Abstract

This thesis explores the possibility of using the smartphone for advanced activity tracking in the mountain bike sport. The capabilities of the platform are compared to those of a device dedicated to the purpose, to uncover if it might be a suitable alternative to systems requiring the integration of external devices.

The automatic classification of sports activities, a comparison of devices

A Master Thesis written by:

BA. Palle Preben Hansen

On the subject of: Human Computer Interaction

In the University of Aalborg, faculty of Computer Science

June 10, 2020

Table of contents

1 Introduction

Advances in microelectronics has produced technology sufficiently small to have practical application in sports. Micro Electromechanical Systems (MEMS) sensors and feedback mechanisms may be worn by athletes or mounted on equipment, benefitting sports people interested in improving their safety or performance, as well as researchers concerned with the analysis of sports activities in the field. This has transformed the way professional athletes and organizations prepare for and practice their sport, while expanding the scientific field to also encompass electronic and computer sciences (James and Petrone 2016). In sports exercised despite involving an element of danger, often referred to as extreme sports or action sports, such technology is applicable for both safety and performance enhancing measures. These characteristics make safety and performance enhancing technologies for widely exercised extreme sports, as for example mountain biking (MTB), desirable for both amateurs and professional athletes. This is a fact that especially commercial enterprises are eager to take advantage of.

The MTB sport has experienced a rapid increase in interest in the latest years (Hansen, The TrailMe System 2019)[1], both on an amateur and a professional level, in various disciplines more or less extreme (cross-country, down-hill, slopestyle etc.). The market for products catering to the domain is growing with it, which might pose lucrative business prospects. The sport is generally characterized by fast and rough riding in steep and uneven terrain, as well as clearing different obstacles.

There is no shortage in commercially available devices intended for the MTB sport. The products range from advanced bicycle computers offering a wide range of extra, external sensors, such as heart rate monitor, cadence sensor and odometer, e.g. (Garmin 2020), to crash detection systems. One thing many of the products have in common is that they can be categorized as activity trackers. Such systems are characterized by integrating functionality which uses sensors to capture the movements or vital signs of the user to enable them in visualizing their efforts, often with the goal of improving their performance or health.

The smartphone also enjoys a strong presence in the MTB domain. (Hansen, The TrailMe System 2019) indicate that upwards of 80 % of riders in Denmark already bring their phone with them on the trail. The platform has experienced significant technological improvements over the last decade especially. Most newer smartphones integrate GPS and advanced sensors for measuring parameters

[1] Report available at https://www.dropbox.com/s/ijyl2t068uxroaj/P8_PPH.pdf?dl=0

such as acceleration, orientation, air pressure and some phones, even vital signs (GSMArena 2019). They are also capable of managing heavy computations and complex tasks, with large multicore processors. Thereby, the platform might represent a cheaper and simpler alternative to some dedicated devices offering like functionality.

The potential of the smartphone as a platform for applications intended for use in sports has not been overlooked. A large variety of solutions for activity tracking, guidance and connecting like-minded sportspeople exist. This includes solutions specifically designed to cater to the MTB sport, for example (Strava 2020, Singletracker 2020). Many of the applications feature activity tracking functions, providing the user with basic parameters, such as speed, distance traveled, session duration and calories burnt (Hansen, The TrailMe System 2019). These features rely mostly on GPS and raw calculations to function, while only implementing a minimal use of the platforms integrated motion sensors. Considering this, the full potential of the smartphones ability to function as an activity tracker might still not have been uncovered, which warrant further examination.

An example of technology designed to improve safety in the MTB sport is described in (Hansen, Smartphone-based, bike crash detection systems for use in extreme sports environments 2019)[2]. The report proposes a smartphone-based solution with the capability of detecting a crash in an MTB environment. The solution makes use of kinematic data collected from the smartphone platforms integrated Inertial Measurement Unit (IMU) to distinguish between events which could be considered harmful to the rider and common "activities of extreme bicycling". The study emphasizes the difficulty in developing incident detection algorithms for use under such extreme conditions, though concludes that it is most likely possible to achieve on the smartphone platform with acceptable results. Yet, multiple similar solutions which require the integration of external, dedicated devices to function are commercially available today (Specialized 2020, Singletracker 2020). While there may be many reasons to develop a dedicated device, such as the possibility of achieving higher accuracy or extended functionality through specialized hardware, it will inevitably add to the cost of the product, and the complexity of the task of developing it.

Several studies on adoption and sustained use of dedicated smart devices offering activity tracking functionalities (Lazar, et al. 2015, Shih, et al. 2015, Canhoto and Arp 2017) conclude that they may be suffering from abandonment issues. Technical shortcomings, such as limited functionality or inaccurate or irrelevant data being collected, have shown to be among the main contributing factors

[2] Report available at https://www.dropbox.com/s/g0qi10lutz8l2kh/P9_PPH.pdf?dl=0

to users neglecting such products, often within short time of purchase. Furthermore, accurate and relevant measurements are essential when designing solutions meant for performance enhancing in sports (Tholander and Nylander 2015). Sports people need reliable and useful data if they are to track and further progression successfully, which makes it an interesting field from a Human Computer Interaction (HCI) perspective. According to (Ericsson ConsumerLab 2016), another factor contributing to users abandoning such devices, arises when functionality requires integration with a smartphone. Moving advanced activity tracking functionalities to the smartphone platform itself may therefore be beneficial since it is already abundantly present in the MTB domain. Though to avoid abandonment issues and provide athletes with useful measurements, it must be able to accurately track activities and provide interesting functionality.

Jumps and drops are probably the two most common obstacles encountered in the MTB domain. Doing jumps and drops are both activities which result in the rider and equipment becoming airborne for an extended time period. This characteristic distinguishes these activities from regular riding, although not from one another. Thereby, the challenge in classifying them lies in distinguishing them from each other. The characteristics of the activities also allow for the deduction of more advanced riding related parameters, such as airtime, jump height and impact forces, which riders might benefit from knowing. (Hansen, Smartphone-based, bike crash detection systems for use in extreme sports environments 2019) showed that it is possible to detect a crash in the MTB domain, though, if the smartphone platform is to have potential as an advanced activity tracker for use in the domain, it must be uncovered if it can also classify riding related activities and provide athletes with beneficial parameters.

To address this, the research questions guiding the making of this thesis were: is the smart phone platform capable of functioning as an advanced activity tracker for use in the MTB sport; and how accurately can it classify activities and derive related parameters compared to a dedicated device? The purpose of the study was to compare the capabilities of the smartphone with a device dedicated to classifying activities in the domain, using pattern recognition machine learning methods. Thereby, the hope was to expand future applications of the smartphone platform in the sport, and possibly lay the foundation for further work in using the platform for more advanced applications within sports, health and HCI research in general.

2 Related work

This chapter presents and discusses the findings of the literature search on sources related to the research questions. Primo, the Aalborg university library search engine, and Google Scholar were the main resources used. Together, they cover a wide range of publishers and databases of academic research literature, ensuring a comprehensive search. Related works found through references was acquired, and chain searches were conducted with keywords discovered to be relevant upon literature review. Search words included: human computer interaction/HCI, sports, mountain biking, smartphone, dedicated device, activity tracking, trick classification, inertial measurement unit/IMU and pattern recognition, in different combinations.

2.1 Existing systems

To identify a candidate suitable for comparison with the smartphone, a search for existing devices capable of advanced activity tracking in the mountain biking sport was conducted. By advanced in this case meaning, that it is capable of detecting and classifying different activities and deriving riding related parameters beyond the basics, such as airtime and impact forces.

Only a single product was found to promote advanced activity tracking capabilities in the MTB domain. The LIT (LIT 2020) is a multiple purpose action sports activity tracker featured on Indiegogo, a crowd funding site. It offers jump recognition and advanced riding related parameters, such as airtime and impact forces, as well as different visualizations of the activities captured, within a range of different sports. Evident from their webpage, it is a watch-like device which might be mounted on the leg while riding MTB for example. As the LIT is a commercial product in the stage of pitching for funding (at the time of writing), it was not possible to acquire a LIT device for these examinations or obtain further technical or academic sources related to the device. Thereby, comparing the smartphones capabilities against a commercially available solution was not an option, and a suitable candidate had to be uncovered in the academic research literature.

2.2 HCI for sports

Literature discovered in the field of Human Computer Interaction for sports is naturally centered almost exclusively on wearable or otherwise highly portable technology. Many studies focus on adoption and sustained use of sports and health related wearables, such as smart watches, Fitbits etc. (Canhoto and Arp 2017, Lazar, et al. 2015, Shih, et al. 2015, Ledger and McCaffrey 2014, Gouveia, Barros and Karapanos 2014). Some studies attempt to devise new approaches to designing wearables for sports. A noteworthy example found in extreme sports is that of (Mencarini, Leonardi, et al. 2018), who describe a method for co-designing wearables for the climbing sport. Other studies concentrate on designing new devices and ways of interacting with them. From a literature review of 57 related papers, (Mencarini, Rapp, et al. 2019) identifies research opportunities within this field: “the investigation of different form factors and types of feedback; the consideration of different sportspeople and collaborative tasks; the need of pushing the boundaries of the sports domain; the exploration of the evolution of sports; the interconnection of different devices; and the increase of methodological rigor” (Mencarini, Rapp, et al. 2019, 314). In attempting to provide a tool for the broader audience to advance and push the boundaries of their abilities and the MTB sport, while simultaneously exploring the possibilities of different form factors as well as the need for the interconnection of devices, this thesis can be said to take advantage of several of these opportunities.

A single study experimenting with mobile devices was discovered within the cycling and mountain biking domain. (Guerra-Rodríguez and Granollers 2016) conducted user experience tests on smartphones and smart watches in outdoor activities, including mountain biking. Comparing the results, they found that smart watches were not preferred over smartphones and reported on several technical issues on both devices experienced by participants in the study, such as loss of connectivity, insufficient battery power and light reflecting in the screen. Though, usability tests overall showed that participants found it easy to execute tasks and feedback was reasonably clear and understandable. These results indicate that the smartphone might represent a viable option for an activity tracker in the MTB domain, in some cases perhaps even preferred over dedicated or otherwise external devices. Additionally, advances in mobile technology in the latest years may have compensated for some of the technical issues reported in the study.

Although the literature in Human Computer Interaction for sports does support and encourage the examination of sports activities through different types of wearable devices, the search for sources on the specific subject of activity tracking in the MTB domain proved unsuccessful.

2.3 Activity tracking

HCI researchers are also occupied with the field of activity tracking in general. (Hansen, Smartphone-based, bike crash detection systems for use in extreme sports environments 2019) identified several useful sources within the field, which overall reported good results. For example, (Nguyen, et al. 2017) achieved a 96.35% accuracy, 95.65% specificity, and 100% sensitivity in detecting falls in the elderly, using a wearable accelerometer motion sensor. Kinematic data collected from the tri-axial accelerometer was analyzed by threshold method to distinguish between activities of daily life (ADLs) and an actual fall. A similar study by (Rungnapakan, Chintakovid and Wuttidittachotti 2018) attempts to detect falls on the smartphone platform, also using a threshold-based algorithm on accelerometer and gyroscope sensor readings, achieving a 97.33% accuracy. Studies in bike crash detection (Williams 2018, Hansen, Smartphone-based, bike crash detection systems for use in extreme sports environments 2019) also show reasonable results with threshold-based algorithms. Such algorithms are possible to implement real-time, taking up only a minimal amount of resources (Nguyen, et al. 2017), which makes them very suitable for mobile devices. Considering the performance and characteristics of threshold algorithms, they may seem to be an attractive solution, although, they are often hardcoded and more suited for detecting impact incidents for example, which might prove too inflexible to stand alone for use in the mountain bike domain.

Several relevant sources were identified upon reviewing literature in sports technology and other related sciences. A large body of works (Baca 2012, Attal, et al. 2015, Migueles, et al. 2017, O'Reilly, et al. 2017) exists in tracking and classifying a wide range of human activities, such as walking, running, lifting heavy objects, and sleeping. Many of these studies experiment with applying more advanced machine learning methods, as for example neural networks, Support Vector Machine (SVM), personal and universal classifiers, k-Nearest Neighbor (k-NN) etc., to accelerometer and sensor fusion data. They also explore the possibilities of achieving it on the smart phone platform and show that such methods are effective and very suited for the purpose of activity tracking on mobile technology in general. Introducing the possibility of adapting to different riders and environments through applying machine learning algorithms would be greatly beneficial in the MTB domain, considering the broad appeal and varied nature of the sport.

