer Douglas Peucker Algorithm. Retrieved from https://karthaus.nl/rdp/ on November 26 2021.

Kingsford, C., & Salzberg, S. L. (2008). What are decision trees?. *Nature biotechnology*, *26*(9), 1011-1013.

Krajina, T. (2021, 4 November). Gpxpy 1.5.0. Retrieved from https://pypi.org/project/gpxpy/#data

Lee, K., & Kwan, M. P. (2018). Automatic physical activity and in-vehicle status classification based on GPS and accelerometer data: A hierarchical classification approach using machine learning techniques. *Transactions in GIS*, *22*(6), 1522-1549.

Liao, S. H., & Wen, C. H. (2007). Artificial neural networks classification and clustering of methodologies and applications–literature analysis from 1995 to 2005. *Expert Systems with applications*, *32*(1), 1-11.

Li, X., Ge, M., Dai, X., Ren, X., Fritsche, M., Wickert, J., & Schuh, H. (2015). Accuracy and reliability of multi-GNSS real-time precise positioning: GPS, GLONASS, BeiDou, and Galileo. *Journal of Geodesy*, *89*(6), 607-635.

Li, L., Zhu, J., Zhang, H., Tan, H., Du, B., & Ran, B. (2020). Coupled application of generative adversarial networks and conventional neural networks for travel mode detection using GPS data. *Transportation Research Part A: Policy and Practice*, *136*, 282-292.

Machine Learning Knowledge (2021, 12 August). How to deal with Missing Data in Machine Learning. Retrieved from https://machinelearningknowledge.ai/missing-data-in-machine-learning/ on 9 February 2022.

Mahapatra, S. (2018, 21 March). Why Deep Learning over Traditional Machine Learning? Retrieved from https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063 on 16 November 2021.

Mardini, M. T., Bai, C., Wanigatunga, A. A., Saldana, S., Casanova, R., & Manini, T. M. (2021). Age Differences in Estimating Physical Activity by Wrist Accelerometry Using Machine Learning. *Sensors*, *21*(10), 3352.

Martin, B. D., Addona, V., Wolfson, J., Adomavicius, G., & Fan, Y. (2017). Methods for real-time prediction of the mode of travel using smartphone-based GPS and accelerometer data. *Sensors*, *17*(9), 2058.

Mitchell, T. (1997). Preface, *Machine learning* (p. XV). New York: McGraw-Hill.

MTL (n.d.). Calculate distance, bearing and more between Latitude/Longitude points. Retrieved from https://www.movable-type.co.uk/scripts/latlong.html on 29 November 2021.

Noble, W. S. (2006). What is a support vector machine?. *Nature biotechnology*, *24*(12), 1565-1567.

Noriega, L. (2005). Multilayer perceptron tutorial. *School of Computing. Staffordshire University*.

Olsen, L.R. (2021, 14 November). Multiple-k: Picking he number of folds for cross-validation. Retrieved from https://cran.r-project.org/web/packages/cvms/vignettes/picking_the_number_of_folds_for_cross-validation.html on 7 February 2022.

OpenStreetMap (n.d.). OpenStreetMap levert kaartgegevens aan duizenden websites, mobiele apps en hardware-apparaten. Retrieved from https://www.openstreetmap.org/about on 25 November 2021.

OSM Wiki (2021, 15 November). Overpass API. Retrieved from https://wiki.openstreetmap.org/wiki/Overpass_API on 15 November 2021.

Ouchi, K., & Doi, M. (2012, September). Indoor-outdoor activity recognition by a smartphone. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (pp. 600-601).

Paziewski, J. (2020). Recent advances and perspectives for positioning and applications with smartphone GNSS observations. *Meas. Sci. Technol*, *31*(091001), 091001.

Paziewski, J., Fortunato, M., Mazzoni, A., & Odolinski, R. (2021). An analysis of multi-GNSS observations tracked by recent Android smartphones and smartphone-only relative positioning results. *Measurement*, *175*, 109162.

Phibo (2021, April 20). Overpy 0.6. Retrieved from https://pypi.org/project/overpy/ on 29 November 2021.

RDocumentation (2018, 25 March). randomForest: Classification and Regression with Random Forest. Retrieved from https://www.rdocumentation.org/packages/randomForest/versions/4.6-14/topics/randomForest on 1 December 2021.

RDocumentation (2021a, 16 September). svm: Support Vector Machines. Retrieved from https://www.rdocumentation.org/packages/e1071/versions/1.7-9/topics/svm on 30 November 2021.

RDocumenation (2021b, 13 August). mlp: Create and train a multi-layer perceptron (MLP). Retrieved from https://www.rdocumentation.org/packages/RSNNS/versions/0.4-14/topics/mlp on 1 December 2021.

Richards, N. M., & King, J. H. (2014). Big data ethics. *Wake Forest L. Rev.*, *49*, 393.

Rodriguez, D. (2020, 16 November). Multilayer-perceptron. Retrieved from https://github.com/d-r-e/multilayer-perceptron on 17 November 2021.

Ross, G. B., Dowling, B., Troje, N. F., Fischer, S. L., & Graham, R. B. (2018). Objectively differentiating movement patterns between elite and novice athletes. *Med Sci Sports Exerc*, *50*(7), 1457-1464.

Rossi, A., Pappalardo, L., Cintia, P., Iaia, F. M., Fernández, J., & Medina, D. (2018). Effective injury forecasting in soccer with GPS training data and machine learning. *PloS one*, *13*(7), e0201264.

Schutz, Y., & Chambaz, A. (1997). Could a satellite-based navigation system (GPS) be used to assess the physical activity of individuals on earth?. *European journal of clinical nutrition*, *51*(5), 338-339.

Scikit-Learn (n.d.). RBF SVM parameters. Retrieved from https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html on 1 December 2021.

Shafique, M. A., & Hato, E. (2016). Travel mode detection with varying smartphone data collection frequencies. *Sensors*, *16*(5), 716.

Shoaib, M., Bosch, S., Incel, O. D., Scholten, H., & Havinga, P. J. (2015). A survey of online activity recognition using mobile phones. *Sensors*, *15*(1), 2059-2085.

Scornet, E. (2017). Tuning parameters in random forests. *ESAIM: Proceedings and Surveys*, *60*, 144-162.

Stathakis, D. (2009). How many hidden layers and nodes?. *International Journal of Remote Sensing*, *30*(8), 2133-2147.

Statista (2021a, August 6). Smartphone users worldwide 2016-2021. Retrieved from https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/ on 11 November 2021.

Statista (2021b, July 6). Number of health and fitness app users in the United States from 2018 to 2022. Retrieved from https://www.statista.com/statistics/1154994/number-us-fitness-health-app-users/ on 11 November 2021.

Strava (2020, 9 December). Servicevoorwaarden Strava. Retrieved from https://www.strava.com/legal/terms on 11 October 2021.

Strava Support (2021, 6 August). How Distance is calculated. Retrieved from https://support.strava.com/hc/en-us/articles/216919487-How-Distance-is-Calculated on 1 December 2021.

Tamura, K., Wilson, J. S., Puett, R. C., Klenosky, D. B., Harper, W. A., & Troped, P. J. (2018). Accelerometer and GPS analysis of trail use and associations with physical activity. *Journal of Physical Activity and Health*, *15*(7), 523-530.

Towards Data Science (2020). Random Forest: Hyperparameters and How to Fine-Tune Them. Retrieved from https://towardsdatascience.com/random-forest-hyperparameters-and-how-to-fine-tune-them-17aee785ee0d on 19 February 2022.