2.3.1 Activity tracking for sports

Using similar methods, some studies focused specifically on recognizing certain activities or classifying tricks within a wide range of sports. (Helten, et al. 2011) introduces an approach for achieving automatic segmentation and classification of trampoline jumps, while other researchers have investigated the possibilities of classifying and visualizing skateboard tricks (Groh, Flaschka, et al. 2016, Groh, Fleckenstein, et al. 2017). Another study by (Lee 2015) proposes an algorithm for the real-time detection and classification of different activities, such as jumping and dropping, in the skiing and snowboarding sports. It combines threshold methods and Support Vector Machine for analyzing sensor fusion data, and achieved an 80.5% accuracy, 93% specificity and 92% sensitivity. The study shows that it is possible to automatically differentiate between activities which share similar characteristics, such as jumping and dropping, when combining threshold methods with classification algorithms.

A semester project in sports technology, done on Aalborg university, was the only study identified to have examined similar methods in cycling sports. (Diemar and Hansen 2018) tested three different machine learning methods: decision tree, Support Vector Machine, and k-Nearest Neighbor, in their ability to classify different airborne tricks in the BMX sport, using IMU sensor data. They collected data from prototype dedicated devices integrating standard, low range IMU (+/-16G), as well as a high range IMU (+/-200G), mounted in different places on the bike. With k-Nearest Neighbor having performed best, they achieved a 95-100% accuracy in classifying tricks. The study also showed that it was possible to extract and calculate airtime and impact forces successfully using Principle Component Analysis (PCA). Although, the choice to add a high range IMU had proven necessary to measure the impact forces sometimes experienced in the BMX sport.

Several important aspects were uncovered from reviewing related work. Evident from the large body of research discovered in the field of activity tracking for sports, there is a need for the ability to monitor and classify sports activities accurately, among academics as well as athletes. From the review, the foundation for designing an approach to this thesis was also established. It showed that detecting and classifying different airborne activities using IMU sensor data is achievable with great precision in the BMX sport. Using threshold algorithms to automatically detect flight phases and machine learning methods, such as SVM and k-NN, to classify the transpired activity, it should be possible to achieve similar results in the MTB domain. The literature also provided methods for extracting and calculating riding related parameters, such as airtime and impact forces. Finally, a prototype device dedicated to the purpose of detecting airborne activities and classifying different tricks in the cycling sports, was identified. Although it was originally intended for the BMX sport, the MTB and BMX sports overall share enough characteristics that the capabilities of the device should be transferable by integrating similar hardware components.

3 Methodology

This study was essentially an experiment to determine if the smartphone could perform equally well as a dedicated device at the purpose of advanced activity tracking in the MTB domain. When approached from a Human Computer Interaction perspective (Lazar, Feng and Hochheiser 2017), this can be expressed as a null hypothesis and an alternative hypothesis, the null hypothesis being the one you hope to reject. In the case of this study, the null hypothesis was that the smartphone could not be made to track MTB related activities as well as a dedicated device. A controlled experiment was conducted to determine this, taking a rigorously methodological approach to the examinations. From the literature review, a method for detecting and classifying activities in the cycling sports was identified, which was attempted on both devices implicated in the study. Each of the main constituents of the proposed technique were examined separately to accommodate for multiple devices being tested, and to achieve optimal insight into the results. The condition examined was the relationship between the two different devices' capabilities of detecting and classifying MTB activities and measuring related parameters. Thereby, the experiment relied on quantitative data captured with IMUs in a realistic environment, ensuring that the study reflected the real-world experience.

A micro electromechanical systems inertial measurement unit (MEMS IMU) measures linear acceleration in meters/second^2, along the x, y, and z dimensions (figure 3.1), by means of suspended silicon nano structures (Sparkfun 2020). Thereby, the slightest motion of an object, such as a moving bike, is captured and transduced into an electrical current, which can be stored, analyzed, and visualized on a computer. One way the data might be described, is in g-forces. G-force is a measurement of gravity; 1G is equal to the pull of the earth's gravity at sea-level, or 9.80665 m/s^2. This is reflected in a tri-axial accelerometer, in that the dimension vertically aligned towards the sky, will read a value approximately equal to 1G at sea-level. The easiest way of testing the accuracy of a sensor is to do a calibration test using this constant. If the z axis is oriented towards the sky, it should read approximately 1G (when at sea-level). When doing airborne activities on a bicycle, be it BMX or MTB, you eventually must enter free-fall, shortly experiencing weightlessness. The free-fall state has a distinct pattern of readings consistent with 0G in this axis, in the time that the rider and equipment is airborne (Diemar and Hansen 2018). This pattern is easily recognizable, even using

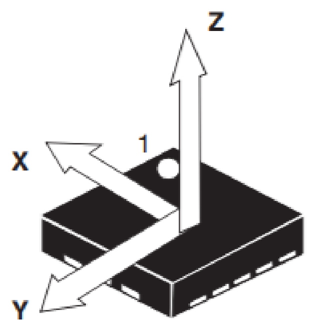

Figure 3.1, source: (Sparkfun 2020).

basic threshold methods, making it possible to detect and classify airborne activities. These are the principles this thesis takes advantage of.

3.1 Materials

The two main materials used for the purpose of the experiment were a prototype dedicated device and a modern, flagship model smartphone. Each device was fitted with customized software for logging data from the IMUs integrated in the platforms. This section describes each device in detail.

3.1.1 Dedicated device

The most promising candidate for a dedicated device discovered upon reviewing related work, was described by (Diemar and Hansen 2018). It was a prototype based on the Shimmer3 IMU development kit (ShimmerSensing 2020), integrated with a high range IMU. They achieved near perfect results in classifying BMX tricks and measuring related parameters. Unfortunately, it was not possible to acquire the same equipment, why it was necessary to design and build a prototype of comparable specifications for the purpose of this thesis.

The prototype devised was based on the development board STM32, model F411RE (STM32 2020), by ST Microelectronics. As evident from the hardware specifications comparison table (table 3.1), it surpasses the specifications of the shimmer3 in all known parameters, and as such was found to be a suitable alternative. (Diemar and Hansen 2018) used an Adafruit ADXL377 accelerometer sensor (Analog devices 2020) for their experiment. It has a maximum 200G range of measurement, which was applied in the study, along with a sample rate of 250Hz (equal to 250 samples/second). The prototype was integrated with a high range Sparkfun H3LIS331DL accelerometer (Sparkfun 2020), capable of measuring forces up to 1000G. While optimizing the prototype, a series of baseline tests showed that the sensor was most accurate when set to 1000G, which was therefore applied for the purpose of the experiment. The H3LIS331DL features a max sample rate of 1KHz, though to best replicate the capabilities of the shimmer3 setup, it was set to 250Hz in this experiment as well. In addition, the prototype was integrated with a standard microSD card reader for data logging, equipped with a 16 GB Scandisk SD card.

Hardware specifications comparison table			
Device:	Shimmer3	F411RE	Samsung Galaxy S8+
CPU	24MHz MSP430	100 MHz ARM Cortex-M4	Octa-core (4x2.3 GHz Mongoose M2 & 4x1.7 GHz Cortex-A53)
Memory	Unknown	512KB Flash, 128KB RAM	4 GB RAM
Storage	SD card, size unknown	SD card, 16 GB	Internal, 64 GB
IMU:	Adafruit ADXL 377	Sparkfun H3LIS331DL	STM LSM6DSL
Sensor type	Analog	Digital	Digital
Max range	200G	1000G	16G
Applied range	200G	400G	8G
Max sample rate	1KHz	1KHz	500Hz (achieved)
Applied sample rate	250Hz	250Hz	250Hz
Filtering	Low pass filter, applied post recording.	high pass filter, coefficient set to 64	Series of high and low pass filters

Table 3.1, sources: (Analog devices 2020, Sparkfun 2020, GSMArena 2019, ShimmerSensing 2020, STM32 2020, Technology Informa 2020)

The software for both devices was designed to only log IMU readings for later processing. As such, the firmware for the prototype dedicated device did not actually run the detection and classification algorithm. The device was simply programmed to acquire readings from the x, y and z dimensions of the integrated IMU in intervals of 4 milliseconds (equal to 250 samples/second), convert the values to G forces and log them to the SD card, pre-fixed with a timestamp (milliseconds since last boot), automatically upon boot up. Although, testing of the prototype showed that introducing further elements or increasing the sampling rate was within the capabilities of the device.

The firmware was composed of 277 lines of code and was written in the arduino IDE (available on Microsoft Store). Although this required for the STM32 Cube Programmer (ST Electronics 2020) to be installed, the arduino platform was chosen over the STM32 native IDE to ease the job of coding the firmware. The prototype implements multithreading for the sensor reading and SD card logging tasks, and model specific support libraries to accommodate for the lack of support in the arduino IDE.

3.1.1.1 Configuration of components

The development process of the prototype device was divided into the three steps: designing, building, and testing. Completing it took approximately 14 days, the time divided 50/50 between constructing the hardware and coding and optimizing the firmware. The hardware components were installed in a black plastic case, large enough to also contain a battery compartment. The compartment was lined with shock absorbing foam, completely incasing the 9-volt battery to avoid unwanted shaking, resulting in false data being captured. Furthermore, the prototype was equipped with a green and a red LED, indicating power on and recording active, respectively. The integrated user button was programmed to safely end the recording and properly save the data. The components were configured as seen in image 3.1:

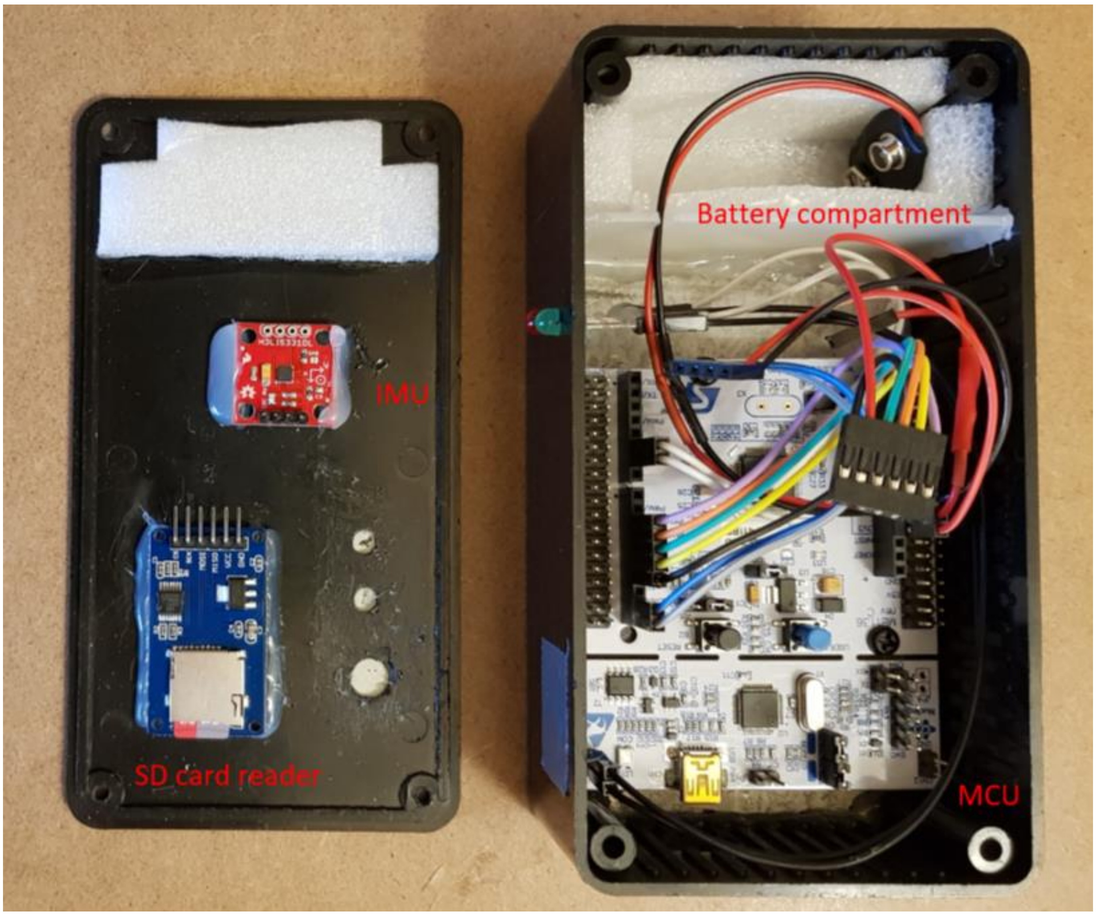

Image 3.1, prototype components configuration.

3.1.1.2 Calibration test

The H3LIS331DL IMU module is factory calibrated to +/-1G, read in from a static memory bank upon every start up, eliminating the need for further calibration (Sparkfun 2020). To accommodate for offset occurring due to post fabrication handling, the module also features an in-built high pass filter. A coefficient of 64, the highest possible, proved the most suitable during testing of the device, and was applied to the high pass filter for the purpose of the examinations. Data loss due to cut off frequency proved non existing at the applied sample rate of 250Hz, which was confirmed visually in validating the timestamps in a series of sample sets. When doing a calibration test on the prototype, as described in section 3, page 12, the sensor proved to perform somewhat as expected. Diagram 3.1 below shows a plot of 2500 samples of raw IMU data captured from the z axis, while the sensor was lying flat on a table with the z axis oriented vertically. The samples had a mean value of 1.36 Gs of vertical acceleration over 10 seconds, and the main part clusters nicely around this value, relative to the +/-1G factory calibration. Though, the signal was found to contain a significant amount of noise, which may be due to several things, such as faulty sensor readings or small vibrations from the surroundings. For this reason, it was necessary to clean up the signal further, applying a filter in the preprocessing phase (see section …).

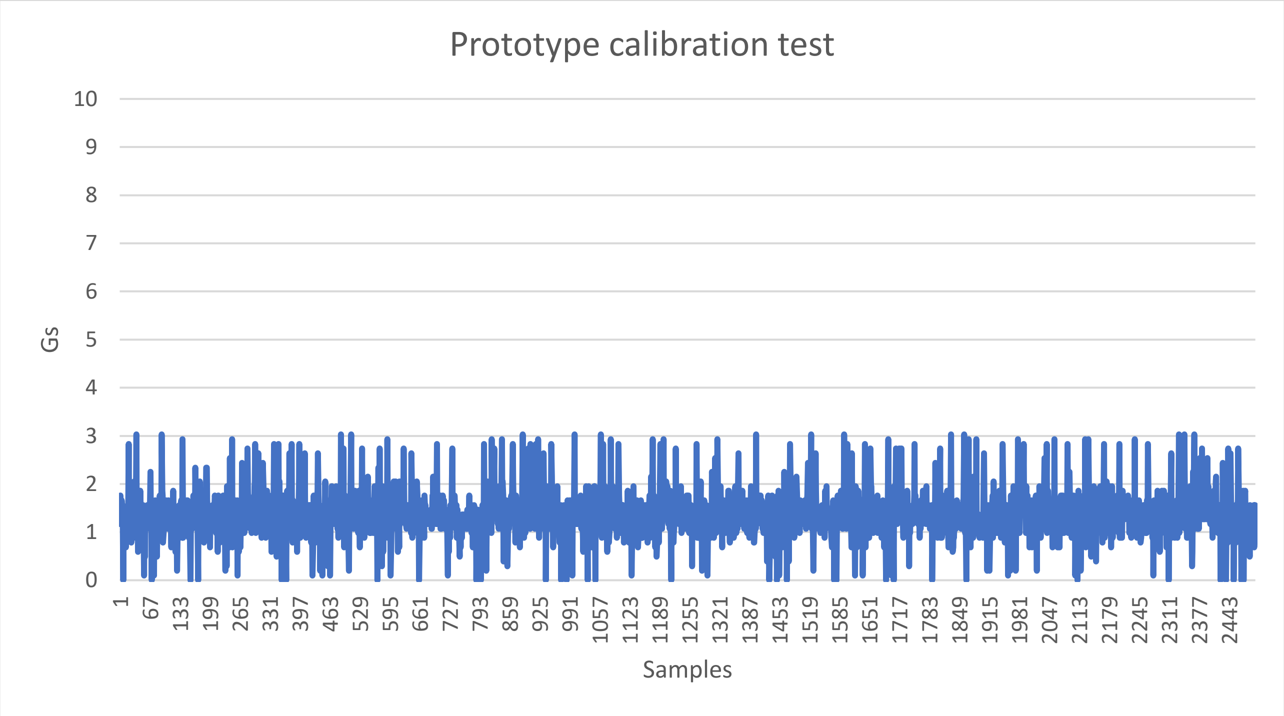

Diagram 3.1, final calibration test, dedicated device. Graph shows gravity of earth captured on the z axis.

3.1.2 Smartphone platform

The smartphone platform used during the experiment was a Samsung Galaxy S8+ (Samsung 2020). As seen in the hardware specifications comparison table (table 3.1), the general specifications of the S8+ far exceed those of the shimmer3 and STM32 setups. In terms of processing power, memory, and internal storage, it has enough resources that it will be able to run the detection and classification algorithms, so that the results may be available immediately or even in real-time. An older flagship model was chosen in an attempt to test a phone with specifications considered common today.

As far as it was possible to uncover (Technology Informa 2020), it features an STM LSM6DS combined accelerometer and gyroscope sensor, which has a max measurement range of 16G. Although, during the testing of the smartphone application, it was discovered that the sensor cuts out at 8G, and efforts to identify a way of setting the range to maximum proved unsuccessful. It is not supported by the Android documentation, nor were any fixes uncovered elsewhere, though, the problem seems to be well known. 8G is probably high even, compared to some older model smartphones. Considering the findings of (Diemar and Hansen 2018), this indicates that the smartphone in general might be unsuitable for measuring some of the riding related parameters. Data recorded during this experiment only included accelerometer readings.

Like the dedicated device, the software was designed to only log data from the IMU. The data was saved on the internal storage of the device, in a folder named “rdl_log”. A timestamp and the x, y, and z values in G forces were the only parameters logged. The app consisted of 151 lines of code and was composed in Android using the official Android IDE. It took about two days to complete, while testing and optimization of the app was done simultaneously with the dedicated device. Although tests showed that the smartphone was capable of logging data with a sampling rate of up to 500Hz, the software was limited to 250Hz to better be able to compare the results, and to achieve a stable logging process. No data loss was registered from any filter cut of frequency, which was confirmed by manually inspecting several datasets.

3.1.2.2 Calibration test

Assuming they are utilized, the LSM6DS accelerometer features a series of high and low pass filters to clean the signal. The accuracy of the sensor was confirmed by doing a calibration test, as with the dedicated device. Diagram 3.2 below shows 2500 samples captured from the z axis on the smart phones integrated IMU. Relative to the dedicated device, this signal is significantly clearer, having all samples clustered in a straight line with a mean value of 1.00 Gs, over a period of 10 seconds. It is unknown if Samsung applies further advanced filtering to the signal, though judging from the baseline, it seems highly probable. In any case, the accuracy of the integrated sensor indicated that it should be adequate for activity detection and trick classification purposes.

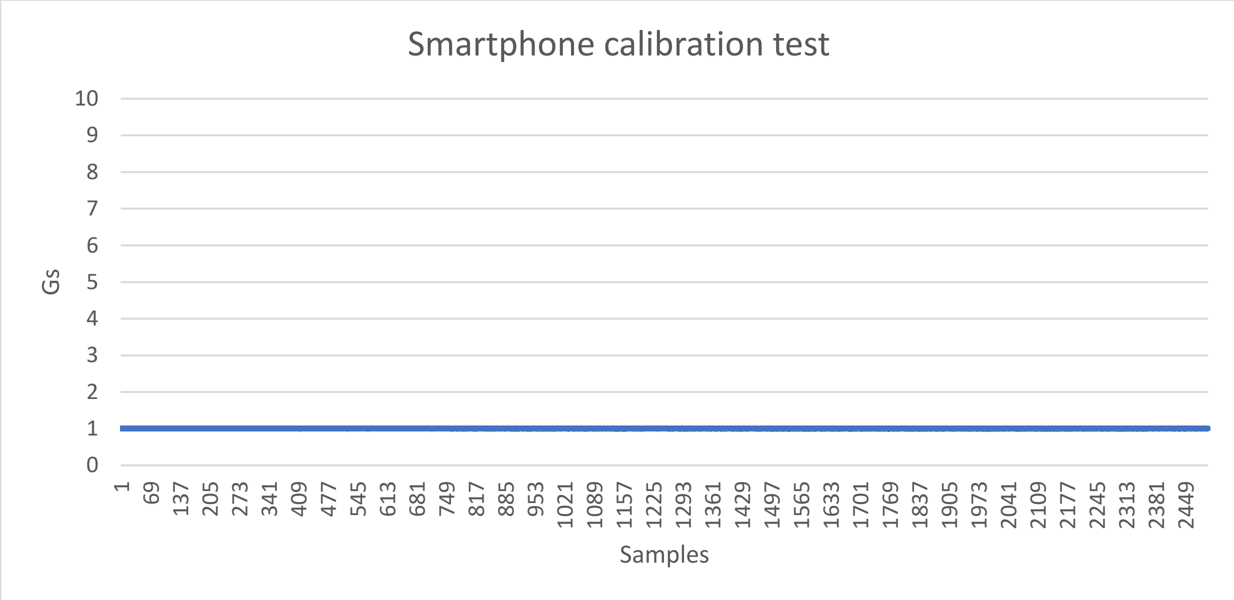

Diagram 3.2, final calibration test, smartphone. Graph shows gravity of earth captured on the z axis.

3.1.3 Bike and placement of devices

A high end, full suspension cross country mountain bike was used for the experiment. It was chosen because it is a common piece of equipment encountered in the sport, and because it was sturdy enough to endure the examinations. The dimensions of the bike also allowed for the devices to be placed in a variety of ways.

A wide variety of phone holders and bags designed to be mounted in various places on the bike are commercially available. Although, often when people use smartphones for activity tracking applications while riding MTB, they have the phone mounted on the handlebar, for example to view live feedback from the system (Hansen, The TrailMe System 2019). A Zéfal Z Console phone holder was therefore chosen to mount the smartphone to the bike handlebar for the purpose of the experiment. It is designed so that it firmly secures the phone to any bar shaped object using rubber elastics. Thereby, it was possible to attach the device in accordance with the norm during the experiment, and thoroughly testing the holder by dropping the bike from different angles showed that it provided adequate stability to avoid significant noise from unintentional movements (image 3.3).

The dedicated device was mounted in front of the saddle, using a piece of foam padding and cable strips as seen in image 3.3. In this study, it was necessary to isolate and examine the vertical acceleration in detail, why both devices were mounted so the z axis was oriented vertically (as seen in image 3.3).

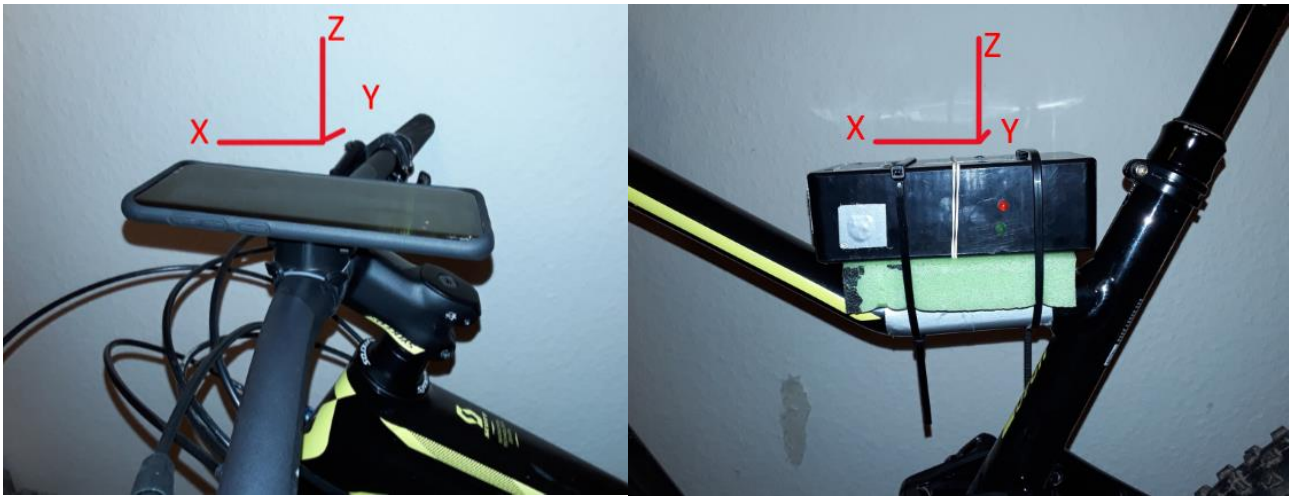

Image 3.3, placement and orientation of devices. Smartphone to the left, dedicated device to the right.