Vogel, E.A. (2020, 9 January). About one-in-five Americans use a smart watch or fitness tracker. Retrieved from https://www.pewresearch.org/fact-tank/2020/01/09/about-one-in-five-americans-use-a-smart-watch-or-fitness-tracker/ on 11 November 2021.

Wood, T. (n.d.) What is the F-score? Retrieved from https://deepai.org/machine-learning-glossary-and-terms/f-score on 3 February 2022.

World Health Organization. (2019). *Global action plan on physical activity 2018-2030: more active people for a healthier world*. World Health Organization.

Wu, L., Yang, B., & Jing, P. (2016). Travel mode detection based on GPS raw data collected by smartphones: a systematic review of the existing methodologies. *Information*, *7*(4), 67.

Xiao, G., Cheng, Q., & Zhang, C. (2019). Detecting travel modes using rule-based classification system and Gaussian process classifier. *IEEE Access*, *7*, 116741-116752.

Xiao, G., Juan, Z., & Zhang, C. (2015). Travel mode detection based on GPS track data and Bayesian networks. *Computers, Environment and Urban Systems*, *54*, 14-22.

Yang, M., Pan, Y., Darzi, A., Ghader, S., Xiong, C., & Zhang, L. (2021). A data-driven travel mode share estimation framework based on mobile device location data. *Transportation*, 1-45.

Zell, A. (1998). Stuttgart Neural Network Simulator. Retrieved from http://www.ra.cs.uni-tuebingen.de/SNNS/welcome.html on 1 December 2021.

Zong, F., Bai, Y., Wang, X., Yuan, Y., & He, Y. (2015). Identifying travel mode with GPS data using support vector machines and genetic algorithm. *Information*, *6*(2), 212-227.

Zong, F., Yuan, Y., Liu, J., Bai, Y., & He, Y. (2017). Identifying travel mode with GPS data. *Transportation planning and technology*, *40*(2), 242-255.

9. Appendices

Appendix A: Table of literature of HAR using GNSS for literature review

Reference	Type of activities	Devices used for data collection	Features	Classification algorithms	Accuracy
Shafique & Hato, 2016	Walk, bicycle, car, bus, train, subway	GNSS, accelerometers, gyroscope	Pitch, roll, maximum average resultant acceleration, maximum resultant acceleration, skewness, kurtosis, resultant acceleration	RF	99.8%
Xiao, Cheng & Zhang, 2019	Subway, walk, bicycle, e-bicycle, bus, car	GNSS	Minimum duration, maximum duration, maximum speed of points, maximum distance from the start point of the segment to the nearest entrance/exit, maximum distance from the end point of the segment to the nearest entrance/exit, maximum distance of the segment to the nearest subway line	SVM	89.16%
				MNL	82.10%
				BN	90.40%
				ANN	93.40%
				GPC	93.82%
Xiao, Juan & Zhang, 2015	Walk, bike, e-bike, car, bus	GNSS	Average speed, 95% percentile speed, average absolute acceleration, and travel distance	SVM	92.32%
				MNL	84.32%
				ANN	91.95%
				BN	94.74%
Zong et al., 2017	Walk, bike, subway, bus, car	GNSS	Average speed, maximum speed, 75th percentile of speed, acceleration, 75th percentile of acceleration, travel time, standard deviation of speed	GIS algorithms and multinomial logit model.	86%

Zong et al,. 2015	Walk, bike, subway, bus, car	GNSS	Average speed, maximum speed, 75th percentile of speed, acceleration, 75th percentile of acceleration, travel time, trip distance, standard deviation of speed	SVM	92.2%
Lee & Kwan, 2018	Running, walking, sitting, standing, in-vehicle, biking	GNSS, Acceler ometer	Average velocity, average acceleration, max velocity, max acceleration, change rate of velocity	GIS algorithms	Ranging from 69.98% to 98.25%
Ellis et al., 2014	Bike, bus, car, sit, stand, walk	GNSS, accelero meter	49 different features	kNN	86.2%
				Naïve Bayes	74.2%
				SVM	87.7%
				Decision tree	83.6%
				RF	89.8%
Feng & Timme rman, 2013	Walking, cycling, running, motorcycle, bus, car, metro, tram	GNSS, accelero meter	Average speed, maximum speed, non-moving time duration, average value, and standard deviation of the three-axis acceleration change	BBN	96%
Martin et al., 2017	Walk, bike, car, bus, rail	GNSS, accelero meter	Mean change in acceleration, 80th percentile speed, variance change in acceleration, variance change in speed, maximum speed, mean speed, mean change in speed, 80th percentile in speed, variance speed, median speed, maximum speed, mean speed	kNN	94%
				RF	97%
Allahba khshi, Conro w Naimi & Weibel , 2020	Cycle, lie, non-level walk, run, sit, stand, walk	GNSS, accelero meter	85 features	RF	Ranging from 81% tot 99%
Feng & Timme	Paid work, daily shopping, non-daily shopping,	GNSS, prompt	Features on spatial location, aggregate timing, and duration information	BN	46.2%
				DT	69.8%

rman, 2015	help parents/children, recreational, social, voluntary work, service, leisure, picking-up people, study	ed recall data		RF	96.8%
Ferri, 2016	Mountain biking, cycling, sailing, kayaking, mountaineering, hiking, running, trail running, trail biking, motorcycling	GNSS	2-dimensional length, 3-dimensional length, moving time, stopped time, moving distance, stopped distance, maximum speed, average speed, uphill meters, downhill meters, maximum elevation, minimum elevation	J48	62.8%
				JRip	60.9%
				Logist	56.9%
				NB	45.5%
				IBK	56.4%
				RF	67.7%
				Bagging	66.9%
				PART	61.0%
				Booster	67.1%
				SVM	54.1%
				LB	61.1%
				MDA	50.8%
Li et al., 2020	Walk, bike, drive, train	GNSS	Speed, acceleration, bearing rate, jerk	ANN	≈ 60%
				SVM	≈ 65%
				RF	≈ 75%
				Mixed methods	87.6%

Appendix B: Software, programming languages and extensions used during the research