3.2 Setting and participants

The cross country discipline of mountain biking is the most practiced in Denmark (Hansen, The TrailMe System 2019), and perhaps also the basis of a large part of the global commercial market in activity tracking smartphone applications for the MTB sport. Although it varies from rider to rider, it is probably the most relaxed MTB discipline, characterized by small to medium sized obstacles. Considering this, and the limitations of the IMUs in the smart phone platform, the XC discipline was chosen as the focus of this thesis.

The MTB technique and training facilities in Vodskov near Aalborg was chosen as the site of the examinations. It is a facility for training your mountain biking skills, used mostly by the local MTB club. The facility hosts a wide variety of obstacles and terrain types, in a small enclosed area. The features are comparable to obstacles commonly encountered in the cross-country discipline and vary in size from small to large in relation to the XC discipline. It is essentially a small outdoor laboratory featuring all the necessary obstacles, and the conditions for properly executing the activities while securing adequate documentation. This ensured that the examinations could be conducted in a controlled, though realistic setting. To accommodate the nature of the experiment, the largest jump and drop at the facilities were identified as the most suited for the study, and thereby chosen as the obstacles for gathering data. The drop measures 72 cm in height, though the landing is at an angle, so the total drop height is higher. The jump is a so-called tabletop jump measuring 3.5 meters in length and 90 cm in height. The features can be seen in images 3.4 and 3.5 below.

The entire experiment, including data collection, was conducted by the author of the study, mainly due to the reason that the project was done in early 2020, during the Covid-19 epidemic. However, the participant (age 34, weight 88kg, height 182cm) has appr. 6 years of experience in riding amateur XC MTB, ensuring the gathering of relevant, reliable data.

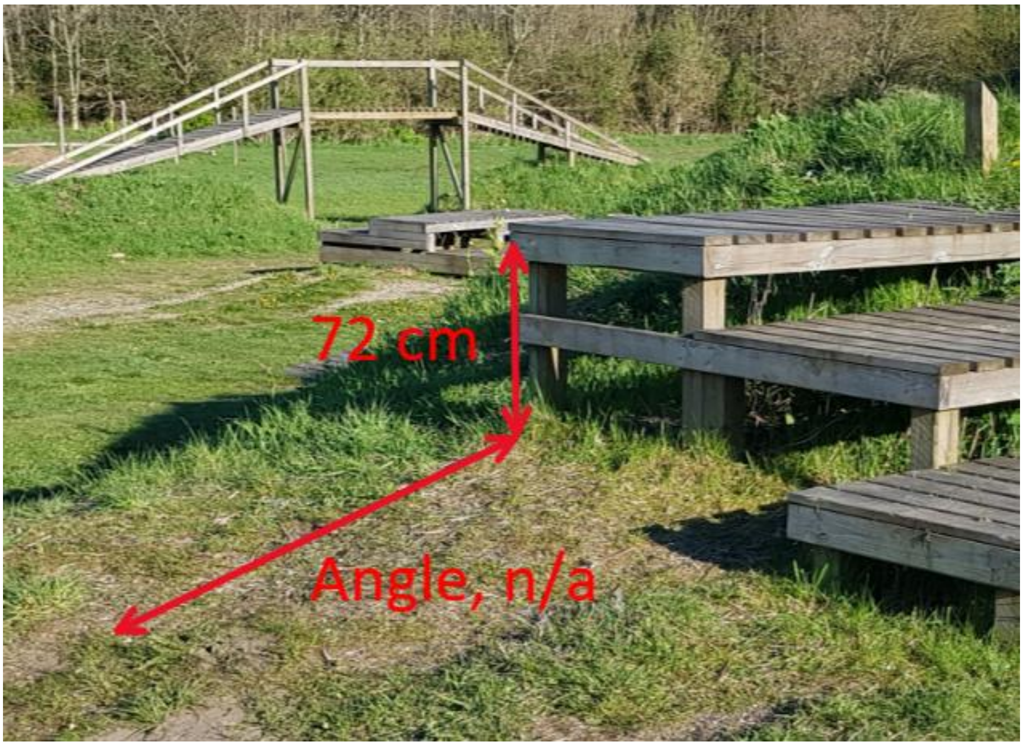

Image 3.4, drop.

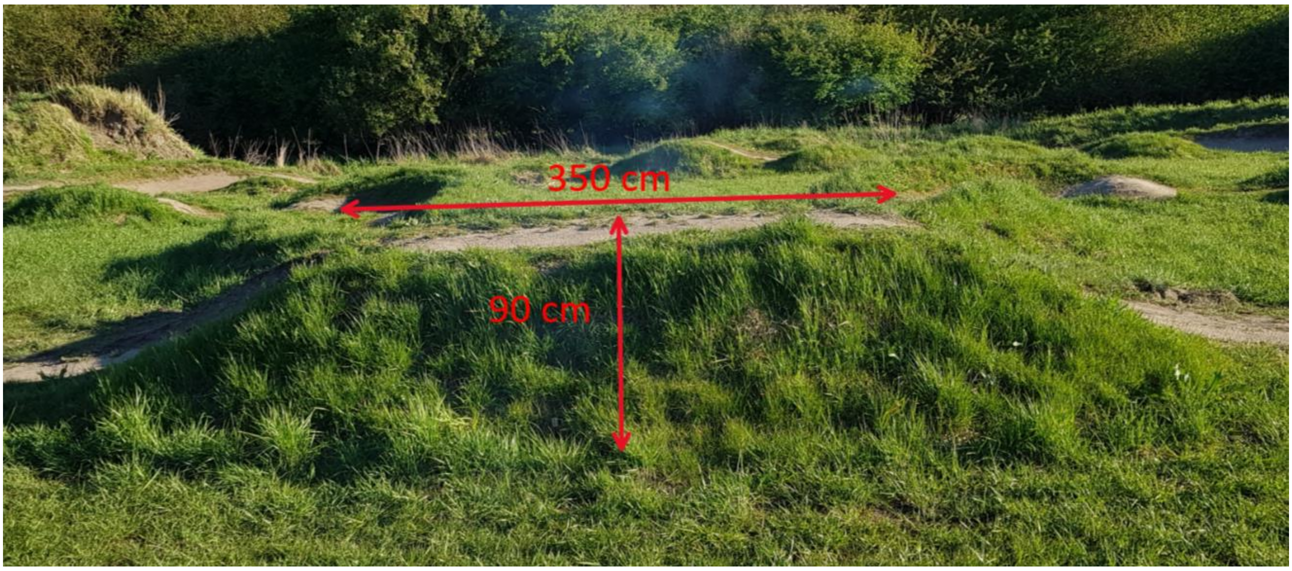

Image 3.5, tabletop jump.

3.3 Data collection

Kinematic data from tri-axial accelerometers was collected through what might be considered an automatic data collection method, intended for measuring the human (Lazar, Feng and Hochheiser 2017). In this case, it was measuring human activity exercised on a mountain bike, more specifically, the G forces that the rider and equipment was subjected to.

The data collection process extended over a couple of days. One session was recorded at the drop location, and two sessions at the jump location. This resulted in the recording of 30 jumps and 30 drops on both devices, amounting to a total of 120 activities captured in 6 separate data sets. All activities were video recorded for later reference and validation of the data. A GoPro Hero 7 Black was used for the purpose of documenting the process. It features high resolution, high speed recording, while offering a wide variety of mounting options. It was set to record at a resolution of 1080*1920, 60 frames/second.

When a session was started, the logging software on both the dedicated device and the smartphone were activated simultaneously, and the entire session was recorded in one stretch. Thereby, the same sessions were captured by both devices, limiting the time spent gathering data. It also resulted in the data being highly uniform in between the devices and different activities, increasing the comparability. When a session was ended, the data was downloaded from the SD card or smartphone to a computer, for safe storage and processing.

3.4 Data analysis

This section presents the overall procedure of the analysis and describes each element in detail.

3.4.1 Procedure

Below chart describes the flow and content of the analysis (figure 4.1). It had three parts in all. First, it was necessary to preprocess the data. This entailed filtering the signal from the devices and segmenting the flight phases to identify the jumps and drops in the data sets. Part 2 of the analysis was the task of classifying the different activities by training a machine learning algorithm to recognize the difference between the two. The last part of the analysis was the parameters calculation. This was done manually on the two parameters: airtime and impact forces.

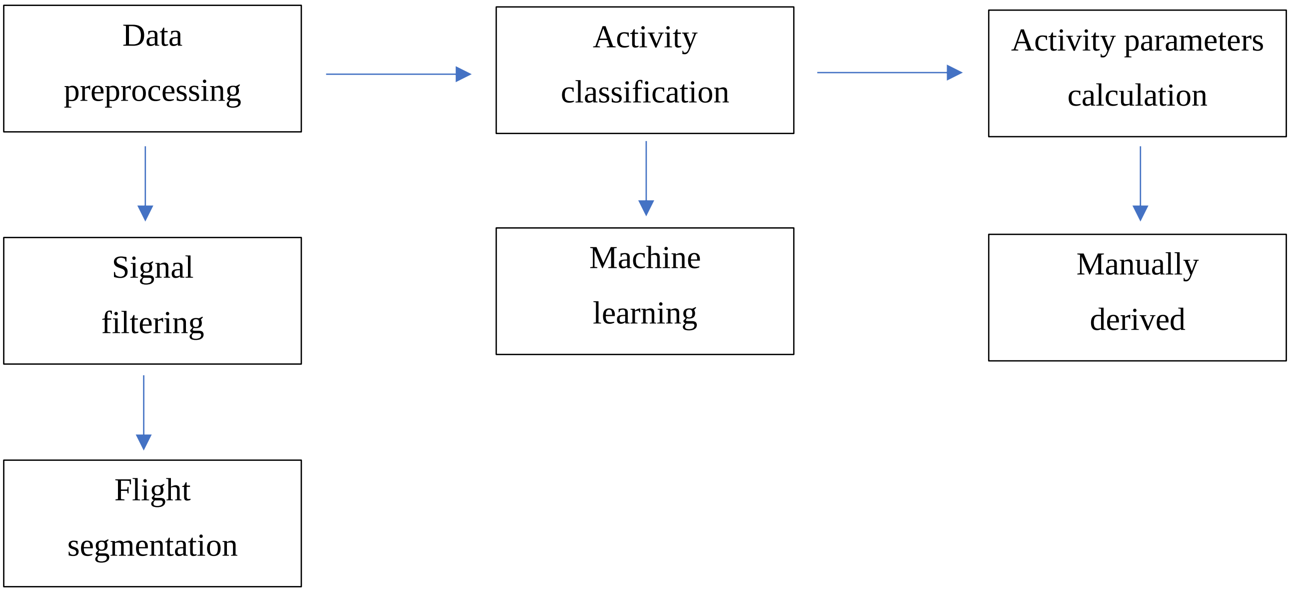

Figure 3.1, analysis flowchart

3.4.2 Data preprocessing

Data preprocessing was necessary to conduct to prepare the raw accelerometer data for analysis with the machine learning algorithms. This part of the analysis involved the two tasks signal filtering and flight segmentation. The different parts are described in detail below.

3.4.2.1 Signal filtering

While the smartphone signal showed signs of already having been filtered in some degree upon recording, the dedicated device was logging the raw, unfiltered XYZ readings. In this study, experimenting with different filter settings showed that applying a lowpass filter to both the devices was beneficial during the flight segmentation phase and the machine learning process.

The filter was designed and applied using the filter designer in MATLAB, which was the main software used for the programmatical task in the project. Both devices had the same filter applied: a lowpass filter made with the equiripple method. To accommodate for the difference in the raw reading of the two devices, the dedicated device had an 8 Hz cut of frequency, while the smartphone signal had 12Hz. Below diagrams (diagrams 3.3-3.6) show the vertical acceleration from a jump and a drop from both devices, in the filtered and unfiltered condition.