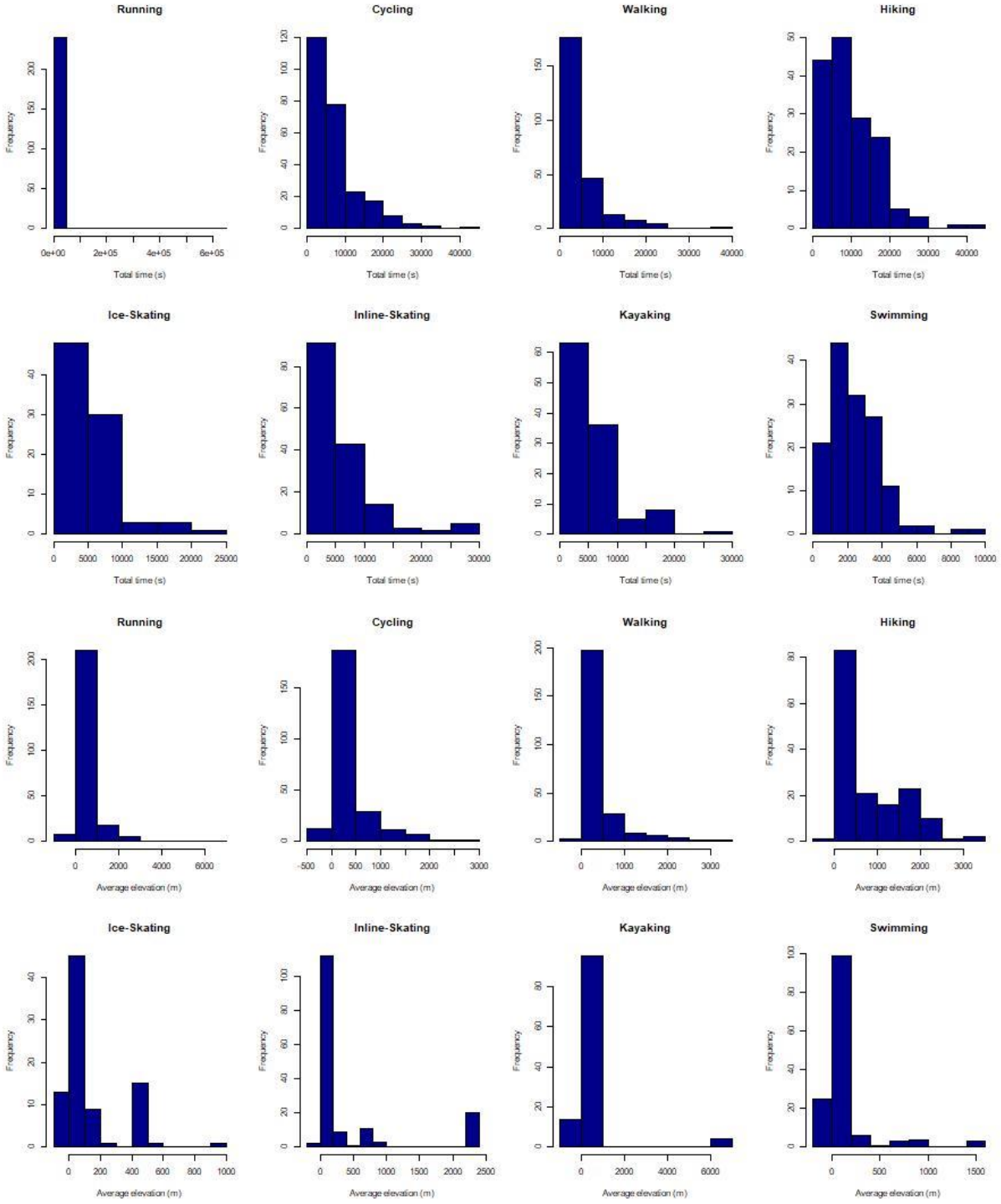
Software	**Programming Language**	**Extensions/Libraries**
Spyder	Python	GpxPy, OS, CSV, math, numPy, time, OverPy, Geopy, Rpy2, rdp
R Studio	R	e1071, randomForest, RSNNS
Overpass API	Overpass QL	
GPXSee	GML	
Strava	GML	

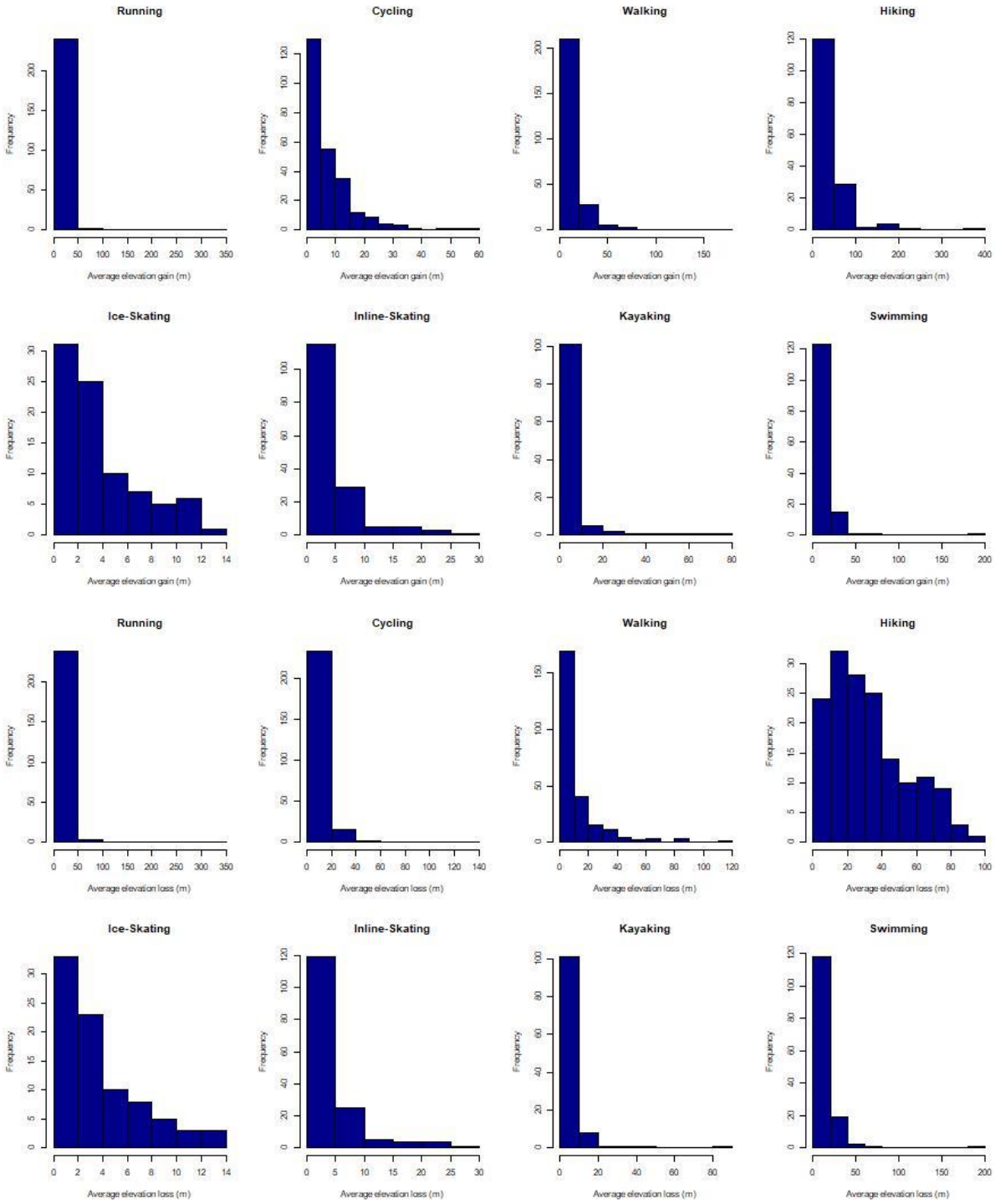
Appendix C: Keys and values used for querying land-use data from OSM

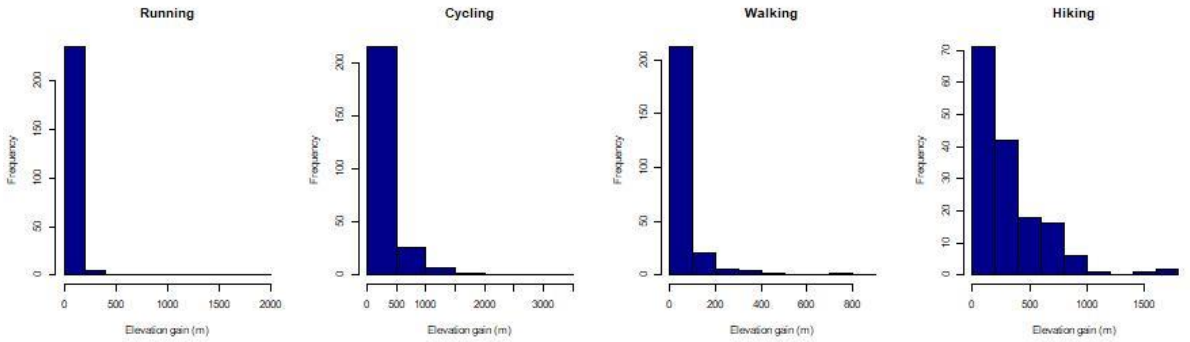
Land-use type	**Key**	**Value**
Waterway	Waterway	All values
	Water	All values
	Leisure	Water_park
	Leisure	Swimming_pool