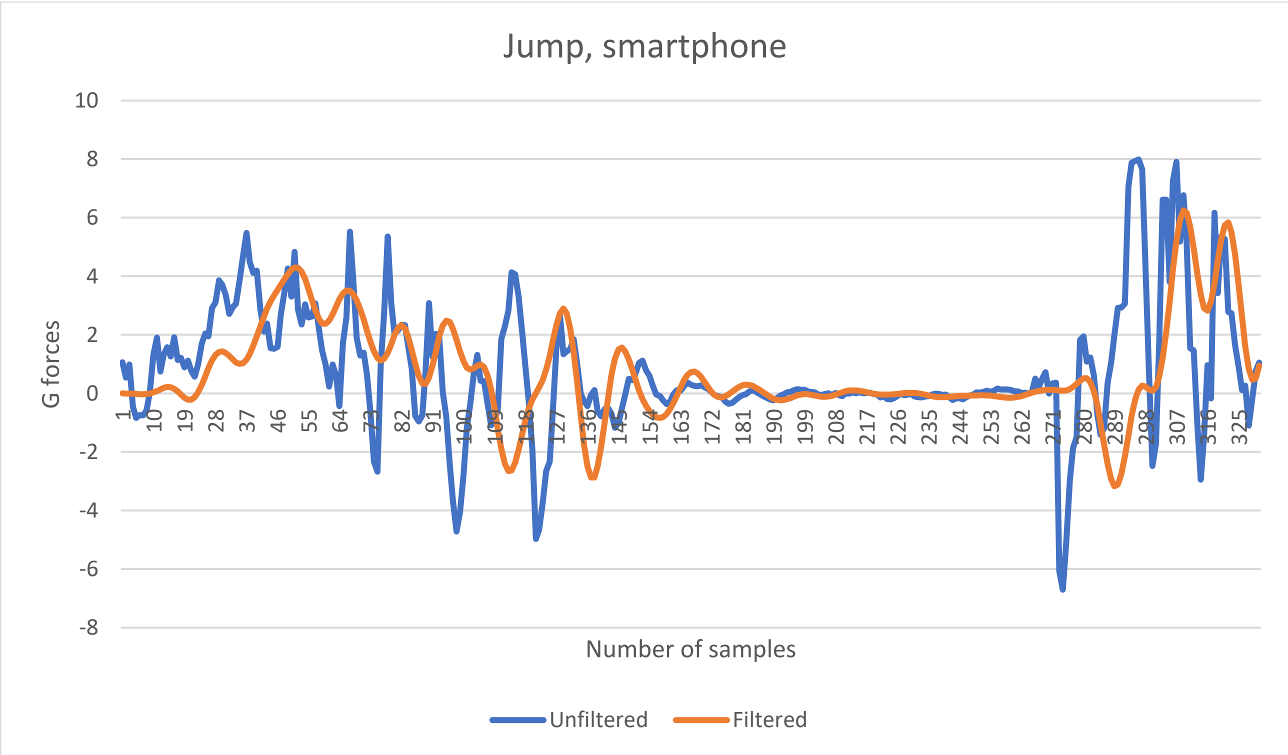

Diagram 3.3, a jump on the smartphone

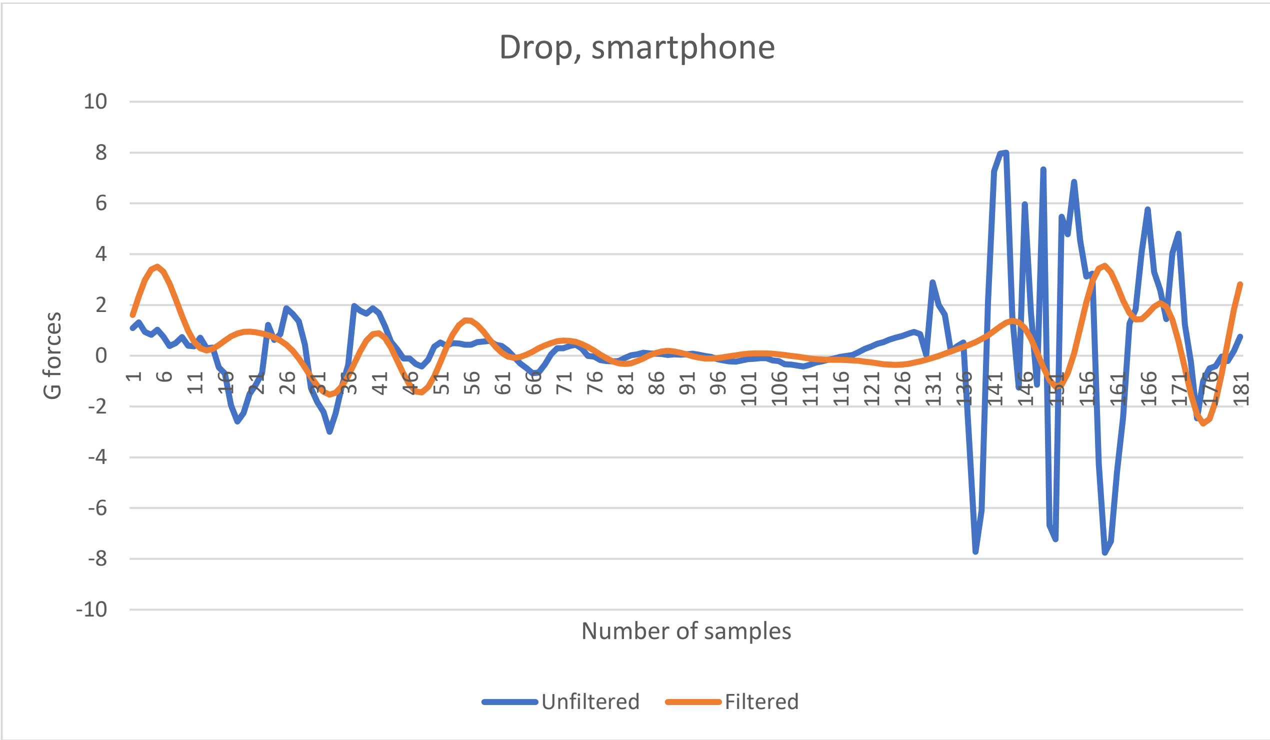

Diagram 3.4, a drop on the smart phone.

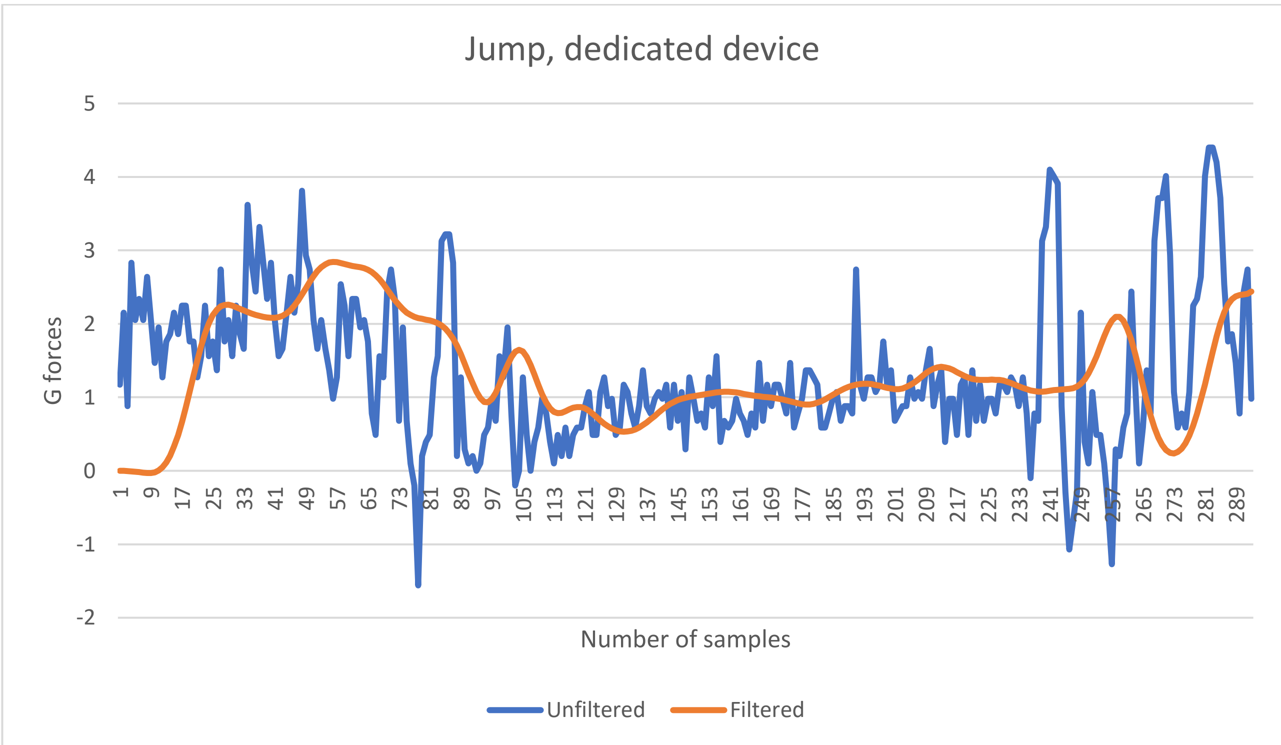

Diagram 3.5, a jump on the dedicated device.

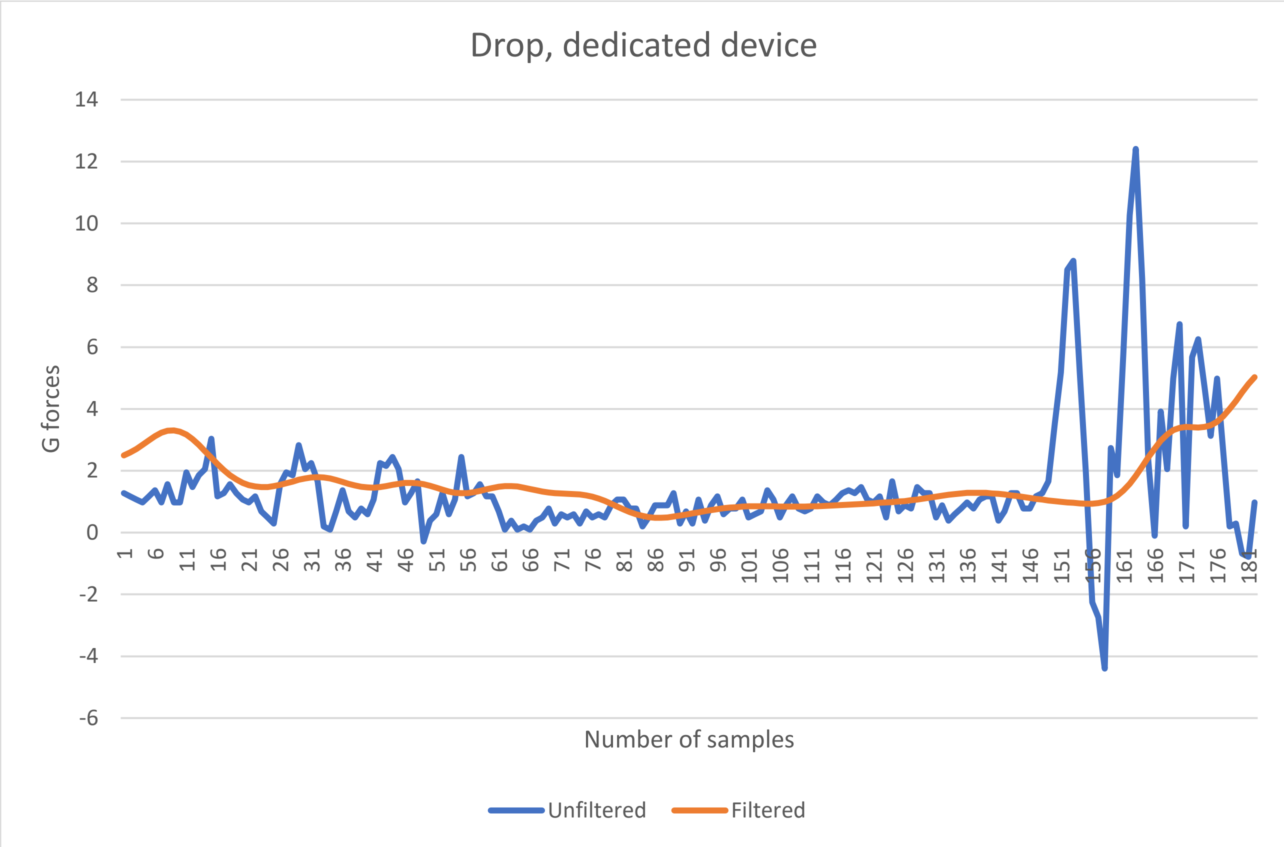

Diagram 3.6, drop on the dedicated device.

As known from the calibration tests (section 3.1.1, diagram 3.1), the dedicated device had a plus margin on the z-axis of 0.37 mean value over 10 seconds, why this must be considered when attempting to segment the flight phases.

3.4.2.2 Flight segmentation

As the airborne activities themselves were the focus of the study, it was necessary to manually segment the flight phases, where the rider and equipment are in a free-fall state, to isolate the activities from the raw data sets for further analysis. The free-fall state can be detected from analyzing the vertical acceleration, as described in section 3.0, why the first step was to isolate and focus on the z-axis of each recorded data set. Since only one activity was performed per session, the class of the activity was known from the session analyzed.

By visual analysis of the video documentation, the start time of each individual jump and drop were noted. Furthermore, the actual airtime was derived from counting frames from the moment the rider and equipment were fully airborne to the precise moment of impact. The video documentation was recorded with 60 frames/second, which is equal to 16.66 milliseconds/frame. For later reference, comments were also made on the execution of the individual jumps and drops.

Using the start time obtained from the video analysis as a reference point, it was possible to identify and isolate all 120 activities successfully. Though, to avoid misleading results from any outside or data related reason, it was necessary to subject the individual activities to a secondary screening process. The criteria for eliminating activities were:

1. Obstacle was not cleared.
2. Corrupt data or false sensor readings from accidental incidents during session.

From reviewing comments and analyzing the data in detail, 6 jumps out of 30 were eliminated from further analysis because the obstacle was not cleared properly. Although the drops were not prone to failed attempts in the same way as jumps, 1 out of 30 drops was eliminated due to false readings, as a result of the dedicated device coming loose during the attempt. Thereby, 24 jumps were identified to be suitable for analysis, while 24 drops were selected randomly from the remaining 29.

The data preprocessing stage ultimately produced four data sets, one set per device per activity.

3.4.3 Activity classification

Machine learning was chosen for the activity classification analysis. (Diemar and Hansen 2018) tested a number of machine learning algorithms in their BMX study, among them is worth mentioning Support Vector Machine and k Nearest Neighbor (kNN). They reported the best results using kNN for the purpose of classifying different tricks in the BMX domain. Therefore, this method was chosen for the method used in this project. kNN attempts to devise a prediction model based on calculations of the distance to the nearest neighbors of samples in a data set (Diemar and Hansen 2018). Thereby, it is possible to predict the right activity from accelerometer data, often with high precision. These characteristics made the method very suitable for use in this study.

The classification learner app in MATLAB was used as a tool for conducting the classification analysis. This ensured that the analysis was easy to perform and that adequate documentation of the process could be secured. The app features several preset classification algorithms, which can be trained and subsequently applied to other data sets. This includes 6 different settings for kNN: fine, medium, coarse, cosine, cubic and weighted. Thereby, these 6 settings were applied to the collected data in a 10-fold validation process. All 6 algorithms were run 10 times on the full XYZ spectrum of the segmented activities from both the dedicated device and the smartphone, to determine which setting could achieve the highest accuracy in classifying jumps and drops. The data sets were divided 66% for training and 33% for validation on each run, and a mean was derived from the results to show which was the best on average. The full analysis can be seen in appendix 1.

3.4.4 Activity parameters calculation

(Diemar and Hansen 2018) identifies two riding related parameters which are possible to derive and validate adequately from the vertical acceleration (z-axis) measured by a MEMS accelerometer alone; airtime and impact forces. Airtime is the time that the rider and equipment are airborne, or in a free fall state, to be precise, while the impact forces are the G forces experienced when landing. These parameters might be beneficial to professional athletes, in because an athlete's ability at completing the activity might be improved from knowing how which technique effects these parameters. Also, the one with the longest airtime earns the bragging rights.

Airtime and impact forces are, as mentioned, possible to derive by extracting the features that activities, such as jumping or dropping, produces in the data. From the physics lecture in section 1.0, we know that when the rider and equipment are airborne the vertical acceleration will approach zero G's. Thereby, it is possible to calculate the airtime by extracting the part of the pattern that has this feature and counting the samples that make up the section. Impact forces can be determined by simply reading out the maximum G-force values in the features produced when landing a jump or drop. Feature extraction and parameter calculation can be done automatically by using Principal Component Analysis (PCA) (Diemar and Hansen 2018) along with the machine learning process, and would ultimately be done as such in a finished algorithm. However, to avoid false readings due to interference from filters and ensure reliable results, the features were extracted manually from the raw z-axis data in this study. The features were extracted as seen in diagrams 3.7 and 3.8 below.

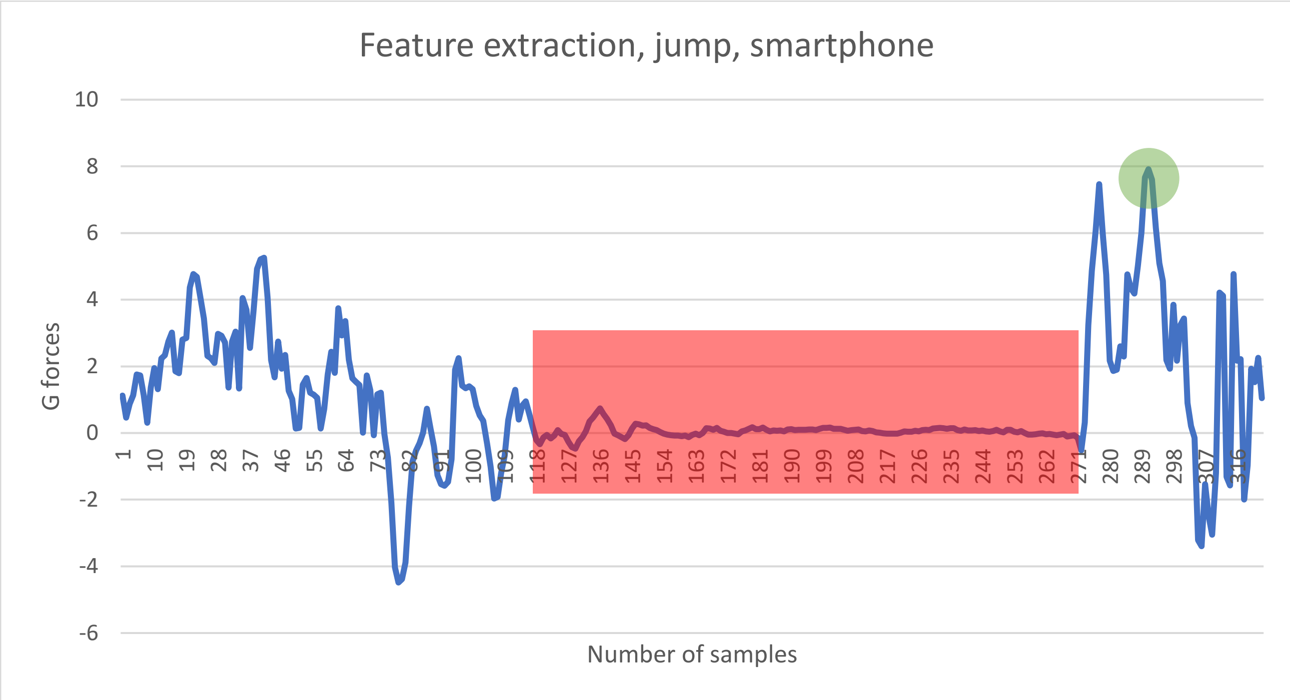

Diagram 3.7, feature extraction from jump, red is the feature related to airtime, green is the impact force.

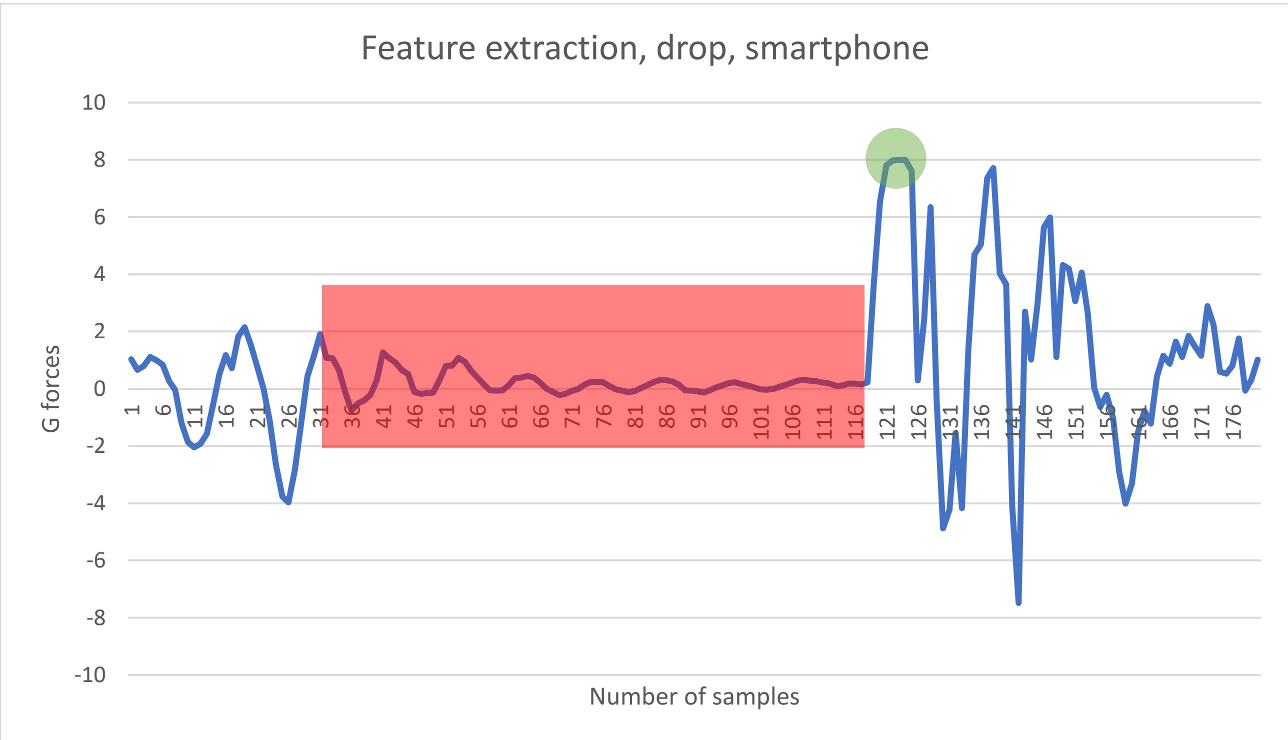

Diagram 3.8, feature extraction from drop, red is the feature related to airtime, green is the impact force.

As with the machine learning activity, a 10-fold validation was conducted, taking 10 random jumps and 10 random drops from both devices, and comparing the results. For further validation, the airtime parameter was also compared to the airtime derived from observations in the video documentation.

4 Results

This chapter presents the results of the activity classification and activity parameters calculation analyses.

4.2 Activity classification

The results of the activity classification analysis can be seen in table 4.1 (dedicated device) and 4.2 (smartphone) below.

	Method					
Run	Fine	Medium	Coarse	Cosine	Cubic	Weighted
1	72.7	76	74.5	68.2	75.9	76.1
2	73.4	77.2	74.1	69.3	77.2	76.7
3	72.7	75.8	73.3	68.2	75.5	75.7
4	73.2	75.8	73.1	67.1	75.6	75.8
5	72.8	74.9	72.6	67.3	75.1	75.4
6	73.6	76.6	73.2	68.9	76.7	76.6
7	73.7	75.3	72.4	67.7	75.2	75.9
8	73.4	75.9	73.7	68.1	75.5	75.7
9	73.7	75.5	72.3	67.8	75.5	76.3
10	72	75.6	73.5	67.8	75.9	75.6
Mean	73.12	75.86	73.27	**68.04**	75.81	**75.98**

Table 4.1, Dedicated device classification analysis results table.

The dedicated device was found to have the highest accuracy in classifying jumps and drops of the two devices, achieving a mean of 75.98% accuracy over 10 runs with the weighted algorithm. In run 2, the dedicated device achieved the maximum accuracy in the experiment, achieving a 77.2% accuracy, with the medium algorithm, which also comes in second in the 10-fold validation, with a mean of 75.86% accuracy. The cosine algorithm had the lowest accuracy, both in terms of the 10-fold validation and in having the lowest accuracy of 64.2%.

	Method					
Run	Fine	Medium	Coarse	Cosine	Cubic	Weighted
1	72	74.6	72.1	66	74.8	75.2
2	70.6	74.6	72.1	66	74.1	74
3	72.1	75.6	72.8	64.4	75.4	75.1
4	70.9	73.8	72	**64.2**	74.3	74.5
5	71.66	75.1	72.8	65.3	75.1	75.4
6	71.2	74.3	72	66.2	74.2	74.2
7	71	75.2	72.6	65.9	75.5	74.7
8	71.2	76	74	65.5	76.1	**75.8**
9	71.1	75.1	73	65.9	75.1	74.9
10	71.7	75	73.7	65.4	75.2	75
Mean	71.35	74.93	72.71	65.48	**74.98**	74.88

Table 4.2, Smartphone classification analysis results table.