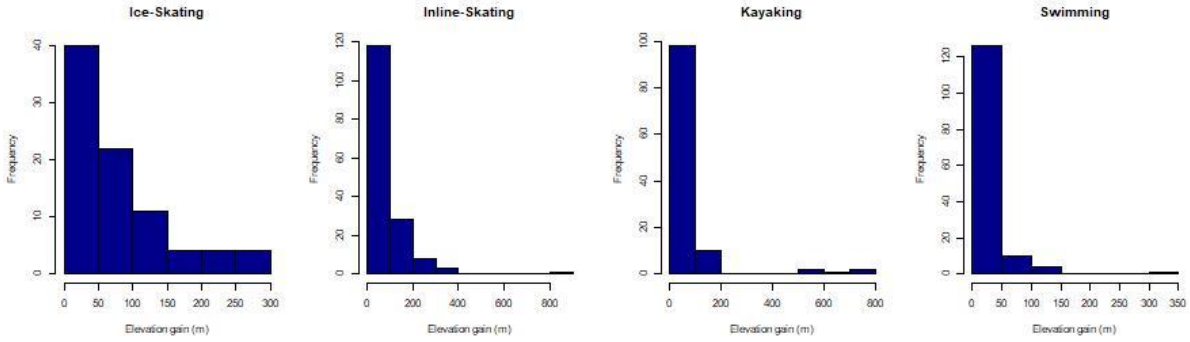
	Leisure	Ice_rink
	Landcover	Water
	Sport	Swimming
	Place	Ocean
	Place	Sea
	Natural	Bay
	Natural	Coastline
	Natural	Water
Foot-path	Highway	Footway
	Highway	Path
	Foot	Yes
	Foot	Designated
	Foot	Private
	Foot	Official
	Footway	All values
	Route	Foot
	Construction	Footway
	Surface	Tartan
Cycle-path	Highway	Cycleway
	Cycleway	Lane
	Cycleway	Track
	Cycleway	Opposite
	Cycleway	Crossing
	Cycleway	Oppostie_lane
	Cycle_network	All values
	Route	Bicycle
	Bicycle	Yes
	Bicycle	Designated
	Bicycle_road	All values

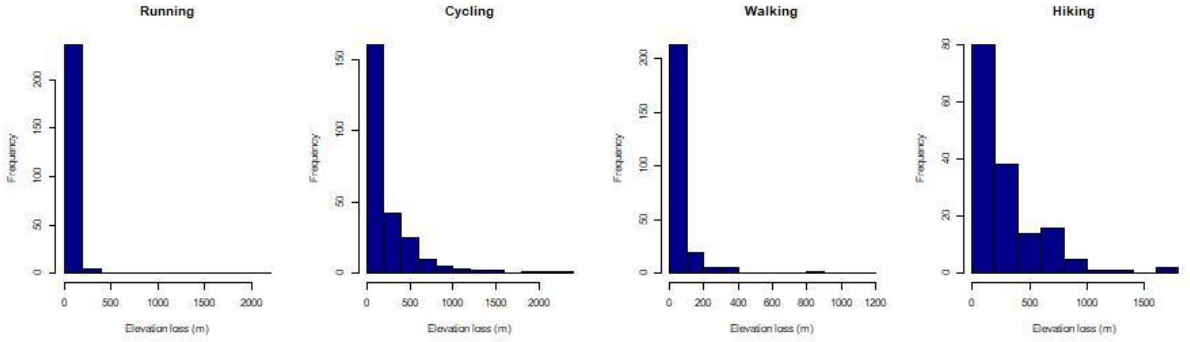
Appendix D: Histograms for all classes per feature

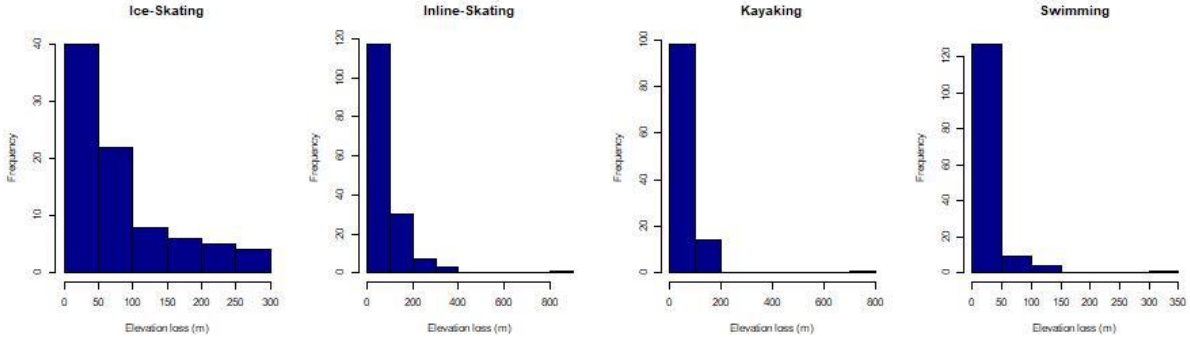


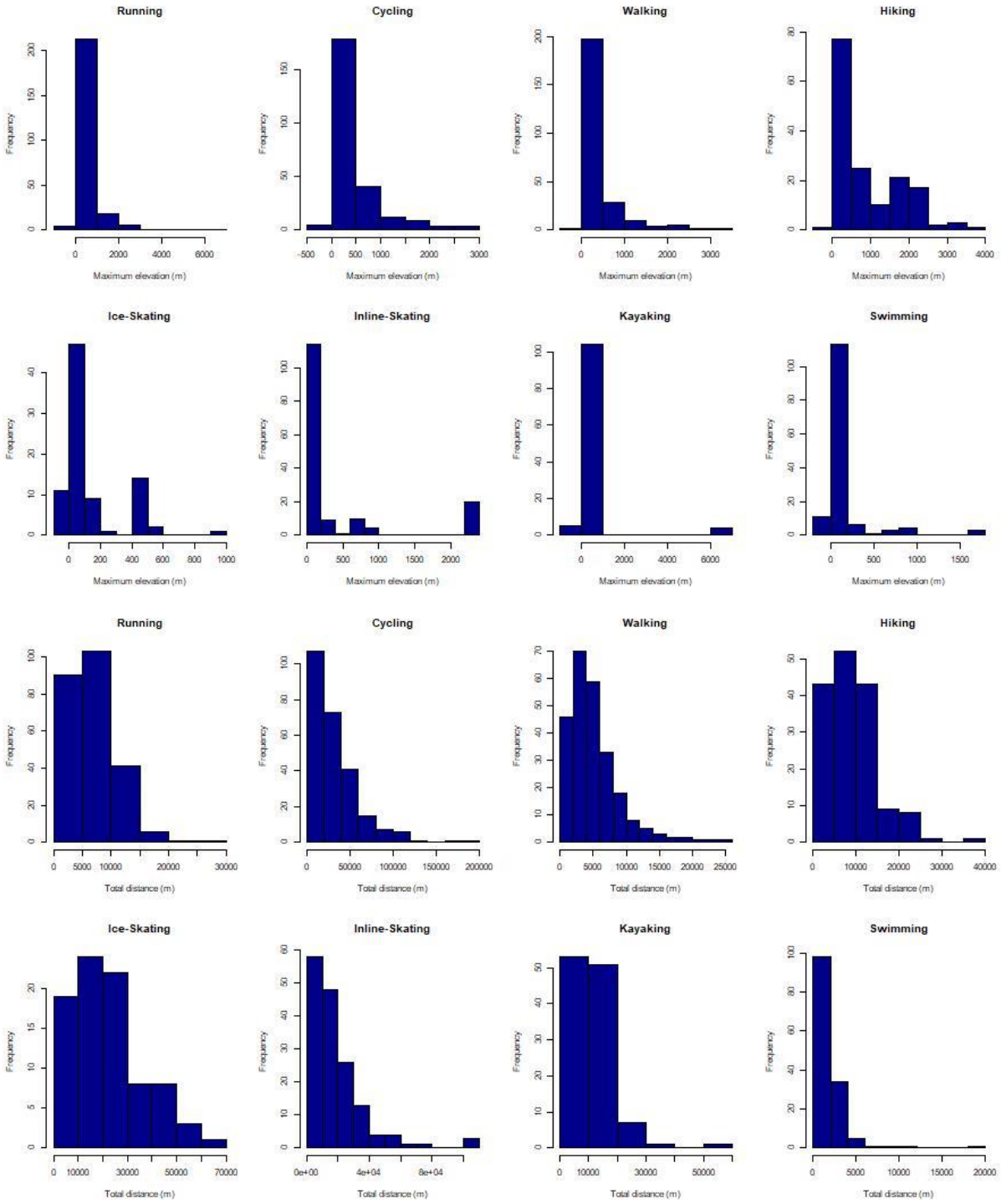

Running
Frequency
Average elevation gain (m)
Cycling
Frequency
Average elevation gain (m)
Walking
Frequency
Average elevation gain (m)
Hiking
Frequency
Average elevation gain (m)
Ice-Skating
Frequency
Average elevation gain (m)
Inline-Skating
Frequency
Average elevation gain (m)
Kayaking
Frequency
Average elevation gain (m)
Swimming
Frequency
Average elevation gain (m)
Running
Frequency
Average elevation loss (m)
Cycling
Frequency
Average elevation loss (m)
Walking
Frequency
Average elevation loss (m)
Hiking
Frequency
Average elevation loss (m)
Ice-Skating
Frequency
Average elevation loss (m)
Inline-Skating
Frequency
Average elevation loss (m)
Kayaking
Frequency
Average elevation loss (m)
Swimming
Frequency
Average elevation loss (m)

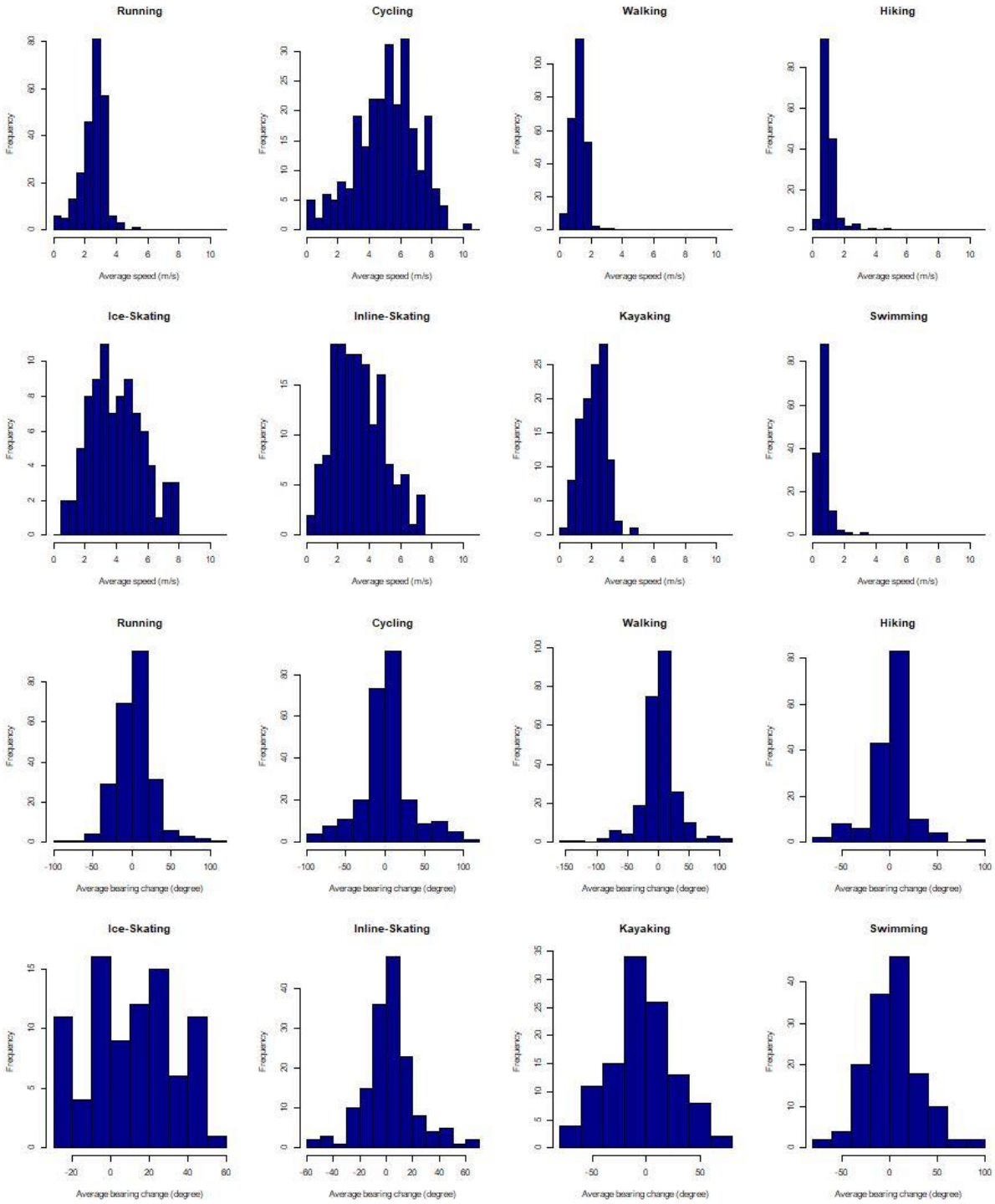

Running
Frequency
Elevation gain (m)
Cycling
Frequency
Elevation gain (m)
Walking
Frequency
Elevation gain (m)
Hiking
Frequency
Elevation gain (m)

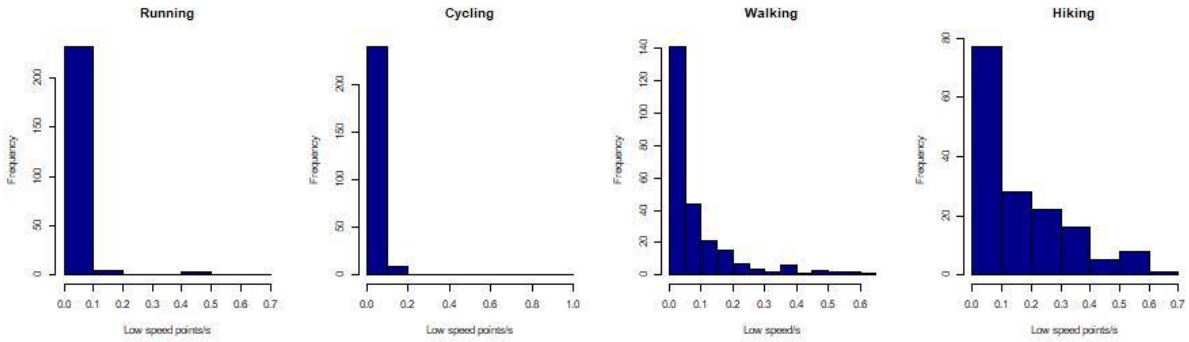

Ice-Skating
Frequency
Elevation gain (m)
Inline-Skating
Frequency
Elevation gain (m)
Kayaking
Frequency
Elevation gain (m)
Swimming
Frequency
Elevation gain (m)