The smart phone achieved the highest accuracy of 75.8% with the weighted algorithm, though, over the 10-fold validation, the cubic algorithm proved to have the highest accuracy for the smart phone with a 74.89% accuracy. The lowest scoring was again the Cosine algorithm, achieving only 64.2% accuracy.

Generally, both devices achieved an accuracy of between 65-75% in classifying the two activities, with the worst performance from the cosine algorithm, and the best from medium, cubic, and weighted algorithms.

4.3 Activity parameters

Below are the results of the activity parameters calculations. First, the airtime parameter is detailed, before presenting the impact force parameters.

4.3.1 Airtime

Tables 4.3 and 4.4 below shows the calculated airtime in seconds for the dedicated device, the smartphone, and the observed airtime for 10 randomly chosen jumps and drops.

Jump nr.	Dedicated device	Smartphone	Observed
1	0.53	0.53	0.52
2	0.58	0.56	0.53
3	0.67	0.67	0.60
4	0.60	0.62	0.53
5	0.70	0.55	0.62
6	0.74	0.64	0.67
7	0.58	0.58	0.60
8	0.69	0.59	0.63
9	0.67	0.63	0.67
10	0.68	0.70	0.70

Table 4.3, jump airtime parameter comparison table, value in seconds.

Drop nr.	Dedicated device	Smartphone	Observed
1	0.39	0.37	0.38
2	0.38	0.39	0.40
3	0.41	0.37	0.40
4	0.36	0.39	0.37
5	0.37	0.37	0.35
6	0.38	0.34	0.37
7	0.40	0.37	0.40
8	0.38	0.36	0.37
9	0.35	0.35	0.37
10	0.38	0.36	0.37

Table 4.4, drop airtime parameter comparison table, value in seconds.

As evident from the jump airtime parameter comparison table, both devices show good and bad results. In 5/10 instances the devices are each off by a lot, as for example in jump 3, where they are both wrong by 0.07 seconds relative to the observed airtime. Although, when considering that this amounts to less than a 1/10 of a second, it is still rather close. The devices also show good results. In

3/10 instances, the extracted features were remarkably close to the observed airtime. In jump nr. 1 for example, both devices came within 0.01 seconds of the observed airtime, and 0.02 seconds in jump 7. The dedicated device was correct in one instance, jump nr, 9, while the smartphone was right in jump 10. Across the devices, the airtime calculated from the feature extraction was generally within 0.1 seconds of the observed airtime, except for a single instance for both devices (marked in red)

As seen in the drop airtime parameter comparison table (table 4.4), the devices perform great in determining airtime in drops. Both the dedicated device and the smartphone are generally within 0.01-0.02 seconds of the observed airtime.

4.3.2 G- forces

Table 3.5 below shows the derived impact forces from 10 Random jumps and 10 random drops.

Dedicated device		Smartphone	
Jumps	Drops	Jumps	Drops
4.4	12.4	8.0	8.0
5.3	8.7	8.0	8.0
4.4	10.4	8.0	8.0
6.1	9.4	8.0	8.0
5.1	13.0	8.0	8.0
7.3	9.5	8.0	8.0
5.4	9.4	8.0	8.0
4.4	12.4	8.0	8.0
4.4	11.1	7.9	8.0
5.2	9.4	8.0	8.0

Table 3.5, Impact force comparison table, values in G forces.

When looking at the impact force comparison table, it is obvious that the smart phone was not very well suited for measuring impact forces. It measured 8.0 Gs in all instances, both in jumps and drops. The dedicated device on the other hand had a more realistic spread. It shows impact forces below 8

Gs in all instances of jumps, and impact forces between 8.7-13 Gs in drops. The fact that the smart phone cuts out at 8 Gs was established already in the beginning of the study, although now it is clear that it is not especially suited at measuring impact forces in the MTB domain. Although the dedicated device shows G forces below 8 Gs in jumps, the smart phone maxes out here as well. this might be due to the placement of the device though. The smart phone is mounted on the handlebar which might experience higher G forces then the dedicated device which is mounted in the middle of the bike.

5 Discussion

This chapter will discuss the findings of the study. First, the activity classification analysis is discussed before moving on to the activity parameters calculation. Along the way the two devices capabilities of functioning as an advanced activity tracker in the MTB domain are compared.

5.1 Activity classification

When considering the results of the activity classification analysis, neither of the devices perform at the same level as some of the systems presented in related work. Many of these studies (Diemar and Hansen 2018, Lee 2015) achieve accuracy levels of over 90%, while the mean of the two devices tested in this study was around 75%. This might be due to the fact that only accelerometer data was included in this experiment. The accuracy might be improved by including more sensors such as magnetometer and gyroscope in further research.

When comparing the two devices the smart phone and the dedicated device was found to perform more or less equally well in classifying drops and jumps in the MTB domain. Both devices achieved accuracy levels, in the 70% range, and the dedicated device achieved a 77.2% max accuracy, which is a relatively good result considering the low number of predictors (sensor readings) loaded into the model.

Through the study, traits like jumps and drops was encountered in regular riding, so some kind of detection algorithm should be in place to filter for false positives. Threshold or otherwise, it could be two part, including spectrum analysis, as in (Diemar and Hansen 2018).

5.2 Activity parameters

The activity parameters calculation analysis showed that both the dedicated device and smart phone were able to produce signals of high enough quality and sample rate to extract features for calculating riding related parameters acceptably. For example, in calculating airtime, it was found that both devices were accurate within a tenth of a second across a 10-fold validation, especially when extracting features from drops. The derived airtime for both devices were comparable to that of (Diemar and Hansen 2018), Which indicates that the dedicated device, as well as the smartphone, would perform reasonably acceptable in extracting features for airtime calculation using principle component analysis also.

When considering the results of the impact forces parameter analysis, the smart phone might not be suited for impact detection in the MTB domain. The fact that the smartphone cuts out at 8 Gs in all activities, even when the dedicated device shows impact forces below 8 Gs, cements this fact.

The activities eliminated from the study may also hold valuable information. It may be possible to detect when you cap a jump with the back wheel or make other mistakes, such as low/high nose, these show up in the data with a clear pattern, which might be useful to athletes.

5.1 Limitations of the study

It should be necessary to record and teach the algorithm other kinds of jumps and drops. In a sense, the algorithm has only been thought to recognize activities performed on the two target obstacles. It may recognize activities performed on other obstacles of similar characteristics, though the diverse nature of MTB obstacles taken into consideration, it should be introduced to other patterns recorded on various obstacle. The study does not take into account that some rider might do tricks over obstacles such as jumps and drops either, which should be addressed in a finished product.

The study does not concern the direct interaction between the user and UI components, it is only an exploratory study (basic research) into the possibilities of achieving advanced activity tracking in the MTB domain.

6 Conclusion

Although the study did not achieve the same accuracy as other studies in the area, it's fairly safe to say that it shows that this smart phone can perform equally well as a dedicated device for the purpose of advanced activity tracking in the MTB domain.

across the 10 fold validation the devices showed accuracy levels of around 75% which could be better.

the parameters calculation analysis showed that the smart phone can perform equally well in deriving airtime from feature extraction than a dedicated device.

7 References

Analog devices. »Adafuit ADXL 377 datasheet.« *Anolog.com.* 01. 05 2020. https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL377.pdf (senest hentet eller vist den 01. 05 2020).

Attal, Ferhat, Samer Mohammed, Mariam Dedabrishvili, Faicel Chamroukhi, Latifa Oukhellou, og Yacine Amirat. »Physical Human Activity Recognition Using Wearable Sensor.« *Sensors, nr. 15*, 2015.

Baca, Arnold. »Methods for Recognition and Classification of Human Motion Patterns – A Prerequisite for Intelligent Devices Assisting in Sports Activities.« *IFAC Proceedings, nr. 45*, 2012: 55-61.

Canhoto, Ana I, and Sabrina Arp. "Exploring the factors that support adoption and sustained use of health and fitness wearables." *Journal of Marketing Management*, 2017: 32-60.

Diemar, Jacob Guy, and Mikkel Jul Hansen. "Classification of BMX tricks using Inertial Measurement Units." 2018.

Ericsson ConsumerLab. *Wearable technology and internet of things.* Ericsson, 2016.

Garmin. *garmin 1030 edge* . 04 28, 2020. https://buy.garmin.com/en-US/US/p/567991/pn/010-01758-00 (accessed 04 28, 2020).

Gouveia, R, S Barros, and E Karapanos. "Understanding users’ disengagement with wearable activity trackers." *Proceedings of the Workshops on Advances in Computer Entertainment Conference – ACE*, 2014: 1-3.

Groh, Benjamin H, Jasmin Flaschka, Markus Wirth, Thomas Kautz, Martin Fleckenstein, og Bjoern M Eskofi. »Wearable Real-Time Skateboard Trick Visualization and Its Community Perception.« 2016.

Groh, Benjamin H, Martin Fleckenstein, Thomas Kautz, og Bjoern M Eskofier. »Classification and visualization of skateboard tricks using wearable sensors.« *Pervasive and Mobile Computing*, 2017: 42-55.

GSMArena. *S8+ Specs.* 03. 01 2019. https://www.gsmarena.com/samsung_galaxy_s8+-8523.php.

Guerra-Rodríguez, David, and Antoni Granollers. "User experience experiments with mobile devices in outdoors activities. Use case: cycling and mountain biking." *ACM*, 2016.

Hansen, Palle Preben. »Smartphone-based, bike crash detection systems for use in extreme sports environments.« Aalborg, 2019.

Hansen, Palle Preben. »The TrailMe System.« Aalborg, 2019.

Helten, Thomas, Heike Brock, Meinard Müller, og Hans-Peter Seidel. »Classification of trampoline jumps using inertial sensors.« *Sports Eng, nr. 14*, 2011: 155-164.

James, Daniel A, and Nicola Petrone. *Sensors and Wearable Technologies in Sport Technologies, Trends and Approaches for Implementation.* Springer, 2016.

Lazar, Amanda, Joshua Tanenbaum, Christian Koehler, og David H Nguyen. »Why We Use and Abandon Smart Devices.« *UbiComp*, 2015.

Lazar, Jonathan, Jinjuan Heidi Feng, and Harry Hochheiser. *Research Methods in Human-Computer Interaction.* Morgan Kaufmann publishers, 2017.

Ledger, D, and D McCaffrey. "Inside wearables: How the science of human behavior change offers the secret to long-term engagement." 2014.

Lee, Tien Jung. »Feature Extraction and Classification of Skiing/Snowboarding Jumps with an Integrated Head-mounted Sensor .« 2015.

LIT. *Indiegogo/LIT.* 04 28, 2020. https://www.indiegogo.com/projects/lit-an-activity-tracker-ready-for-action/x/23525494#/ (accessed 04 28, 2020).

Mencarini, Eleonora, Amon Rapp, Lia Tirabeni, and Massimo Zancanaro. "Designing Wearable Systems for Sports: A Review of Trends and Opportunities in Human–Computer Interaction." *IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS*, august 2019: 314-325.

Mencarini, Eleonora, Chiara Leonardi, Alessandro Cappelletti, Davide Giovanelli, Antonella DeAngeli, og Massimo Zancanaro. »Co-designing wearable devices for sports: The case study of sport climbing.« *International Journal of Human-Computer Studies 124*, 2018: 26-43.

Migueles, Jairo H, et al. »Accelerometer Data Collection and Processing Criteria to Assess Physical Activity and Other Outcomes: A Systematic Review and Practical Considerations.« *Sports Med, nr. 47*, 2017: 1821-1845.

Nguyen, Hoa, Asif M Naeem, Farhaan Mirza, and Mirza M Baig. "Detecting Falls Using a Wearable Accelerometer Motion Sensor." *MobiQuitous*, 11 10, 2017.

O'Reilly, Martin A, Darragh F Whelana, Tomas E Ward, Eamonn Delahunt, og Brian M Caulfield. »Classification of deadlift biomechanics with wearable inertial measurement units.« *Journal of Biomechanics, nr 58*, 2017: 155-161.

Rungnapakan, Traitot, Thippaya Chintakovid, and Pongpisi Wuttidittachotti. "Fall Detection Using Accelerometer, Gyroscope & Impact Force Calculation on Android Smartphones." *Proceedings of CHIuXiD 2018 – The 4th ACM In Cooperation International Conference in HCI and UX*, 03 29, 2018: 49-53.

Samsung. *S8+ specs.* 01. 05 2020. https://www.samsung.com/global/galaxy/galaxy-s8/specs/ (senest hentet eller vist den 01. 05 2020).

Shih, Patrick C, Kyungsik Han, Erika S Poole, Mary Beth Rosson, og John M Carroll. »Use and Adoption Challenges of Wearable Activity Trackers.« *iConference*, 2015.

ShimmerSensing. *Consensys IMU Development Kits.* 01. 05 2020. http://www.shimmersensing.com/products/shimmer3-development-kit (senest hentet eller vist den 01. 05 2020).

Singletracker. *Singletracker.dk.* 01. 05 2020. https://singletracker.dk/.

—. *TILT DETECTION SYSTEM – DIN SIKKERHED PÅ SPORET.* 12. 01 2020. https://blog.singletracker.dk/index.php/2017/07/05/tilt-detection-system-din-sikkerhed-paa-sporet/.

Sparkfun. »H3LIS331DL datasheet.« *Sparkfun.com.* 01. 05 2020. https://cdn.sparkfun.com/assets/c/6/5/8/d/en.DM00053090.pdf (senest hentet eller vist den 01. 05 2020).

Specialized. *Angi.* 12. 01 2020. https://www.specialized.com/dk/da/stories/angi.

ST Electronics. *STM32CubeProg.* 01. 05 2020. https://www.st.com/en/development-tools/stm32cubeprog.html (senest hentet eller vist den 01. 05 2020).

STM32. »STM32F411RE .« *st.com.* 01. 05 2020. https://www.st.com/en/microcontrollers-microprocessors/stm32f411re.html (senest hentet eller vist den 01. 05 2020).

Strava. *Strava.* 01. 05 2020. https://www.strava.com/.

Technology Informa. »samsung galaxy s8.« *technology.informa.com.* 01. 05 2020. https://technology.informa.com/api/binary/592077 (senest hentet eller vist den 01. 05 2020).

Tholander, Jackb, and Stina Nylander. "Snot, Sweat, Pain, Mud, and Snow — Performance and Experience in the Use of Sports Watches." *Wellness and wearables*, 2015.

Williams, B. *BICYCLE CRASH DETECTION: USING A VOICE-ASSISTANT FOR MORE ACCURATE REPORTING.* Proquest, 2018.

8 Appendices

8.1 Classification analysis, dedicated device

1:

1.1	KNN	Last change: Fine KNN	Accuracy: 72.7% 3/3 features
1.2	KNN	Last change: Medium KNN	Accuracy: 76.0% 3/3 features
1.3	KNN	Last change: Coarse KNN	Accuracy: 74.5% 3/3 features
1.4	KNN	Last change: Cosine KNN	Accuracy: 68.2% 3/3 features
1.5	KNN	Last change: Cubic KNN	Accuracy: 75.9% 3/3 features
1.6	KNN	Last change: Weighted KNN	Accuracy: **76.1%** 3/3 features

2:

1.1	KNN	Last change: Fine KNN	Accuracy: 73.4% 3/3 features
1.2	KNN	Last change: Medium KNN	Accuracy: **77.2%** 3/3 features
1.3	KNN	Last change: Coarse KNN	Accuracy: 74.1% 3/3 features
1.4	KNN	Last change: Cosine KNN	Accuracy: 69.3% 3/3 features
1.5	KNN	Last change: Cubic KNN	Accuracy: 77.2% 3/3 features
1.6	KNN	Last change: Weighted KNN	Accuracy: 76.7% 3/3 features

3:

Model	Last change	Accuracy	Features
1.1 KNN	Fine KNN	72.7%	3/3 features
1.2 KNN	Medium KNN	**75.8%**	3/3 features
1.3 KNN	Coarse KNN	73.3%	3/3 features
1.4 KNN	Cosine KNN	68.2%	3/3 features
1.5 KNN	Cubic KNN	75.5%	3/3 features
1.6 KNN	Weighted KNN	75.7%	3/3 features

4:

Model	Last change	Accuracy	Features
1.1 KNN	Fine KNN	73.2%	3/3 features
1.2 KNN	Medium KNN	75.8%	3/3 features
1.3 KNN	Coarse KNN	73.1%	3/3 features
1.4 KNN	Cosine KNN	67.1%	3/3 features
1.5 KNN	Cubic KNN	75.6%	3/3 features
1.6 KNN	Weighted KNN	**75.8%**	3/3 features

5:

1.1 ☆ KNN	Last change: Fine KNN	Accuracy: 72.8% 3/3 features
1.2 ☆ KNN	Last change: Medium KNN	Accuracy: 74.9% 3/3 features
1.3 ☆ KNN	Last change: Coarse KNN	Accuracy: 72.6% 3/3 features
1.4 ☆ KNN	Last change: Cosine KNN	Accuracy: 67.3% 3/3 features
1.5 ☆ KNN	Last change: Cubic KNN	Accuracy: 75.1% 3/3 features
1.6 ☆ KNN	Last change: Weighted KNN	Accuracy: **75.4%** 3/3 features

6:

1.1 ☆ KNN	Last change: Fine KNN	Accuracy: 73.6% 3/3 features
1.2 ☆ KNN	Last change: Medium KNN	Accuracy: 76.6% 3/3 features
1.3 ☆ KNN	Last change: Coarse KNN	Accuracy: 73.2% 3/3 features
1.4 ☆ KNN	Last change: Cosine KNN	Accuracy: 68.9% 3/3 features
1.5 ☆ KNN	Last change: Cubic KNN	Accuracy: **76.7%** 3/3 features
1.6 ☆ KNN	Last change: Weighted KNN	Accuracy: 76.6% 3/3 features

7:

Model	Type	Last change	Accuracy	Features
1.1	KNN	Fine KNN	73.7%	3/3 features
1.2	KNN	Medium KNN	75.3%	3/3 features
1.3	KNN	Coarse KNN	72.4%	3/3 features
1.4	KNN	Cosine KNN	67.7%	3/3 features
1.5	KNN	Cubic KNN	75.2%	3/3 features
1.6	KNN	Weighted KNN	**75.9%**	3/3 features

8:

Model	Type	Last change	Accuracy	Features
1.1	KNN	Fine KNN	73.4%	3/3 features
1.2	KNN	Medium KNN	**75.9%**	3/3 features
1.3	KNN	Coarse KNN	73.7%	3/3 features
1.4	KNN	Cosine KNN	68.1%	3/3 features
1.5	KNN	Cubic KNN	75.5%	3/3 features
1.6	KNN	Weighted KNN	75.7%	3/3 features

9:

Model	Last change	Accuracy	Features
1.1 KNN	Fine KNN	73.7%	3/3 features
1.2 KNN	Medium KNN	75.5%	3/3 features
1.3 KNN	Coarse KNN	72.3%	3/3 features
1.4 KNN	Cosine KNN	67.8%	3/3 features
1.5 KNN	Cubic KNN	75.5%	3/3 features
1.6 KNN	Weighted KNN	**76.3%**	3/3 features

10:

Model	Last change	Accuracy	Features
1.1 KNN	Fine KNN	72.0%	3/3 features
1.2 KNN	Medium KNN	75.6%	3/3 features
1.3 KNN	Coarse KNN	73.5%	3/3 features
1.4 KNN	Cosine KNN	67.8%	3/3 features
1.5 KNN	Cubic KNN	**75.9%**	3/3 features
1.6 KNN	Weighted KNN	75.6%	3/3 features

8.1 Classification analysis, smartphone

1:

1.1 KNN	Last change: Fine KNN	Accuracy: 72.0% 3/3 features
1.2 KNN	Last change: Medium KNN	Accuracy: 74.6% 3/3 features
1.3 KNN	Last change: Coarse KNN	Accuracy: 72.1% 3/3 features
1.4 KNN	Last change: Cosine KNN	Accuracy: 66.0% 3/3 features
1.5 KNN	Last change: Cubic KNN	Accuracy: 74.8% 3/3 features
1.6 KNN	Last change: Weighted KNN	Accuracy: **75.2%** 3/3 features

2:

1.1 KNN	Last change: Fine KNN	Accuracy: 70.6% 3/3 features
1.2 KNN	Last change: Medium KNN	Accuracy: **74.6%** 3/3 features
1.3 KNN	Last change: Coarse KNN	Accuracy: 72.1% 3/3 features
1.4 KNN	Last change: Cosine KNN	Accuracy: 66.0% 3/3 features
1.5 KNN	Last change: Cubic KNN	Accuracy: 74.1% 3/3 features
1.6 KNN	Last change: Weighted KNN	Accuracy: 74.0% 3/3 features

3:

1.1	KNN Last change: Fine KNN	Accuracy: 72.1% 3/3 features
1.2	KNN Last change: Medium KNN	Accuracy: **75.6%** 3/3 features
1.3	KNN Last change: Coarse KNN	Accuracy: 72.8% 3/3 features
1.4	KNN Last change: Cosine KNN	Accuracy: 64.4% 3/3 features
1.5	KNN Last change: Cubic KNN	Accuracy: 75.4% 3/3 features
1.6	KNN Last change: Weighted KNN	Accuracy: 75.1% 3/3 features

4:

1.1	KNN Last change: Fine KNN	Accuracy: 70.9% 3/3 features
1.2	KNN Last change: Medium KNN	Accuracy: 73.8% 3/3 features
1.3	KNN Last change: Coarse KNN	Accuracy: 72.0% 3/3 features
1.4	KNN Last change: Cosine KNN	Accuracy: 64.2% 3/3 features
1.5	KNN Last change: Cubic KNN	Accuracy: 74.3% 3/3 features
1.6	KNN Last change: Weighted KNN	Accuracy: **74.5%** 3/3 features

5:

1.1	KNN Last change: Fine KNN	Accuracy: 71.6%	3/3 features
1.2	KNN Last change: Medium KNN	Accuracy: 75.1%	3/3 features
1.3	KNN Last change: Coarse KNN	Accuracy: 72.8%	3/3 features
1.4	KNN Last change: Cosine KNN	Accuracy: 65.3%	3/3 features
1.5	KNN Last change: Cubic KNN	Accuracy: 75.1%	3/3 features
1.6	KNN Last change: Weighted KNN	Accuracy: **75.4%**	3/3 features

6:

1.1	KNN Last change: Fine KNN	Accuracy: 71.2%	3/3 features
1.2	KNN Last change: Medium KNN	Accuracy: **74.3%**	3/3 features
1.3	KNN Last change: Coarse KNN	Accuracy: 72.0%	3/3 features
1.4	KNN Last change: Cosine KNN	Accuracy: 66.2%	3/3 features
1.5	KNN Last change: Cubic KNN	Accuracy: 74.2%	3/3 features
1.6	KNN Last change: Weighted KNN	Accuracy: 74.2%	3/3 features

7:

Model	Last change	Accuracy	Features
1.1 KNN	Fine KNN	71.0%	3/3 features
1.2 KNN	Medium KNN	75.2%	3/3 features
1.3 KNN	Coarse KNN	72.6%	3/3 features
1.4 KNN	Cosine KNN	65.9%	3/3 features
1.5 KNN	Cubic KNN	**75.5%**	3/3 features
1.6 KNN	Weighted KNN	74.7%	3/3 features

8:

Model	Last change	Accuracy	Features
1.1 KNN	Fine KNN	71.2%	3/3 features
1.2 KNN	Medium KNN	76.0%	3/3 features
1.3 KNN	Coarse KNN	74.0%	3/3 features
1.4 KNN	Cosine KNN	65.5%	3/3 features
1.5 KNN	Cubic KNN	**76.1%**	3/3 features
1.6 KNN	Weighted KNN	75.8%	3/3 features

9:

1.1 KNN	Last change: Fine KNN	Accuracy: 71.1% 3/3 features
1.2 KNN	Last change: Medium KNN	Accuracy: 75.1% 3/3 features
1.3 KNN	Last change: Coarse KNN	Accuracy: 73.0% 3/3 features
1.4 KNN	Last change: Cosine KNN	Accuracy: 65.9% 3/3 features
1.5 KNN	Last change: Cubic KNN	Accuracy: **75.1%** 3/3 features
1.6 KNN	Last change: Weighted KNN	Accuracy: 74.9% 3/3 features

10:

1.1 KNN	Last change: Fine KNN	Accuracy: 71.7% 3/3 features
1.2 KNN	Last change: Medium KNN	Accuracy: 75.0% 3/3 features
1.3 KNN	Last change: Coarse KNN	Accuracy: 73.7% 3/3 features
1.4 KNN	Last change: Cosine KNN	Accuracy: 65.4% 3/3 features
1.5 KNN	Last change: Cubic KNN	Accuracy: **75.2%** 3/3 features
1.6 KNN	Last change: Weighted KNN	Accuracy: 75.0% 3/3 features