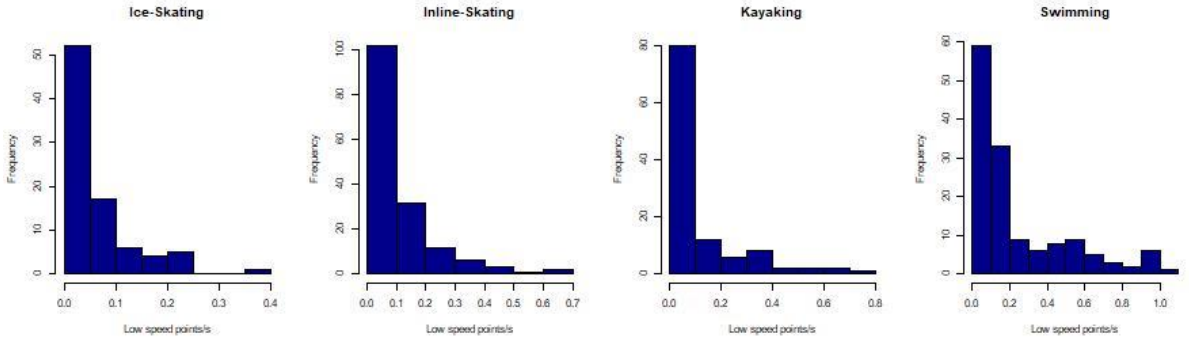

Running
Frequency
Elevation loss (m)
Cycling
Frequency
Elevation loss (m)
Walking
Frequency
Elevation loss (m)
Hiking
Frequency
Elevation loss (m)

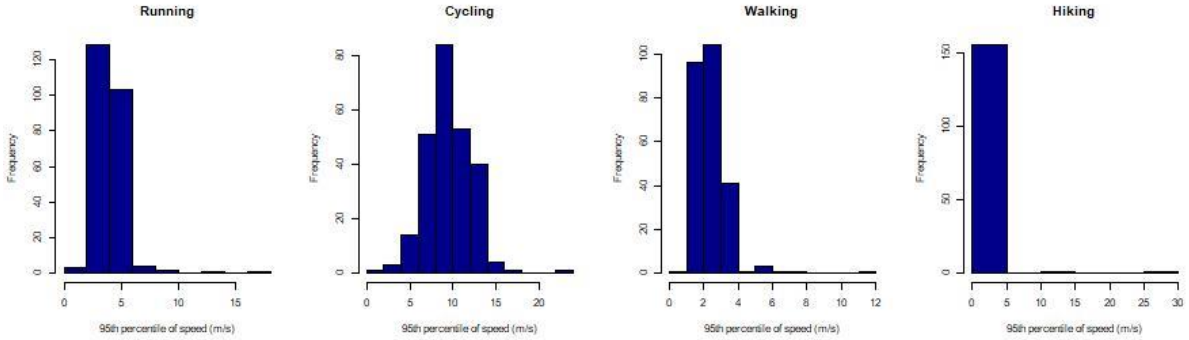

Ice-Skating
Frequency
Elevation loss (m)
Inline-Skating
Frequency
Elevation loss (m)
Kayaking
Frequency
Elevation loss (m)
Swimming
Frequency
Elevation loss (m)

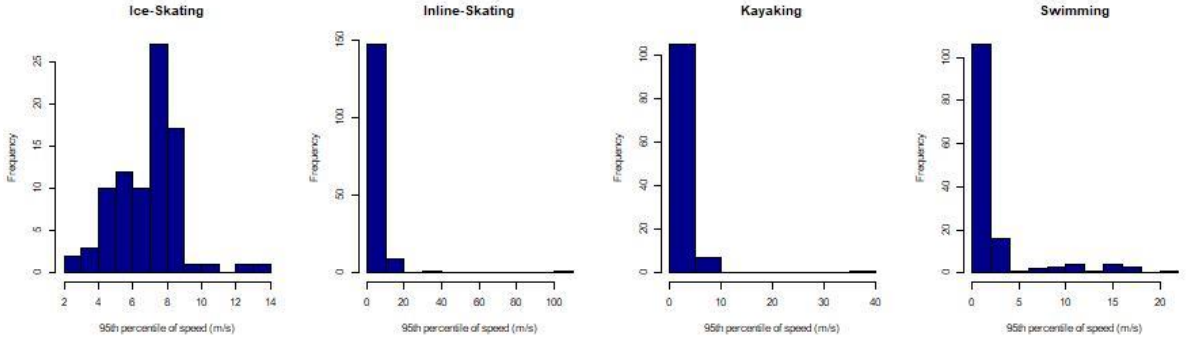

Running
Frequency
Maximum elevation (m)
Cycling
Frequency
Maximum elevation (m)
Walking
Frequency
Maximum elevation (m)
Hiking
Frequency
Maximum elevation (m)
Ice-Skating
Frequency
Maximum elevation (m)
Inline-Skating
Frequency
Maximum elevation (m)
Kayaking
Frequency
Maximum elevation (m)
Swimming
Frequency
Maximum elevation (m)
Running
Frequency
Total distance (m)
Cycling
Frequency
Total distance (m)
Walking
Frequency
Total distance (m)
Hiking
Frequency
Total distance (m)
Ice-Skating
Frequency
Total distance (m)
Inline-Skating
Frequency
Total distance (m)
Kayaking
Frequency
Total distance (m)
Swimming
Frequency
Total distance (m)

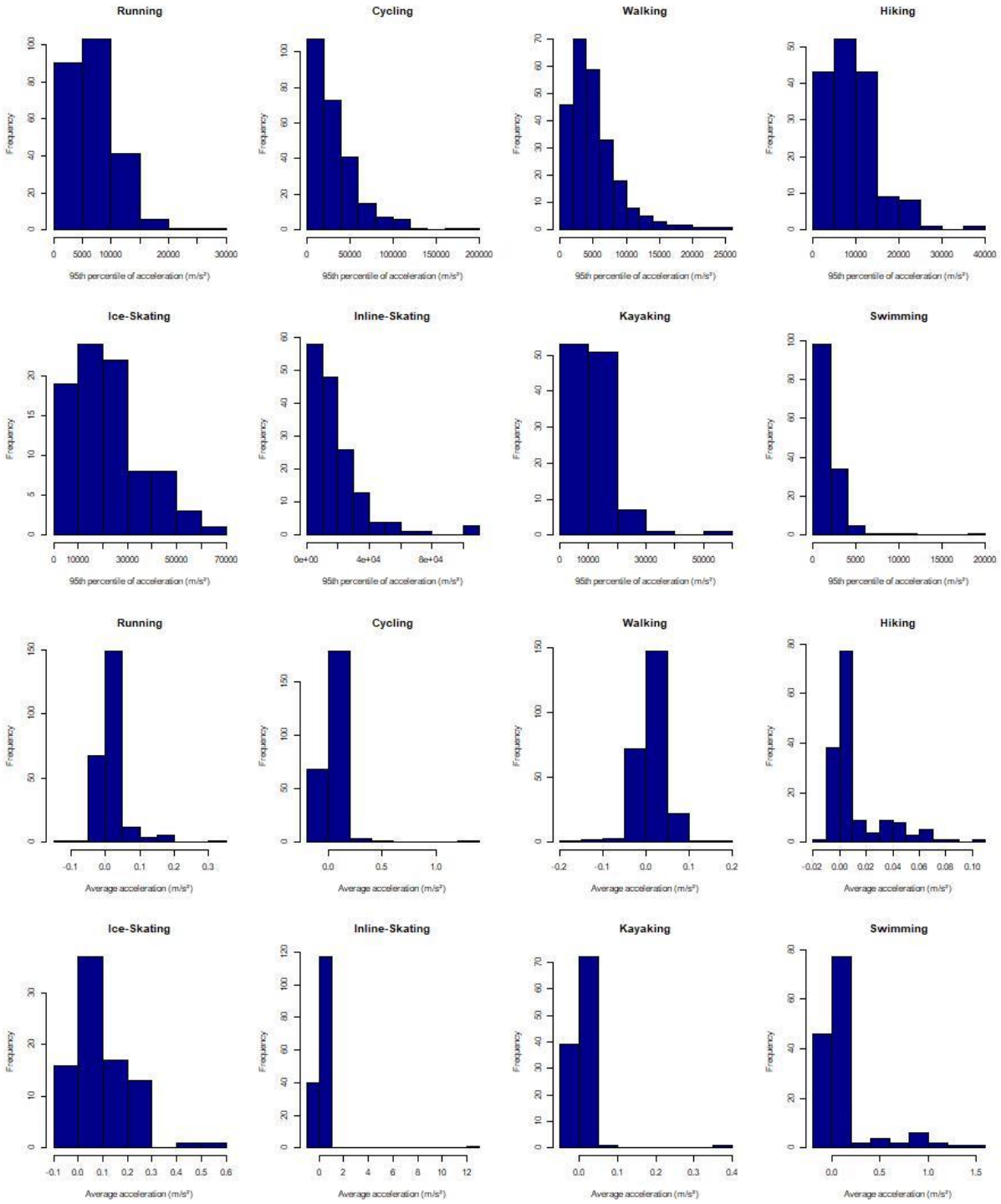

Running
Cycling
Walking
Hiking
Ice-Skating
Inline-Skating
Kayaking
Swimming
Running
Cycling
Walking
Hiking
Ice-Skating
Inline-Skating
Kayaking
Swimming
Frequency
Average speed (m/s)
Average bearing change (degree)

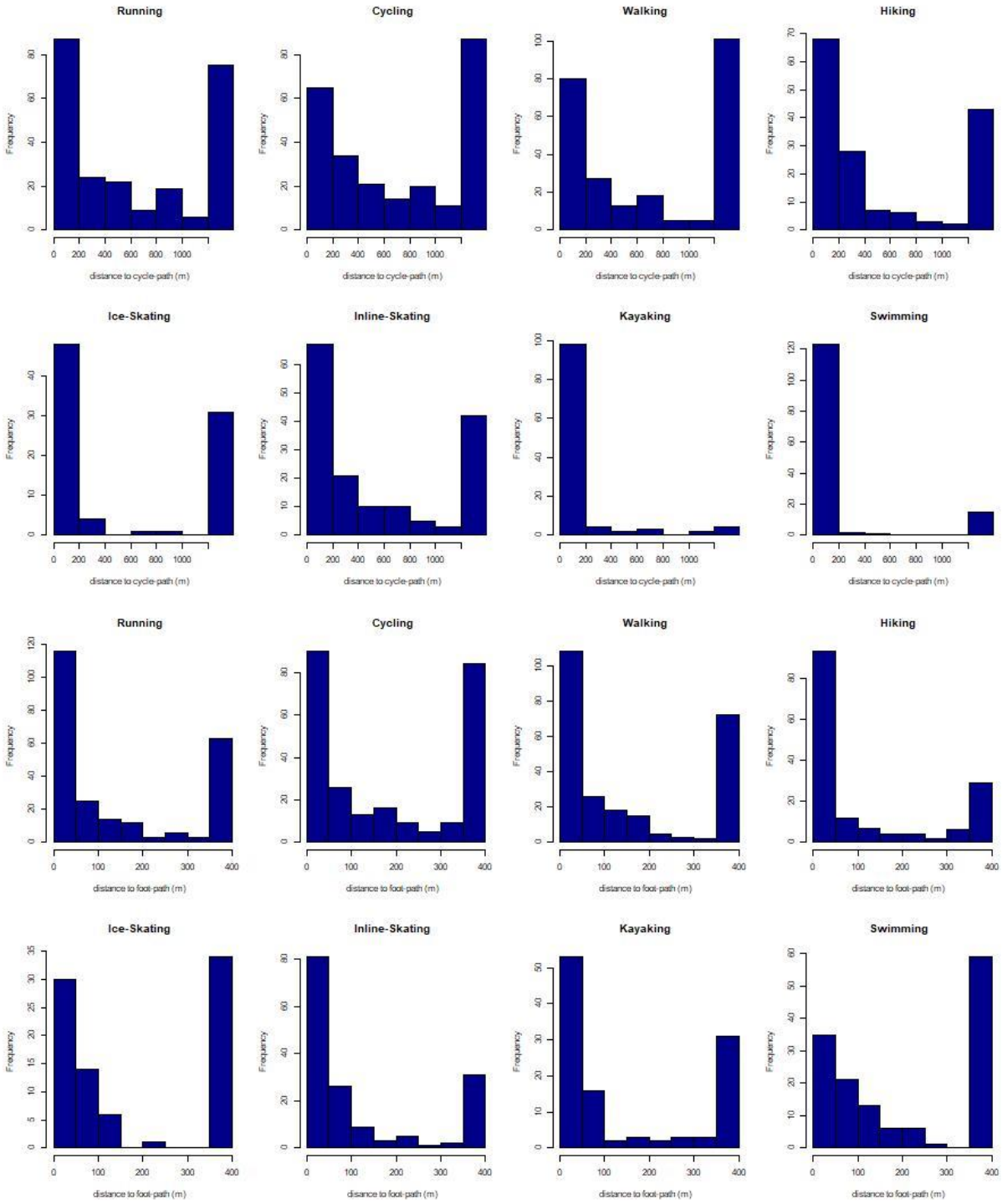

Running
Cycling
Walking
Hiking
Frequency
Low speed points/s
Low speed/s

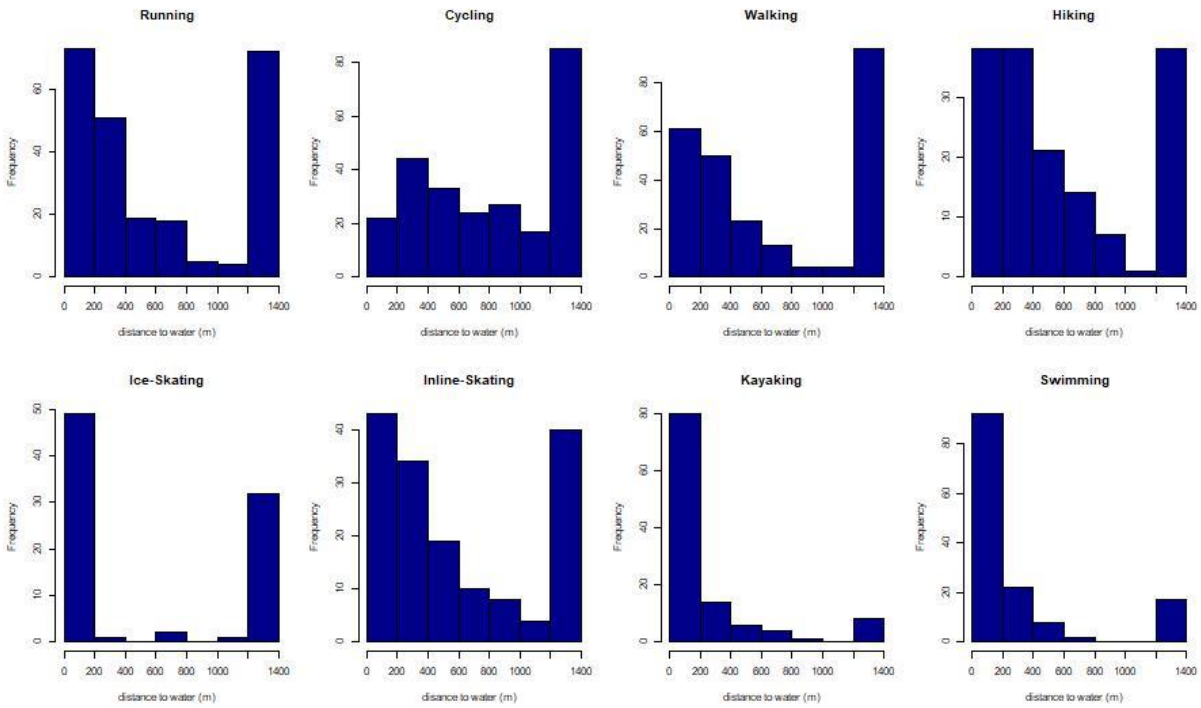

Ice-Skating
Inline-Skating
Kayaking
Swimming
Frequency
Low speed points/s


Running
Cycling
Walking
Hiking
Frequency
95th percentile of speed (m/s)

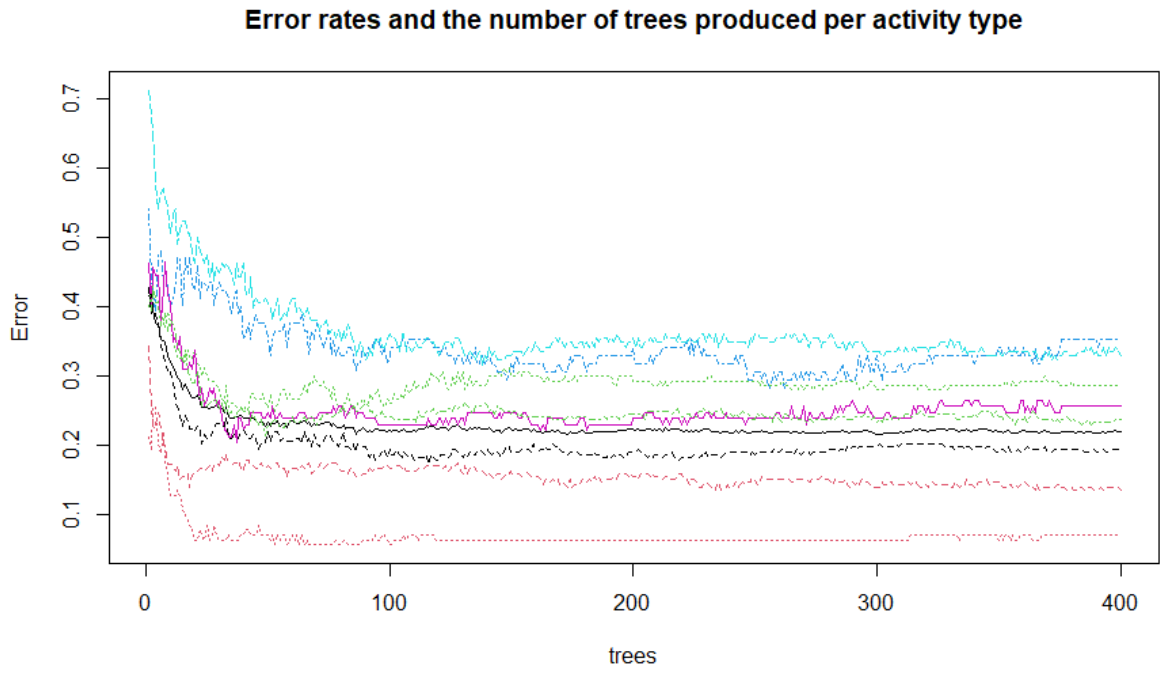

Ice-Skating
Inline-Skating
Kayaking
Swimming
Frequency
95th percentile of speed (m/s)


Running
Cycling
Walking
Hiking
Ice-Skating
Inline-Skating
Kayaking
Swimming
Frequency
95th percentile of acceleration (m/s²)
Average acceleration (m/s²)

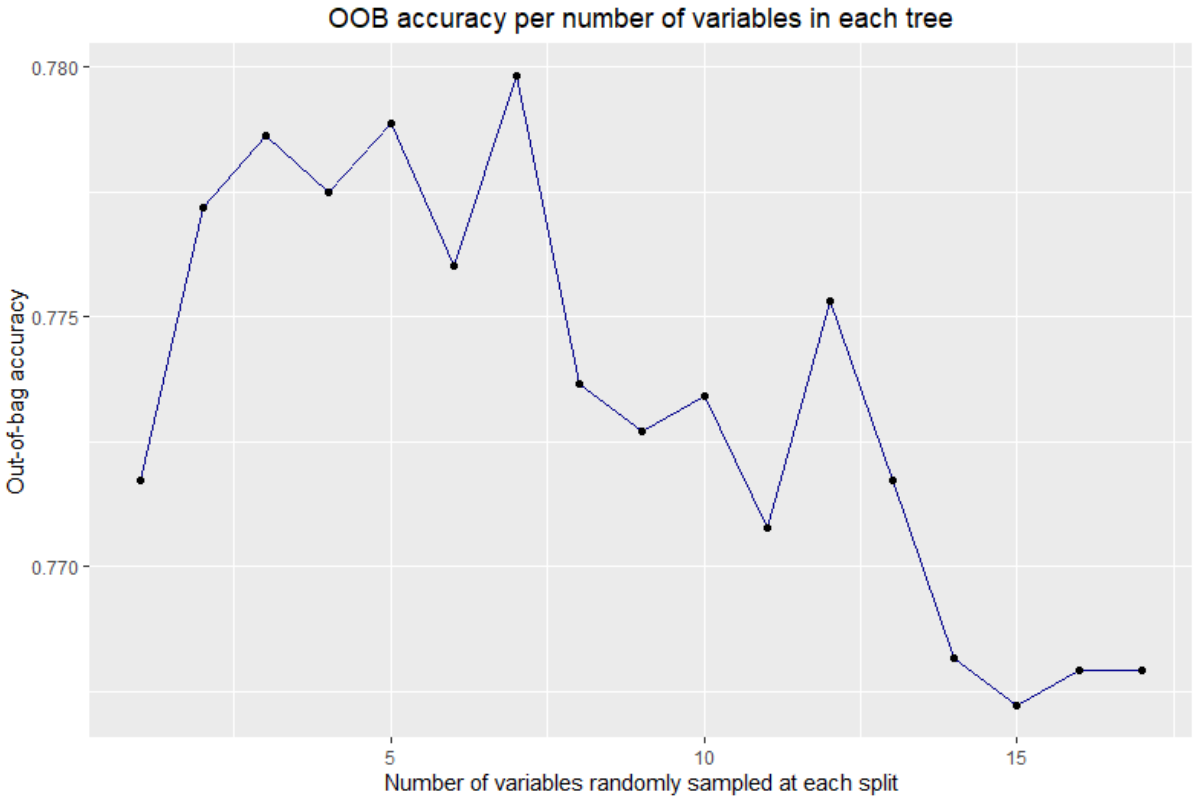

Running
Cycling
Walking
Hiking
Ice-Skating
Inline-Skating
Kayaking
Swimming
Frequency
distance to cycle-path (m)
distance to foot-path (m)



Appendix E: Error rates and the number of trees produced per activity type in random forest



Appendix F: OOB accuracy per number of variables in each tree for random forest



Appendix G: OOB accuracy per number of nodes in each tree random forest